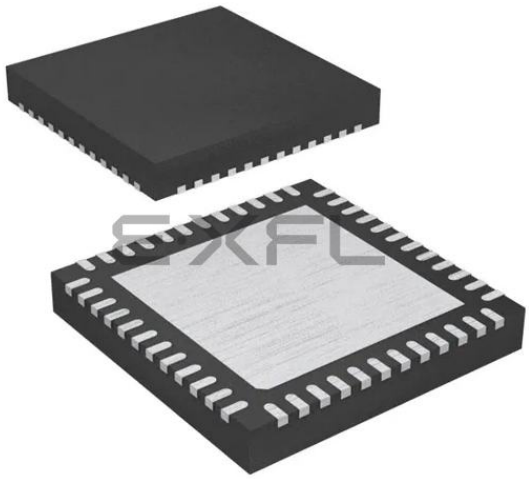


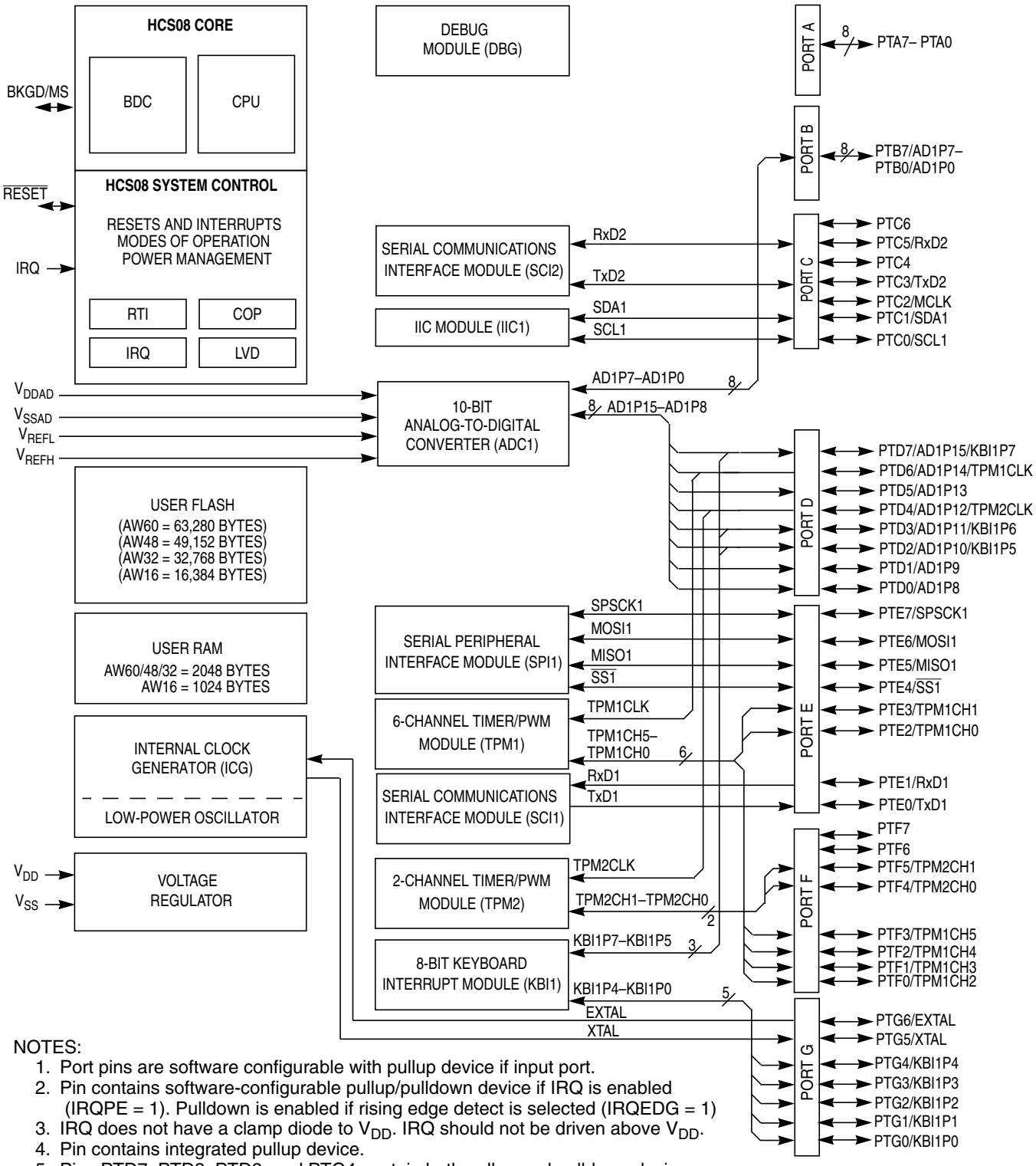
Welcome to E-XFL.COM

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	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	38
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08aw32e7cfde



NOTES:
1. Port pins are software configurable with pullup device if input port.
2. Pin contains software-configurable pullup/pulldown device if IRQ is enabled (IRQPE = 1). Pulldown is enabled if rising edge detect is selected (IRQEDG = 1)
3. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. Pins PTD7, PTD3, PTD2, and PTG4 contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).

