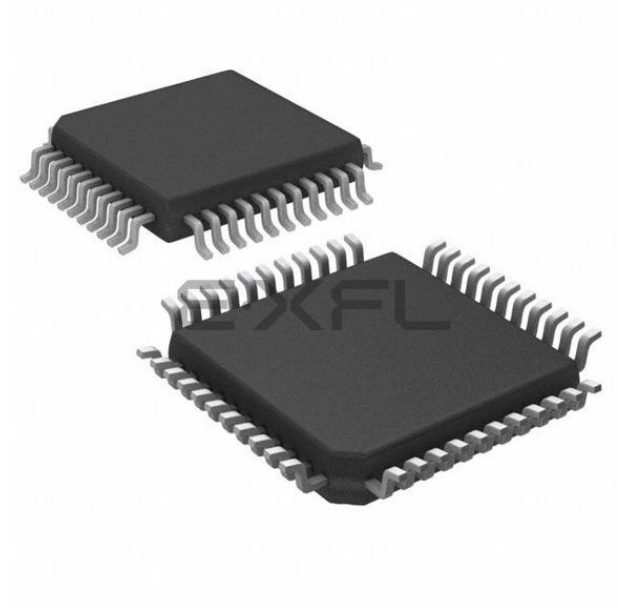


Welcome to E-XFL.COM

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	I²C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ap8cfbe

Chapter 13
Serial Peripheral Interface Module (SPI)

Chapter 2
Memory

2.1 Introduction

The CPU08 can address 64k-bytes of memory space. The memory map, shown in Figure 2-1, includes:

- 62,368 bytes of user FLASH — MC68HC908AP64
  32,768 bytes of user FLASH — MC68HC908AP32
  16,384 bytes of user FLASH — MC68HC908AP16
  8,192 bytes of user FLASH — MC68HC908AP8
- 2,048 bytes of RAM — MC68HC908AP64 and MC68HC908AP32
  1,024 bytes of RAM — MC68HC908AP16 and MC68HC908AP8
- 48 bytes of user-defined vectors
- 959 bytes of monitor ROM

2.2 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of $0000–$005F. Additional I/O registers have these addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE02; Reserved
- $FE03; SIM break flag control register, SBFCR
- $FE04; interrupt status register 1, INT1
- $FE05; interrupt status register 2, INT2
- $FE06; interrupt status register 3, INT3
- $FE07; Reserved
- $FE08; FLASH control register, FLCR
- $FE09; FLASH block protect register, FLBPR
- $FE0A; Reserved
- $FE0B; Reserved
- $FE0C; Break address register high, BRKH
- $FE0D; Break address register low, BRKL
- $FE0E; Break status and control register, BRKSCR
- $FE0F; LVI Status register, LVISR
- $FFCF; Mask option register, MOR (FLASH register)
- $FFFF; COP control register, COPCTL

2.3 Monitor ROM

The 959 bytes at addresses $FC00–$FDFF and $FE10–$FFCE are reserved ROM addresses that contain the instructions for the monitor functions. (See Chapter 8 Monitor ROM (MON).)

3.4 Configuration Register 2 (CONFIG2)

Address:	$001D							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	STOP_ ICLKDIS	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	0	SCIBD- SRC
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 3-3. Configuration Register 2 (CONFIG2)

STOP_ICLKDIS — Internal Oscillator Stop Mode Disable

STOP_ICLKDIS disables the internal oscillator during stop mode. Setting the STOP_ICLKDIS bit disables the oscillator during stop mode. (See Chapter 5 Oscillator (OSC).)

Reset clears this bit.

1 = Internal oscillator disabled during stop mode

0 = Internal oscillator enabled to operate during stop mode

STOP_RCLKEN — RC Oscillator Stop Mode Enable Bit

STOP_RCLKEN enables the RC oscillator to continue operating during stop mode. Setting the STOP_RCLKEN bit allows the oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See Chapter 5 Oscillator (OSC).)

Reset clears this bit.

1 = RC oscillator enabled to operate during stop mode

0 = RC oscillator disabled during stop mode

STOP_XCLKEN — X-tal Oscillator Stop Mode Enable Bit

STOP_XCLKEN enables the crystal (x-tal) oscillator to continue operating during stop mode. Setting the STOP_XCLKEN bit allows the x-tal oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See Chapter 5 Oscillator (OSC).) Reset clears this bit.

1 = X-tal oscillator enabled to operate during stop mode

0 = X-tal oscillator disabled during stop mode

OSCCLK1, OSCCLK0 — Oscillator Output Control Bits

OSCCLK1 and OSCCLK0 select which oscillator output to be driven out as OSCCLK to the timebase module (TBM). Reset clears these two bits.

OSCCLK1	OSCCLK0	Timebase Clock Source
0	0	Internal oscillator (ICLK)
0	1	RC oscillator (RCCLK)
1	0	X-tal oscillator (XTAL)
1	1	Not used

4.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC:$FFFD or with $FEFC:$FEFD in monitor mode

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

4.7 Instruction Set Summary

Table 4-1 provides a summary of the M68HC08 instruction set.