Figure 1-1. MC9S08AW60 Series Block Diagram

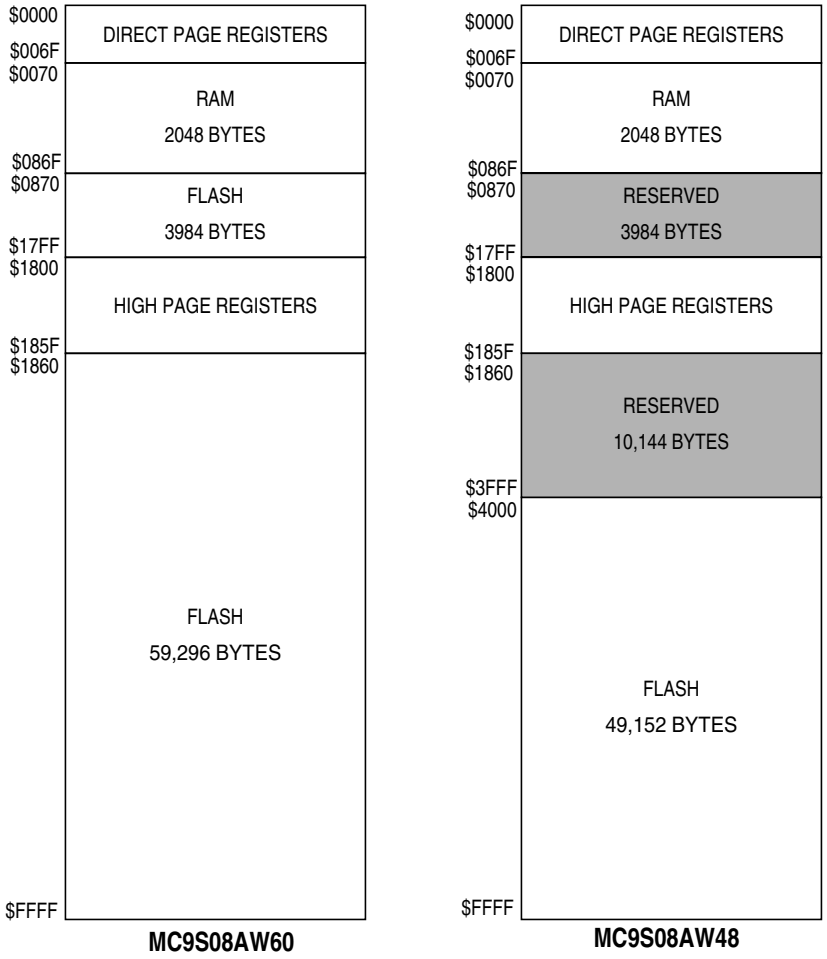


Figure 4-1. MC9S08AW60 and MC9S08AW48 Memory Map

Table 4-2. Direct-Page Register Summary (Sheet 1 of 3)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0000	**PTAD**	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
$0001	**PTADD**	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
$0002	**PTBD**	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
$0003	**PTBDD**	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
$0004	**PTCD**	0	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
$0005	**PTCDD**	0	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
$0006	**PTDD**	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
$0007	**PTDDD**	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
$0008	**PTED**	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
$0009	**PTEDD**	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
$000A	**PTFD**	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
$000B	**PTFDD**	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
$000C	**PTGD**	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
$000D	**PTGDD**	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
$000E– $000F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
$0010	**ADC1SC1**	COCO	AIEN	ADCO	ADCH				
$0011	**ADC1SC2**	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
$0012	**ADC1RH**	0	0	0	0	0	0	ADR9	ADR8
$0013	**ADC1RL**	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
$0014	**ADC1CVH**	0	0	0	0	0	0	ADCV9	ADCV8
$0015	**ADC1CVL**	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
$0016	**ADC1CFG**	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
$0017	**APCTL1**	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
$0018	**APCTL2**	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
$0019	**APCTL3**	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
$001A– $001B	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
$001C	**IRQSC**	0	0	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
$001D	Reserved	—	—	—	—	—	—	—	—
$001E	**KBI1SC**	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
$001F	**KBI1PE**	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
$0020	**TPM1SC**	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0021	**TPM1CNTH**	Bit 15	14	13	12	11	10	9	Bit 8
$0022	**TPM1CNTL**	Bit 7	6	5	4	3	2	1	Bit 0
$0023	**TPM1MODH**	Bit 15	14	13	12	11	10	9	Bit 8
$0024	**TPM1MODL**	Bit 7	6	5	4	3	2	1	Bit 0
$0025	**TPM1C0SC**	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0026	**TPM1C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
$0027	**TPM1C0VL**	Bit 7	6	5	4	3	2	1	Bit 0

5.9.2 System Reset Status Register (SRS)

This register includes seven read-only status flags to indicate the source of the most recent reset. When a debug host forces reset by writing 1 to BDFR in the SBDFR register, none of the status bits in SRS will be set. Writing any value to this register address clears the COP watchdog timer without affecting the contents of this register. The reset state of these bits depends on what caused the MCU to reset.

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	0	ICG	LVD	0
W	Writing any value to SIMRS address clears COP watchdog timer.							
POR	1	0	0	0	0	0	1	0
LVR:	U	0	0	0	0	0	1	0
Any other reset:	0	(1)	(1)	(1)	0	(1)	0	0

U = Unaffected by reset

1 Any of these reset sources that are active at the time of reset will cause the corresponding bit(s) to be set; bits corresponding to sources that are not active at the time of reset will be cleared.

Figure 5-3. System Reset Status (SRS)

Table 5-3. SRS Register Field Descriptions

Field	Description
7 POR	**Power-On Reset** — Reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVR) status bit is also set to indicate that the reset occurred while the internal supply was below the LVR threshold. 0 Reset not caused by POR. 1 POR caused reset.
6 PIN	**External Reset Pin** — Reset was caused by an active-low level on the external reset pin. 0 Reset not caused by external reset pin. 1 Reset came from external reset pin.
5 COP	**Computer Operating Properly (COP) Watchdog** — Reset was caused by the COP watchdog timer timing out. This reset source may be blocked by COPE = 0. 0 Reset not caused by COP timeout. 1 Reset caused by COP timeout.
4 ILOP	**Illegal Opcode** — Reset was caused by an attempt to execute an unimplemented or illegal opcode. The STOP instruction is considered illegal if stop is disabled by STOPE = 0 in the SOPT register. The BGND instruction is considered illegal if active background mode is disabled by ENBDM = 0 in the BDCSC register. 0 Reset not caused by an illegal opcode. 1 Reset caused by an illegal opcode.

	7	6	5	4	3	2	1	0
R W	PTFDS7	PTFDS6	PTFDS5	PTFDS4	PTFDS3	PTFDS2	PTFDS1	PTFDS0
Reset	0	0	0	0	0	0	0	0

Figure 6-38. Output Drive Strength Selection for Port F (PTFDS)

Table 6-31. PTFDS Register Field Descriptions

Field	Description
7:0 PTFDS[7:0]	**Output Drive Strength Selection for Port F Bits** — Each of these control bits selects between low and high output drive for the associated PTF pin. 0 Low output drive enabled for port F bit n. 1 High output drive enabled for port F bit n.

8.4.11 Fixed Frequency Clock

The ICG provides a fixed frequency clock output, XCLK, for use by on-chip peripherals. This output is equal to the internal bus clock, BUSCLK, in all modes except FEE. In FEE mode, XCLK is equal to ICGERCLK ÷ 2 when the following conditions are met:

- $(P \times N) \div R \geq 4$ where P is determined by RANGE (see Table 8-11), N and R are determined by MFD and RFD respectively (see Table 8-12).
- LOCK = 1.

If the above conditions are not true, then XCLK is equal to BUSCLK.

When the ICG is in either FEI or SCM mode, XCLK is turned off. Any peripherals which can use XCLK as a clock source must not do so when the ICG is in FEI or SCM mode.

8.4.12 High Gain Oscillator

The oscillator has the option of running in a high gain oscillator (HGO) mode, which improves the oscillator's resistance to EMC noise when running in FBE or FEE modes. This option is selected by writing a 1 to the HGO bit in the ICGC1 register. HGO is used with both the high and low range oscillators but is only valid when REFS = 1 in the ICGC1 register. When HGO = 0, the standard low-power oscillator is selected. This bit is writable only once after any reset.

8.5 Initialization/Application Information

8.5.1 Introduction

The section is intended to give some basic direction on which configuration a user would want to select when initializing the ICG. For some applications, the serial communication link may dictate the accuracy of the clock reference. For other applications, lowest power consumption may be the chief clock consideration. Still others may have lowest cost as the primary goal. The ICG allows great flexibility in choosing which is best for any application.

Chapter 9
Keyboard Interrupt (S08KBIV1)

9.1 Introduction

The MC9S08AW60 Series has one KBI module with eight keyboard interrupt inputs that are shared with port D and port G pins. See Chapter 2, "Pins and Connections," for more information about the logic and hardware aspects of these pins.

9.2 Keyboard Pin Sharing

The KBI input KBIP7 shares a common pin with PTD7 and AD15. When KBIP7 is enabled the pin is forced to its input state regardless of the value of the associated port D data direction bit. The port D pullup enable is still used to control the pullup resistor and the pin state can be sensed through a read of the port D data register (this requires that bit 7 of the port D DDR is 0). In the case that the pin is enabled as an ADC input, both the PTD7 and KBIP7 functions are disabled, including the pullup resistor.

The KBI input KBIP6 shares a common pin with PTD3 and AD11, and KBI input KBIP5 shares a common pin with PTD2 and AD10. The sharing of each of these inputs with port and ADC functions operates in the same way as described above for KBIP7.