4.8 Opcode Map

The opcode map is provided in Table 4-2.

Table 4-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2

Chapter 6
Clock Generator Module (CGM)

6.1 Introduction

This section describes the clock generator module (CGM). The CGM generates the base clock signal, CGMOUT, which is based on either the oscillator clock divided by two or the divided phase-locked loop (PLL) clock, CGMPCLK, divided by two. CGMOUT is the clock from which the SIM derives the system clocks, including the bus clock, which is at a frequency of CGMOUT 2.

The PLL is a frequency generator designed for use with a low frequency crystal (typically 32.768 kHz) to generate a base frequency and dividing to a maximum bus frequency of 8 MHz.

6.2 Features

Features of the CGM include:

- Phase-locked loop with output frequency in integer multiples of an integer dividend of the crystal reference
- Low-frequency crystal operation with low-power operation and high-output frequency resolution
- Programmable prescaler for power-of-two increases in frequency
- Programmable hardware voltage-controlled oscillator (VCO) for low-jitter operation
- Automatic bandwidth control mode for low-jitter operation
- Automatic frequency lock detector
- CPU interrupt on entry or exit from locked condition
- Configuration register bit to allow oscillator operation during stop mode

6.3 Functional Description

The CGM consists of three major sub-modules:

- Oscillator module — The oscillator module generates the constant reference frequency clock, CGMRCLK (buffered CGMXCLK).
- Phase-locked loop (PLL) — The PLL generates the programmable VCO frequency clock, CGMVCLK, and the divided VCO clock, CGMPCLK.
- Base clock selector circuit — This software-controlled circuit selects either CGMXCLK divided by two or the divided VCO clock, CGMPCLK, divided by two as the base clock, CGMOUT. The SIM derives the system clocks from either CGMOUT or CGMXCLK.

Figure 6-1 shows the structure of the CGM.

Figure 6-2 is a summary of the CGM registers.

6.5.3 PLL Multiplier Select Registers

The PLL multiplier select registers (PMSH and PMSL) contain the programming information for the modulo feedback divider.

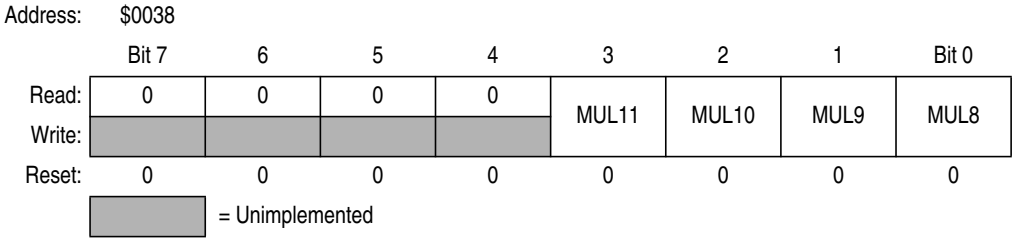


Figure 6-6. PLL Multiplier Select Register High (PMSH)

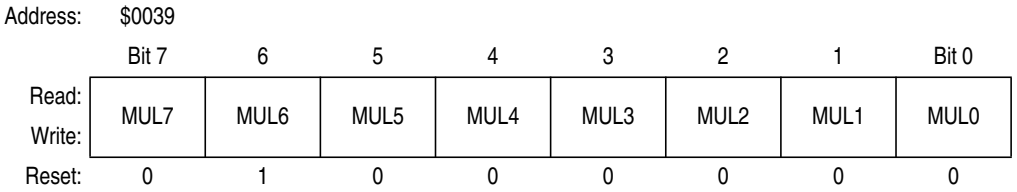


Figure 6-7. PLL Multiplier Select Register Low (PMSL)

MUL[11:0] — Multiplier Select Bits

These read/write bits control the modulo feedback divider that selects the VCO frequency multiplier N. (See 6.3.3 PLL Circuits and 6.3.6 Programming the PLL.) A value of $0000 in the multiplier select registers configure the modulo feedback divider the same as a value of $0001. Reset initializes the registers to $0040 for a default multiply value of 64.

NOTE

The multiplier select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).

6.5.4 PLL VCO Range Select Register

The PLL VCO range select register (PMRS) contains the programming information required for the hardware configuration of the VCO.

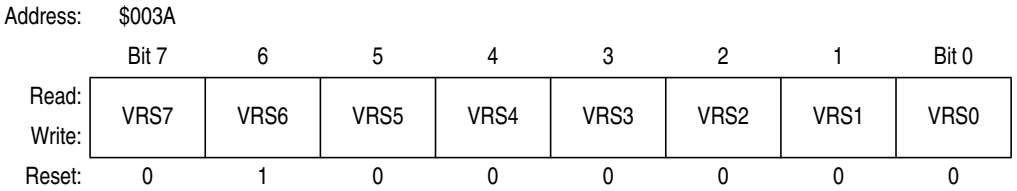


Figure 6-8. PLL VCO Range Select Register (PMRS)

VRS[7:0] — VCO Range Select Bits

These read/write bits control the hardware center-of-range linear multiplier L which, in conjunction with E (See 6.3.3 PLL Circuits, 6.3.6 Programming the PLL, and 6.5.1 PLL Control Register.), controls the hardware center-of-range frequency, f_{VRS}. VRS[7:0] cannot be written when the PLLON bit in the PCTL is set. (See 6.3.7 Special Programming Exceptions.) A value of $00 in the VCO range select

$FE04	Interrupt Status Register 1 (INT1)	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	IF21	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 7-2. SIM I/O Register Summary

7.2 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 7-3. This clock can come from either an external oscillator or from the on-chip PLL. (See Chapter 6 Clock Generator Module (CGM).)