The KBI inputs KBIP4 – KBIP0 are shared on common pins with PTG4 – PTG0. These pins all operate in the same way as described above for KBIP7 except that none are shared with an ADC input.

KBIP3 – KBIP0 are always falling-edge/low-level sensitive. KBIP7 – KBIP4 can be configured for rising-edge/high-level or for falling-edge/low-level sensitivity. When any of the inputs KBIP7 – KBIP0 are enabled and configured to detect rising edges/high levels, and the pin pullup is enabled through the corresponding port pullup enable bit for that pin, a pulldown resistor rather than a pullup resistor is enabled on the pin.

Table 9-1. KBI and Parallel I/O Interaction

PTxPEn (Pull Enable)	PTxDDn (Data Direction)	KBIPEn (KBI Pin Enable)	KBEDGn (KBI Edge Select)	Pullup	Pulldown
0	0	0	x[1]	disabled	disabled
1	0	0	x	enabled	disabled
x	1	0	x	disabled	disabled
1	x	1	0	enabled	disabled
1	x	1	1	disabled	enabled
0	x	1	x	disabled	disabled

1 x = Don't care

Table 10-2. TPM Clock Source Selection

CLKSB:CLKSA	TPM Clock Source to Prescaler Input
0:0	No clock selected (TPMx disabled)
0:1	Bus rate clock (BUSCLK)
1:0	Fixed system clock (XCLK)
1:1	External source (TPMxCLK)[1,2]

1 The maximum frequency that is allowed as an external clock is one-fourth of the bus frequency.

2 If the external clock input is shared with channel n and is selected as the TPM clock source, the corresponding ELSnB:ELSnA control bits should be set to 0:0 so channel n does not try to use the same pin for a conflicting function.

Table 10-3. Prescale Divisor Selection

PS2:PS1:PS0	TPM Clock Source Divided-By
0:0:0	1
0:0:1	2
0:1:0	4
0:1:1	8
1:0:0	16
1:0:1	32
1:1:0	64
1:1:1	128

10.4.2 Timer x Counter Registers (TPMxCNTH:TPMxCNTL)

The two read-only TPM counter registers contain the high and low bytes of the value in the TPM counter. Reading either byte (TPMxCNTH or TPMxCNTL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This allows coherent 16-bit reads in either order. The coherency mechanism is automatically restarted by an MCU reset, a write of any value to TPMxCNTH or TPMxCNTL, or any write to the timer status/control register (TPMxSC).

Reset clears the TPM counter registers.

	7	6	5	4	3	2	1	0
R	Bit 15	14	13	12	11	10	9	Bit 8
W	Any write to TPMxCNTH clears the 16-bit counter.							
Reset	0	0	0	0	0	0	0	0

Figure 10-4. Timer x Counter Register High (TPMxCNTH)

11.1.1 Features

Features of SCI module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:
  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Receiver wakeup by idle-line or address-mark
- Optional 13-bit break character
- Selectable transmitter output polarity

11.1.2 Modes of Operation

See Section 11.3, “Functional Description,” for a detailed description of SCI operation in the different modes.

- 8- and 9-bit data modes
- Stop modes — SCI is halted during all stop modes
- Loop mode
- Single-wire mode

11.1.3 Block Diagram

Figure 11-2 shows the transmitter portion of the SCI.

11.3.1 Baud Rate Generation

As shown in Figure 11-12, the clock source for the SCI baud rate generator is the bus-rate clock.

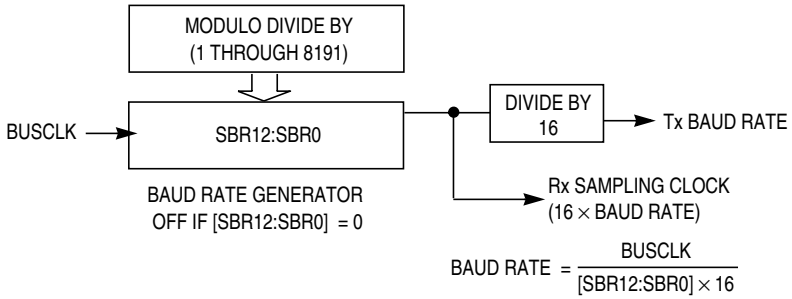


Figure 11-12. SCI Baud Rate Generation

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a Freescale Semiconductor SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

11.3.2 Transmitter Functional Description

This section describes the overall block diagram for the SCI transmitter, as well as specialized functions for sending break and idle characters. The transmitter block diagram is shown in Figure 11-2.

The transmitter output (TxD) idle state defaults to logic high (TXINV = 0 following reset). The transmitter output is inverted by setting TXINV = 1. The transmitter is enabled by setting the TE bit in SCIxC2. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register (SCIxD).

The central element of the SCI transmitter is the transmit shift register that is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, we will assume M = 0, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at SCIxD.