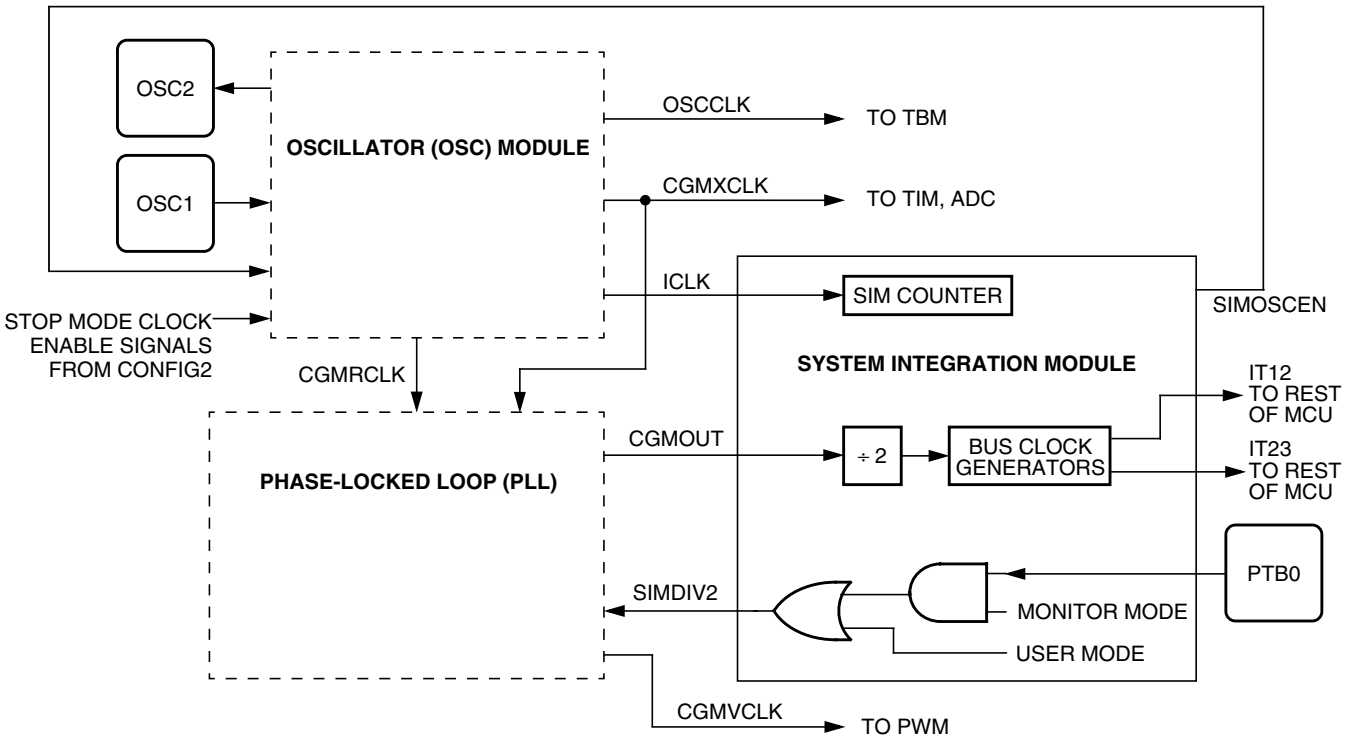


Figure 7-3. CGM Clock Signals

7.2.1 Bus Timing

In user mode, the internal bus frequency is either the oscillator output (CGMXCLK) divided by four or the divided PLL output (CGMPCLK) divided by four.

8.5.6 EE_WRITE

EE_WRITE is used to write a set of data from the data array to FLASH.

Table 8-16. EE_WRITE Routine

Routine Name	EE_WRITE
Routine Description	Emulated EEPROM write. Data size ranges from 7 to 15 bytes at a time.
Calling Address	$FF36
Stack Used	30 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE)(1) Starting address (ADDRH)(2) Starting address (ADDRL)(1) Data 1 : Data N

1. The minimum data size is 7 bytes. The maximum data size is 15 bytes.
2. The start address must be a page boundary start address.

The start location of the FLASH to be programmed is specified by the address ADDRH:ADDRL and the number of bytes in the data array is specified by DATASIZE. The minimum number of bytes that can be programmed in one routine call is 7 bytes, the maximum is 15 bytes. ADDRH:ADDRL must always be the start of boundary address (the page start address: $X000, $X200, $X400, $X600, $X800, $XA00, $XC00, or $XE00) and DATASIZE must be the same size when accessing the same page.

In some applications, the user may want to repeatedly store and read a set of data from an area of non-volatile memory. This is easily possible when using an EEPROM array. As the write and erase operations can be executed on a byte basis. For FLASH memory, the minimum erase size is the page — 512 bytes per page for MC68HC908AP64. If the data array size is less than the page size, writing and erasing to the same page cannot fully utilize the page. Unused locations in the page will be wasted. The EE_WRITE routine is designed to emulate the properties similar to the EEPROM. Allowing a more efficient use of the FLASH page for data storage.

When the user dedicates a page of FLASH for data storage, and the size of the data array defined, each call of the EE_WRTIE routine will automatically transfer the data in the data array (in RAM) to the next blank block of locations in the FLASH page. Once a page is filled up, the EE_WRITE routine automatically erases the page, and starts reuse the page again. In the 512-byte page, an 9-byte control block is used by the routine to monitor the utilization of the page. In effect, only 503 bytes are used for data storage. (see Figure 8-10). The page control operations are transparent to the user.

When using this routine to store a 8-byte data array, the FLASH page can be programmed 62 times before the an erase is required. In effect, the write/erase endurance is increased by 62 times. When a 15-byte data array is used, the write/erase endurance is increased by 33 times. Due to the FLASH page size limitation, the data array is limited from 7 bytes to 15 bytes.

TBR[2:0] — Timebase Rate Selection

These read/write bits are used to select the rate of timebase interrupts as shown in Table 10-1.

NOTE

Do not change TBR[2:0] bits while the timebase is enabled (TBON = 1).

Table 10-1. Timebase Rate Selection for OSCCLK = 32.768-kHz

TBR2	TBR1	TBR0	Divider	Timebase Interrupt Rate	
				Hz	ms
0	0	0	262144	0.125	8000
0	0	1	131072	0.25	4000
0	1	0	65536	0.5	2000
0	1	1	32768	1	1000
1	0	0	64	512	~2
1	0	1	32	1024	~1
1	1	0	16	2048	~0.5
1	1	1	8	4096	~0.24

TACK — Timebase ACKnowledge

The TACK bit is a write-only bit and always reads as 0. Writing a logic 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a logic 0 to this bit has no effect.

1 = Clear timebase interrupt flag
0 = No effect

TBIE — Timebase Interrupt Enabled

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt enabled
0 = Timebase interrupt disabled

TBON — Timebase Enabled

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase enabled
0 = Timebase disabled and the counter initialized to 0’s

11.5 Low-Power Modes

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

11.5.1 Wait Mode

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

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

Refer to 7.6 Low-Power Modes for information on exiting wait mode.

11.5.2 Stop Mode

The SCI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect SCI register states. SCI module operation resumes after an external interrupt.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

Refer to 7.6 Low-Power Modes for information on exiting stop mode.

11.6 SCI During Break Module Interrupts

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 logic 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 logic 0 to the BCFE bit. With BCFE at logic 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 logic 0. After the break, doing the second step clears the status bit.

11.7 I/O Signals

Port B shares two of its pins with the SCI module.
The two SCI I/O pins are:

- PTB2/TxD — Transmit data
- PTB3/RxD — Receive data

11.7.1 TxD (Transmit Data)

When the SCI is enabled (ENSCI=1), the PTB2/TxD pin becomes the serial data output, TxD, from the SCI transmitter regardless of the state of the DDRB2 bit in data direction register B (DDRB). The TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.

11.8.5 SCI Status Register 2

SCI status register 2 contains flags to signal the following conditions:

- Break character detected
- Incoming data

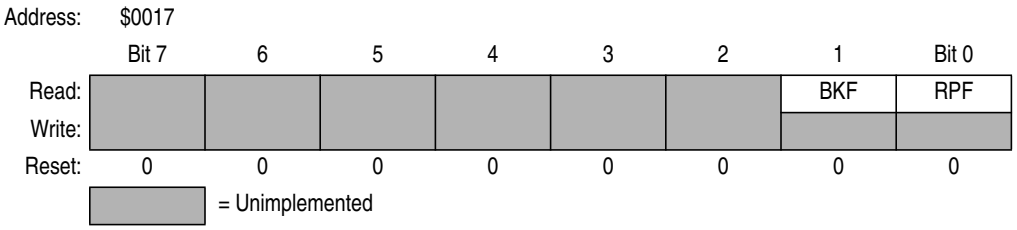


Figure 11-14. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI 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 logic 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 logic 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 SCI module or entering stop mode can show whether a reception is in progress.

1 = Reception in progress
0 = No reception in progress

11.8.6 SCI Data Register

The SCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the SCI data register.