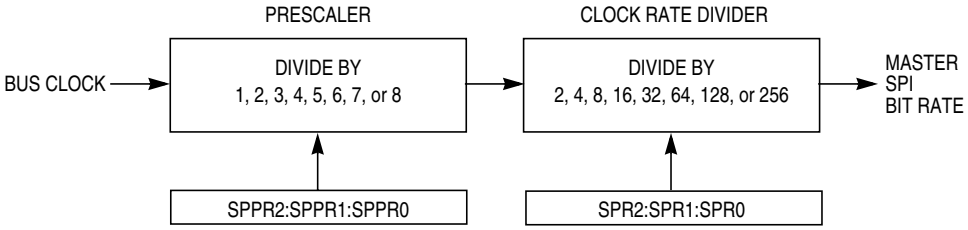


Figure 12-4. SPI Baud Rate Generation

12.1 External Signal Description

The SPI optionally shares four port pins. The function of these pins depends on the settings of SPI control bits. When the SPI is disabled (SPE = 0), these four pins revert to being general-purpose port I/O pins that are not controlled by the SPI.

12.1.1 SPSCK — SPI Serial Clock

When the SPI is enabled as a slave, this pin is the serial clock input. When the SPI is enabled as a master, this pin is the serial clock output.

12.1.2 MOSI — Master Data Out, Slave Data In

When the SPI is enabled as a master and SPI pin control zero (SPC0) is 0 (not bidirectional mode), this pin is the serial data output. When the SPI is enabled as a slave and SPC0 = 0, this pin is the serial data input. If SPC0 = 1 to select single-wire bidirectional mode, and master mode is selected, this pin becomes the bidirectional data I/O pin (MOMI). Also, the bidirectional mode output enable bit determines whether the pin acts as an input (BIDIROE = 0) or an output (BIDIROE = 1). If SPC0 = 1 and slave mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

12.1.3 MISO — Master Data In, Slave Data Out

When the SPI is enabled as a master and SPI pin control zero (SPC0) is 0 (not bidirectional mode), this pin is the serial data input. When the SPI is enabled as a slave and SPC0 = 0, this pin is the serial data output. If SPC0 = 1 to select single-wire bidirectional mode, and slave mode is selected, this pin becomes the bidirectional data I/O pin (SISO) and the bidirectional mode output enable bit determines whether the pin acts as an input (BIDIROE = 0) or an output (BIDIROE = 1). If SPC0 = 1 and master mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

12.1.4 $\overline{SS}$ — Slave Select

When the SPI is enabled as a slave, this pin is the low-true slave select input. When the SPI is enabled as a master and mode fault enable is off (MODFEN = 0), this pin is not used by the SPI and reverts to being a general-purpose port I/O pin. When the SPI is enabled as a master and MODFEN = 1, the slave select output enable bit determines whether this pin acts as the mode fault input (SSOE = 0) or as the slave select output (SSOE = 1).

12.4.2 SPI Interrupts

There are three flag bits, two interrupt mask bits, and one interrupt vector associated with the SPI system. The SPI interrupt enable mask (SPIE) enables interrupts from the SPI receiver full flag (SPRF) and mode fault flag (MODF). The SPI transmit interrupt enable mask (SPTIE) enables interrupts from the SPI transmit buffer empty flag (SPTEF). When one of the flag bits is set, and the associated interrupt mask bit is set, a hardware interrupt request is sent to the CPU. If the interrupt mask bits are cleared, software can poll the associated flag bits instead of using interrupts. The SPI interrupt service routine (ISR) should check the flag bits to determine what event caused the interrupt. The service routine should also clear the flag bit(s) before returning from the ISR (usually near the beginning of the ISR).

12.4.3 Mode Fault Detection

A mode fault occurs and the mode fault flag (MODF) becomes set when a master SPI device detects an error on the $\overline{SS}$ pin (provided the $\overline{SS}$ pin is configured as the mode fault input signal). The $\overline{SS}$ pin is configured to be the mode fault input signal when MSTR = 1, mode fault enable is set (MODFEN = 1), and slave select output enable is clear (SSOE = 0).

The mode fault detection feature can be used in a system where more than one SPI device might become a master at the same time. The error is detected when a master's $\overline{SS}$ pin is low, indicating that some other SPI device is trying to address this master as if it were a slave. This could indicate a harmful output driver conflict, so the mode fault logic is designed to disable all SPI output drivers when such an error is detected.

When a mode fault is detected, MODF is set and MSTR is cleared to change the SPI configuration back to slave mode. The output drivers on the SPSCK, MOSI, and MISO (if not bidirectional mode) are disabled.

MODF is cleared by reading it while it is set, then writing to the SPI control register 1 (SPI1C1). User software should verify the error condition has been corrected before changing the SPI back to master mode.