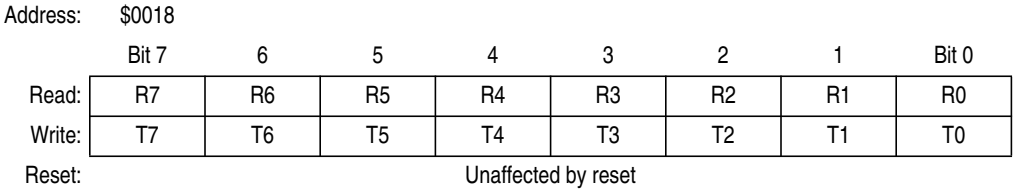


Figure 11-15. SCI Data Register (SCDR)

R7/T7–R0/T0 — Receive/Transmit Data Bits

Reading the SCDR accesses the read-only received data bits, R7–R0. Writing to the SCDR writes the data to be transmitted, T7–T0. Reset has no effect on the SCDR.

NOTE

Do not use read/modify/write instructions on the SCI data register.

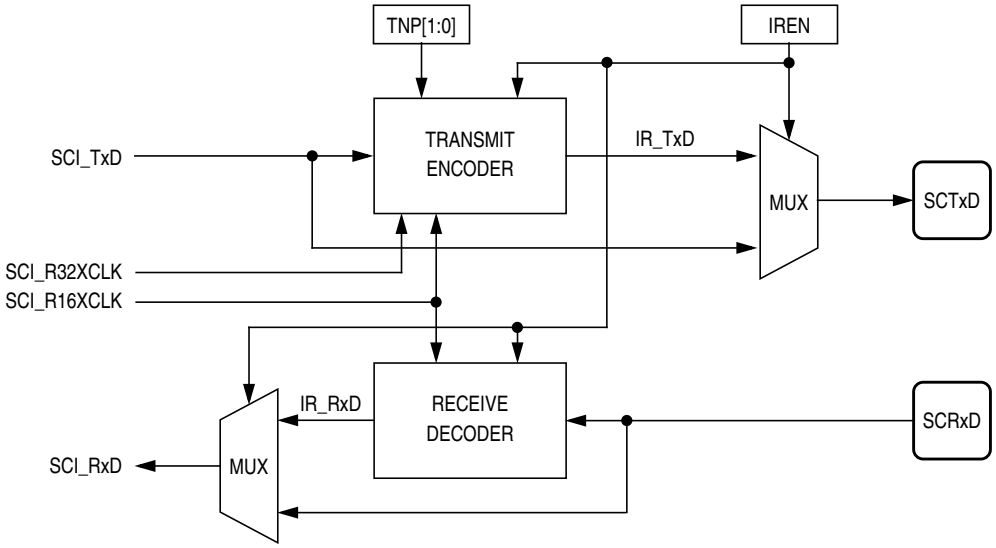


Figure 12-3. Infrared Sub-Module Diagram

12.4.1 Infrared Transmit Encoder

The infrared transmit encoder converts the "0" bits in the serial data stream from the SCI module to narrow "low" pulses, to the TxD pin. The narrow pulse is sent with a duration of 1/32, 1/16, or 3/16 of a data bit width**.** When two consecutive zeros are sent, the two consecutive narrow pulses will be separated by a time equal to a data bit width.

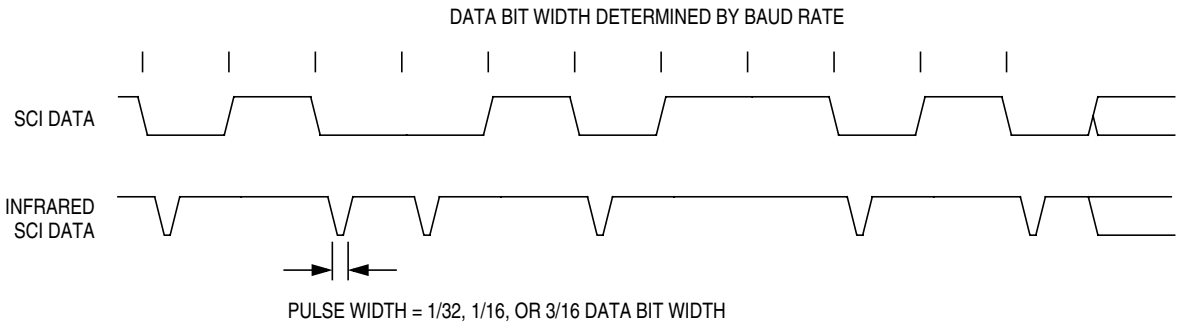


Figure 12-4. Infrared SCI Data Example

12.4.2 Infrared Receive Decoder

The infrared receive decoder converts low narrow pulses from the RxD pin to standard SCI data bits. The reference clock, SCI_R16XCLK, clocks a four bit internal counter which counts from 0 to 15. An incoming pulse starts the internal counter and a "0" is sent out to the IR_RxD output. Subsequent incoming pulses are ignored when the counter count is between 0 and 7; IR_RxD remains "0". Once the counter passes 7, an incoming pulse will reset the counter; IR_RxD remains "0". When the counter reaches 15, the IR_RxD output returns to "1", the counter stops and waits for further pulses. A pulse is interpreted as jitter if it arrives shortly after the counter reaches 15; IR_RxD remains "1".

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0010	SPI Control Register (SPCR)	Read: Write:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Reset:	0	0	1	0	1	0	0	0
$0011	SPI Status and Control Register (SPSCR)	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved

Figure 13-1. SPI I/O Register Summary

13.4 Functional Description

Figure 13-2 shows the structure of the SPI module.

The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. Software can poll the SPI status flags or SPI operation can be interrupt-driven.

The following paragraphs describe the operation of the SPI module.

13.4.1 Master Mode

The SPI operates in master mode when the SPI master bit, SPMSTR, is set.

NOTE

Configure the SPI modules as master or slave before enabling them. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI. (See 13.13.1 SPI Control Register.)

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the transmit data register. If the shift register is empty, the byte immediately transfers to the shift register, setting the SPI transmitter empty bit, SPTE. The byte begins shifting out on the MOSI pin under the control of the serial clock. (See Figure 13-3.)

The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. (See 13.13.2 SPI Status and Control Register.) Through the SPSCK pin, the baud rate generator of the master also controls the shift register of the slave peripheral.

As the byte shifts out on the MOSI pin of the master, another byte shifts in from the slave on the master's MISO pin. The transmission ends when the receiver full bit, SPRF, becomes set. At the same time that SPRF becomes set, the byte from the slave transfers to the receive data register. In normal operation, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Writing to the SPI data register clears the SPTE bit.

MMSRW — MMIIC Slave Read/Write Select

This bit indicates the data direction when the module is in slave mode. It is updated after the calling address is received from a master device. MMSRW = 1 when the calling master is reading data from the module (slave transmit mode). MMSRW = 0 when the master is writing data to the module (receive mode).

1 = Slave mode transmit
0 = Slave mode receive

MMRXAK — MMIIC Receive Acknowledge

When this bit is cleared, it indicates an acknowledge signal has been received after the completion of eight data bits transmission on the bus. When MMRXAK is set, it indicates no acknowledge signal has been detected at the 9th clock; the module will release the SDA line for the master to generate STOP or repeated START condition. Reset sets this bit.

1 = No acknowledge signal received at 9th clock
0 = Acknowledge signal received at 9th clock

MMCRCBF — CRC Data Buffer Full Flag

This flag is set when the CRC data register (MMCRCDR) is loaded with a CRC byte for the current received or transmitted data.

In transmit mode, after a byte of data has been sent (MMTXIF = 1), the MMCRCBF will be set when the CRC byte has been generated and ready in the MMCRCDR. The content of the MMCRCDR should be copied to the MMDTR for transmission.

In receive mode, the MMCRCBF is set when the CRC byte has been generated and ready in MMCRCDR, for the current byte of received data.

The MMCRCBF bit is cleared when the CRC data register is read. Reset also clears this bit.

1 = Data ready in CRC data register (MMCRCDR)
0 = Data not ready in CRC data register (MMCRCDR)

MMTXBE — MMIIC Transmit Buffer Empty

This flag indicates the status of the data transmit register (MMDTR). When the CPU writes the data to the MMDTR, the MMTXBE flag will be cleared. MMTXBE is set when MMDTR is emptied by a transfer of its data to the output circuit. Reset sets this bit.

1 = Data transmit register empty
0 = Data transmit register full

MMRXBF — MMIIC Receive Buffer Full

This flag indicates the status of the data receive register (MMDRR). When the CPU reads the data from the MMDRR, the MMRXBF flag will be cleared. MMRXBF is set when MMDRR is full by a transfer of data from the input circuit to the MMDRR. Reset clears this bit.

1 = Data receive register full
0 = Data receive register empty

14.6.5 MMIIC Data Transmit Register (MMDTR)

Address: $004C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MMTD7	MMTD6	MMTD5	MMTD4	MMTD3	MMTD2	MMTD1	MMTD0
Reset:	0	0	0	0	0	0	0	0

Figure 14-8. MMIIC Data Transmit Register (MMDTR)

The ADC conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is either the bus clock or CGMXCLK and is selectable by the ADICLK bit located in the ADC clock register. The divide ratio is selected by the ADIV[2:0] bits.

For example, if a 4MHz CGMXCLK is selected as the ADC input clock source, with a divide-by-four prescale, and the bus speed is set at 2MHz:

$$\text{Conversion time} = \frac{16 \text{ to} 17 \text{ ADC cycles}}{4\text{MHz} \div 4} = 16 \text{ to } 17 \ \mu\text{s}$$

$$\text{Number of bus cycles} = 16 \ \mu\text{s} \times 2\text{MHz} = 32 \text{ to } 34 \text{ cycles}$$

NOTE

The ADC frequency must be between f_{ADIC} minimum and f_{ADIC} maximum to meet A/D specifications. (See 22.5 5V DC Electrical Characteristics.).

Since an ADC cycle may be comprised of several bus cycles (four in the previous example) and the start of a conversion is initiated by a bus cycle write to the ADSCR, from zero to four additional bus cycles may occur before the start of the initial ADC cycle. This results in a fractional ADC cycle and is represented as the 17th cycle.