Table 13-2. IIC1A Register Field Descriptions

Field	Description
7:6 MULT	**IIC Multiplier Factor** — The MULT bits define the multiplier factor mul. This factor is used along with the SCL divider to generate the IIC baud rate. The multiplier factor mul as defined by the MULT bits is provided below. 00 mul = 01 01 mul = 02 10 mul = 04 11 Reserved
5:0 ICR	**IIC Clock Rate** — The ICR bits are used to prescale the bus clock for bit rate selection. These bits are used to define the SCL divider and the SDA hold value. The SCL divider multiplied by the value provided by the MULT register (multiplier factor mul) is used to generate IIC baud rate. IIC baud rate = bus speed (Hz)/(mul * SCL divider) SDA hold time is the delay from the falling edge of the SCL (IIC clock) to the changing of SDA (IIC data). The ICR is used to determine the SDA hold value. SDA hold time = bus period (s) * SDA hold value Table 13-3 provides the SCL divider and SDA hold values for corresponding values of the ICR. These values can be used to set IIC baud rate and SDA hold time. For example: Bus speed = 8 MHz MULT is set to 01 (mul = 2) Desired IIC baud rate = 100 kbps IIC baud rate = bus speed (Hz)/(mul * SCL divider) 100000 = 8000000/(2*SCL divider) SCL divider = 40 Table 13-3 shows that ICR must be set to 0B to provide an SCL divider of 40 and that this will result in an SDA hold value of 9. SDA hold time = bus period (s) * SDA hold value SDA hold time = 1/8000000 * 9 = 1.125 µs If the generated SDA hold value is not acceptable, the MULT bits can be used to change the ICR. This will result in a different SDA hold value.

13.3.5 IIC Data I/O Register (IIC1D)

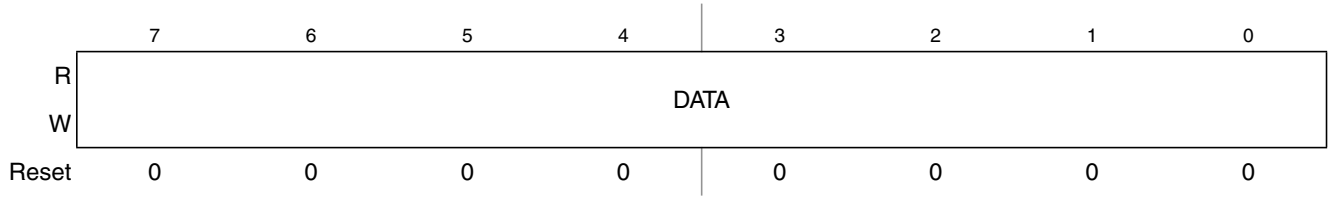


Figure 13-7. IIC Data I/O Register (IIC1D)

Table 13-6. IIC1D Register Field Descriptions

Field	Description
7:0 DATA	**Data** — In master transmit mode, when data is written to the IIC1D, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data.

NOTE

When transmitting out of master receive mode, the IIC mode should be switched before reading the IIC1D register to prevent an inadvertent initiation of a master receive data transfer.

In slave mode, the same functions are available after an address match has occurred.

Note that the TX bit in IIC1C must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, then reading the IIC1D will not initiate the receive.

Reading the IIC1D will return the last byte received while the IIC is configured in either master receive or slave receive modes. The IIC1D does not reflect every byte that is transmitted on the IIC bus, nor can software verify that a byte has been written to the IIC1D correctly by reading it back.

In master transmit mode, the first byte of data written to IIC1D following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7–bit 1) concatenated with the required R/W bit (in position bit 0).

converter yields the lower code (and vice-versa). However, even very small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is normally around 1/2LSB and will increase with noise. This error may be reduced by repeatedly sampling the input and averaging the result. Additionally the techniques discussed in Section 14.7.2.3 will reduce this error.

Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Missing codes are those values which are never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and to have no missing codes.

Figure 15-2 shows an external host transmitting a logic 1 or 0 to the BKGD pin of a target HCS08 MCU. The host is asynchronous to the target so there is a 0-to-1 cycle delay from the host-generated falling edge to where the target perceives the beginning of the bit time. Ten target BDC clock cycles later, the target senses the bit level on the BKGD pin. Typically, the host actively drives the pseudo-open-drain BKGD pin during host-to-target transmissions to speed up rising edges. Because the target does not drive the BKGD pin during the host-to-target transmission period, there is no need to treat the line as an open-drain signal during this period.