15.3.4 Continuous Conversion

In the continuous conversion mode, the ADC continuously converts the selected channel, filling the ADC data register with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit is set after each conversion and can be cleared by writing to the ADC status and control register or reading of the ADRL0 data register.

15.3.5 Auto-Scan Mode

In auto-scan mode, the ADC input channel is selected by the value of the 2-bit up-counter, instead of the channel select bits, ADCH[4:0]. The value of the counter also defines the data register ADRLx to be used to store the conversion result. When ASCAN bit is set, a write to ADC status and control register (ADSCR) will reset the auto-scan up-counter and ADC conversion will start on the channel 0 up to the channel number defined by the integer value of AUTO[1:0]. After a channel conversion is completed, data is stored in ADRLx and the COCO-bit will be set. The counter value will be incremented by 1 and a new conversion will start. This process will continue until the counter value reaches the value of AUTO[1:0]. When this happens, it indicates that the current channel is the last channel to be converted. Upon the completion on the last channel, the counter value will not be incremented and no further conversion will be performed. To start another auto-scan cycle, a write to ADSCR must be performed.

NOTE

The system only provides 8-bit data storage in auto-scan code, user must clear MODE[1:0] bits to select 8-bit truncation mode before entering auto-scan mode.

It is recommended that user should disable the auto-scan function before switching channel and also before entering STOP mode.

Table 15-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	ADC input clock ÷ 1
0	0	1	ADC input clock ÷ 2
0	1	0	ADC input clock ÷ 4
0	1	1	ADC input clock ÷ 8
1	X	X	ADC input clock ÷ 16

X = don't care

ADICLK — ADC Input Clock Select Bit

ADICLK selects either bus clock or CGMXCLK as the input clock source to generate the internal ADC clock. Reset selects CGMXCLK as the ADC clock source.

If the external clock (CGMXCLK) is equal to or greater than 1 MHz, CGMXCLK can be used as the clock source for the ADC. If CGMXCLK is less than 1 MHz, use the PLL-generated bus clock as the clock source. As long as the internal ADC clock is at f_{ADIC}, correct operation can be guaranteed.

1 = Internal bus clock
0 = External clock, CGMXCLK

$$f_{ADIC} = \frac{\text{CGMXCLK or bus frequency}}{\text{ADIV[2:0]}}$$

MODE1 and MODE0 — Modes of Result Justification

MODE1 and MODE0 selects between four modes of operation. The manner in which the ADC conversion results will be placed in the ADC data registers is controlled by these modes of operation. Reset returns right-justified mode.

Table 15-3. ADC Mode Select

MODE1	MODE0	Justification Mode
0	0	8-bit truncated mode
0	1	Right justified mode
1	0	Left justified mode
1	1	Left justified sign data mode

15.7.3 ADC Data Register 0 (ADRH0 and ADRL0)

The ADC data register 0 consist of a pair of 8-bit registers: high byte (ADRH0), and low byte (ADRL0). This pair form a 16-bit register to store the 10-bit ADC result for the selected ADC result justification mode.

In 8-bit truncated mode, the ADRL0 holds the eight most significant bits (MSBs) of the 10-bit result. The ADRL0 is updated each time an ADC conversion completes. In 8-bit truncated mode, ADRL0 contains no interlocking with ADRH0. (See Figure 15-5 . ADRH0 and ADRL0 in 8-Bit Truncated Mode.)

Table 16-5 summarizes the operation of the port D pins.

Table 16-5. Port D Pin Functions

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

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

Chapter 18
Keyboard Interrupt Module (KBI)

18.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTD0–PTD7. When a port pin is enabled for keyboard interrupt function, an internal 30kΩ pullup device is also enabled on the pin.

18.2 Features

Features of the keyboard interrupt module include the following:

- Eight keyboard interrupt pins with pullup devices
- Separate keyboard interrupt enable bits and one keyboard interrupt mask
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-lower modes

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER)	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-1. KBI I/O Register Summary

18.3 I/O Pins

The eight keyboard interrupt pins are shared with standard port I/O pins. The full name of the KBI pins are listed in Table 18-1. The generic pin name appear in the text that follows.

Table 18-1. Pin Name Conventions

KBI Generic Pin Name	Full MCU Pin Name	Pin Selected for KBI Function by KBIEx Bit in KBIER
KBI0–KBI7	PTD0/KBI0–PTD7/KBI7	KBIE0–KBIE7

21.4.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set. (see Chapter 7 System Integration Module (SIM)) Clear the BW bit by writing logic 0 to it.

21.4.2 Stop Mode

A break interrupt causes exit from stop mode and sets the SBSW bit in the break status register.

21.5 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

21.5.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

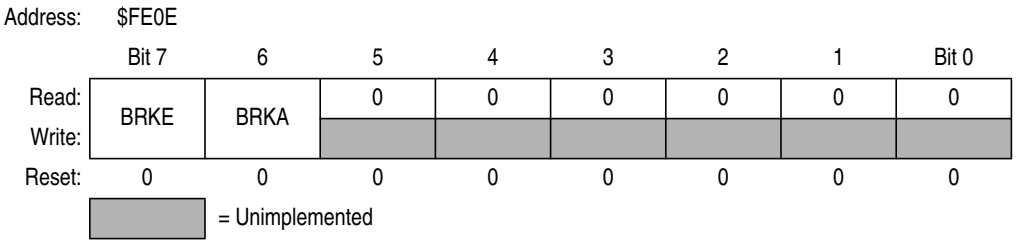


Figure 21-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

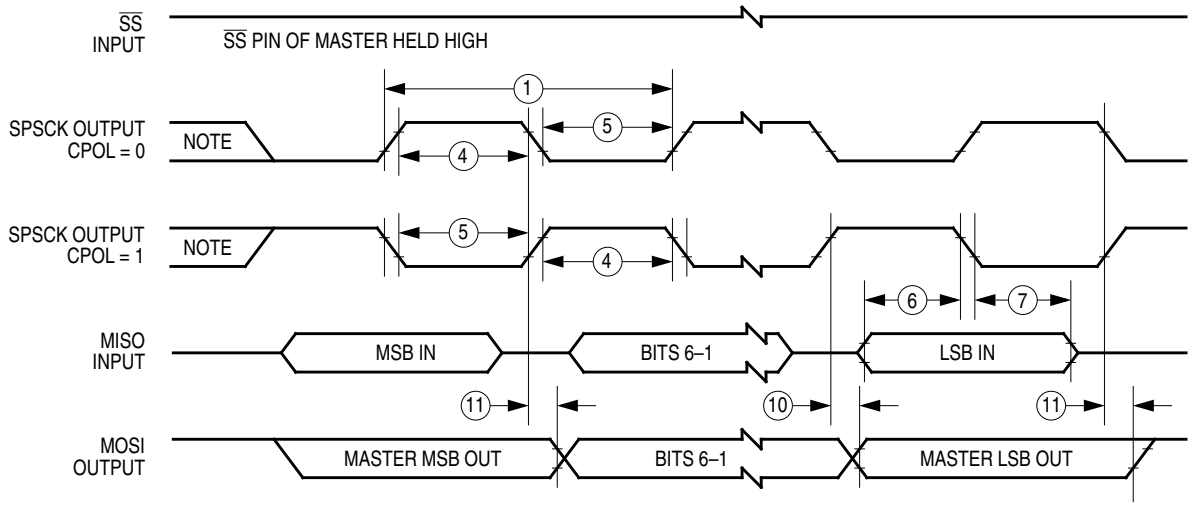
This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

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

BRKA — Break Active Bit

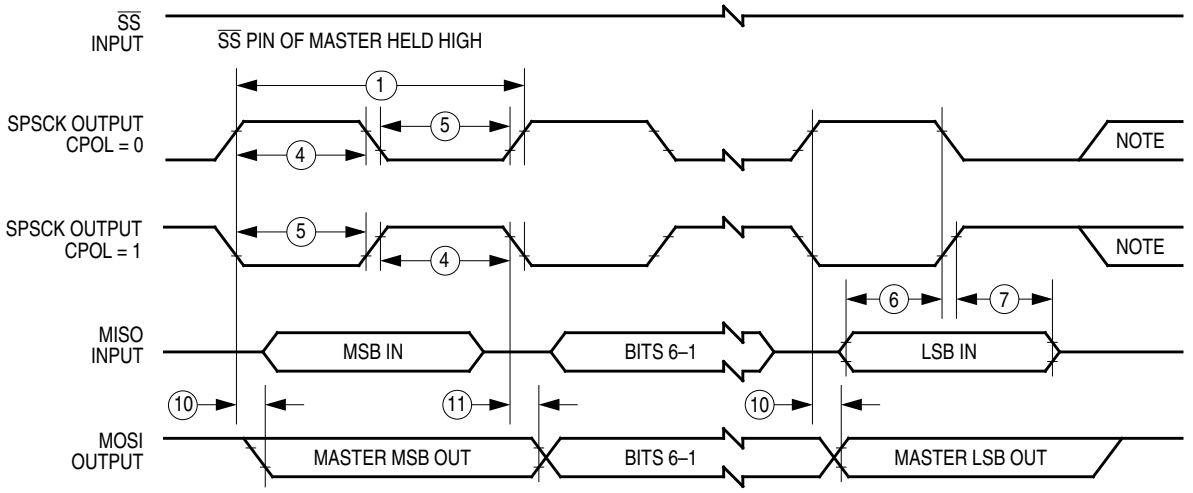
This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = (When read) Break address match
0 = (When read) No break address match



Note: This first clock edge is generated internally, but is not seen at the SPSCK pin.

a) SPI Master Timing (CPHA = 0)



Note: This last clock edge is generated internally, but is not seen at the SPSCK pin.

b) SPI Master Timing (CPHA = 1)

Figure 22-3. SPI Master Timing