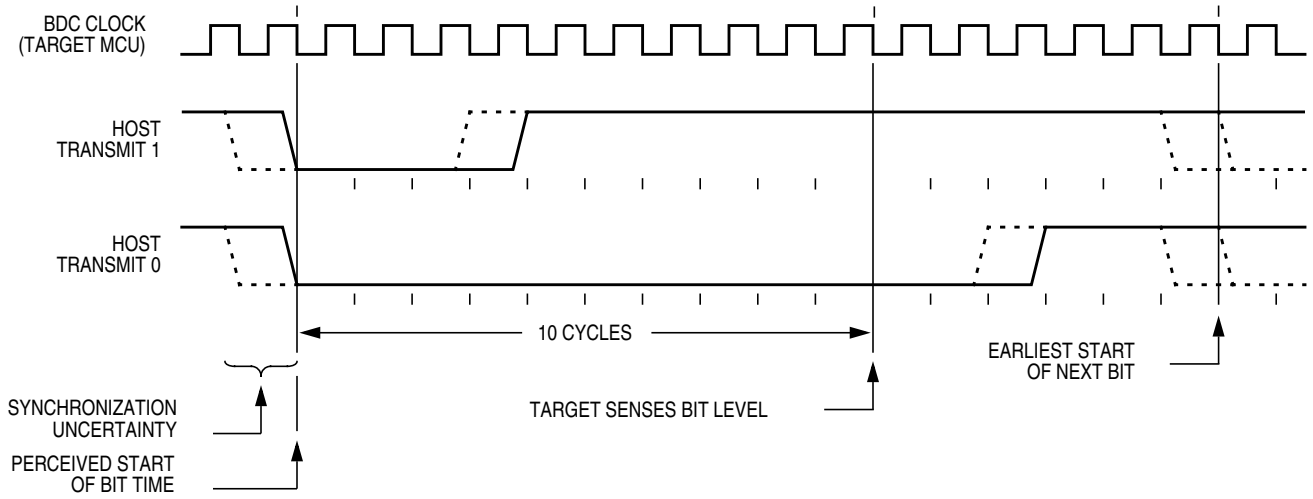


Figure 15-2. BDC Host-to-Target Serial Bit Timing

15.4.3.8 Debug Trigger Register (DBGT)

This register can be read any time, but may be written only if ARM = 0, except bits 4 and 5 are hard-wired to 0s.

	7	6	5	4	3	2	1	0
R	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-8. Debug Trigger Register (DBGT)

Table 15-5. DBGT Register Field Descriptions

Field	Description
7 TRGSEL	**Trigger Type** — Controls whether the match outputs from comparators A and B are qualified with the opcode tracking logic in the debug module. If TRGSEL is set, a match signal from comparator A or B must propagate through the opcode tracking logic and a trigger event is only signalled to the FIFO logic if the opcode at the match address is actually executed. 0 Trigger on access to compare address (force) 1 Trigger if opcode at compare address is executed (tag)
6 BEGIN	**Begin/End Trigger Select** — Controls whether the FIFO starts filling at a trigger or fills in a circular manner until a trigger ends the capture of information. In event-only trigger modes, this bit is ignored and all debug runs are assumed to be begin traces. 0 Data stored in FIFO until trigger (end trace) 1 Trigger initiates data storage (begin trace)
3:0 TRG[3:0]	**Select Trigger Mode** — Selects one of nine triggering modes, as described below. 0000 A-only 0001 A OR B 0010 A Then B 0011 Event-only B (store data) 0100 A then event-only B (store data) 0101 A AND B data (full mode) 0110 A AND NOT B data (full mode) 0111 Inside range: A ≤ address ≤ B 1000 Outside range: address < A or address > B 1001 – 1111 (No trigger)

Table A-10. 10-bit ADC Characteristics (V_{REFH} = V_{DDAD}, V_{REFL} = V_{SSAD}) (continued)

Characteristic	Conditions	C	Symb	Min	Typ[1]	Max	Unit
ADC asynchronous clock source t_{ADACK} = $1/f_{ADACK}$	High speed (ADLPC = 0)	P	f_{ADACK}	2	3.3	5	MHzS
	Low power (ADLPC = 1)			1.25	2	3.3	
Conversion time (Including sample time)	Short sample (ADLSMP = 0)	P	t_{ADC}	—	20	—	ADCK cycles
	Long sample (ADLSMP = 1)			—	40	—	
Sample time	Short sample (ADLSMP = 0)	P	t_{ADS}	—	3.5	—	ADCK cycles
	Long sample (ADLSMP = 1)			—	23.5	—	
Total unadjusted error Includes quantization	10-bit mode	P	E_{TUE}	—	±1	±2.5	LSB[2]
	8-bit mode			—	±0.5	±1.0	
Differential non-linearity	10-bit mode	P	DNL	—	±0.5	±1.0	LSB[2]
	8-bit mode			—	±0.3	±0.5	
	Monotonicity and no-missing-codes guaranteed						
Integral non-linearity	10-bit mode	C	INL	—	±0.5	±1.0	LSB[2]
	8-bit mode			—	±0.3	±0.5	
Zero-scale error V_{ADIN} = V_{SSA}	10-bit mode	P	E_{ZS}	—	±0.5	±1.5	LSB[2]
	8-bit mode			—	±0.5	±0.5	
Full-scale error V_{ADIN} = V_{DDA}	10-bit mode	P	E_{FS}	—	±0.5	±1.5	LSB[2]
	8-bit mode			—	±0.5	±0.5	
Quantization error	10-bit mode	D	E_Q	—	—	±0.5	LSB[2]
	8-bit mode			—	—	±0.5	
Input leakage error Pad leakage[3] * R_{AS}	10-bit mode	D	E_{IL}	—	±0.2	±2.5	LSB[2]
	8-bit mode			—	±0.1	±1	
Temp sensor slope	-40°C - 25°C	T	m	—	3.266	—	mV/°C
	25°C - 125°C			—	3.638	—	
Temp sensor voltage	25°C	P	V_{TEMP25}	—	1.396	—	V

1 Typical values assume V_{DDAD} = 5.0V, Temp = 25C, f_{ADCK}=1.0 MHz unless otherwise stated. Typical values are for reference only and are not tested in production.

2 1 LSB = $(V_{REFH} - V_{REFL})/2^N$

3 Based on input pad leakage current. Refer to pad electricals.

A.10 AC Characteristics

This section describes ac timing characteristics for each peripheral system. For detailed information about how clocks for the bus are generated, see Chapter 8, "Internal Clock Generator (S08ICGV4)."

A.10.1 Control Timing

Table A-13. Control Timing

Num	C	Parameter	Symbol	Min	Typ[1]	Max	Unit
1		Bus frequency ($t_{cyc} = 1/f_{Bus}$)	f_{Bus}	dc	—	20	MHz
2	P	Real-time interrupt internal oscillator period	t_{RTI}	700		1300	µs
3		External reset pulse width[2] ($t_{cyc} = 1/f_{Self\_reset}$)	t_{extrst}	1.5 x t_{Self_reset}		—	ns
4		Reset low drive[3]	t_{rstdrv}	34 x t_{cyc}		—	ns
5		Active background debug mode latch setup time	t_{MSSU}	25		—	ns
6		Active background debug mode latch hold time	t_{MSH}	25		—	ns
7		IRQ pulse width Asynchronous path[2] Synchronous path[4]	t_{ILIH}, t_{IHIL}	 100 1.5 x t_{cyc}	—	—	ns
8		KBIPx pulse width Asynchronous path[2] Synchronous path[3]	t_{ILIH}, t_{IHIL}	 100 1.5 x t_{cyc}	—	—	ns
9	T	Port rise and fall time — Low output drive (PTxDS = 0) (load = 50 pF)[5] Slew rate control disabled (PTxSE = 0) Slew rate control enabled (PTxSE = 1)	t_{Rise}, t_{Fall}	 — —	 40 75	 — —	ns
		Port rise and fall time — High output drive (PTxDS = 1) (load = 50 pF) Slew rate control disabled (PTxSE = 0) Slew rate control enabled (PTxSE = 1)	t_{Rise}, t_{Fall}	 — —	 11 35	 — —	ns

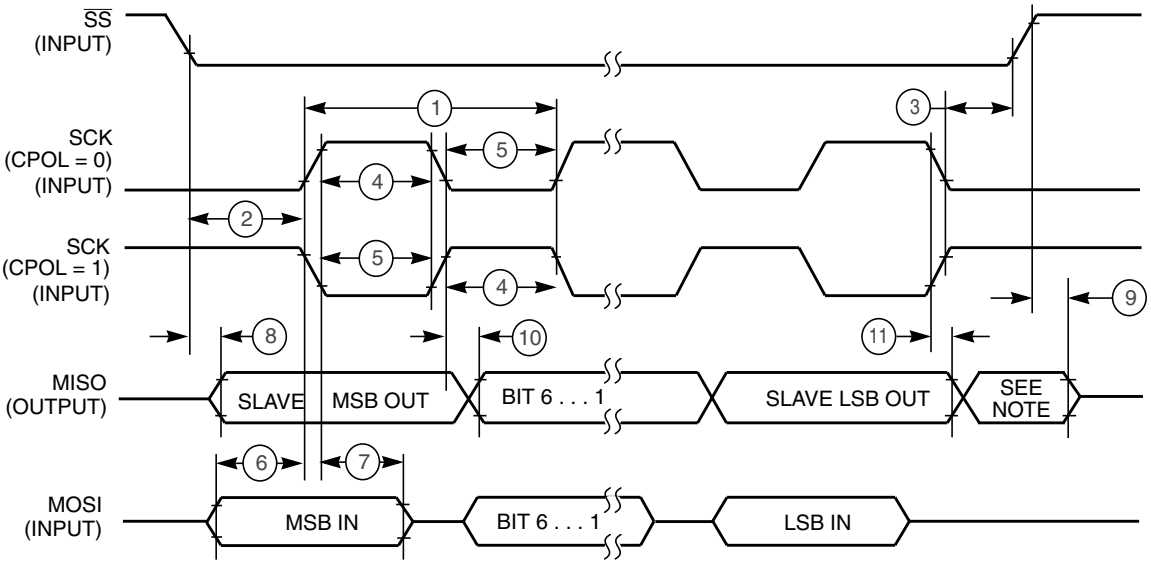
1 Typical values are based on characterization data at V_{DD} = 5.0V, 25°C unless otherwise stated.

2 This is the shortest pulse that is guaranteed to be recognized as a reset pin request. Shorter pulses are not guaranteed to override reset requests from internal sources.

3 When any reset is initiated, internal circuitry drives the reset pin low for about 34 bus cycles and then samples the level on the reset pin about 38 bus cycles later to distinguish external reset requests from internal requests.

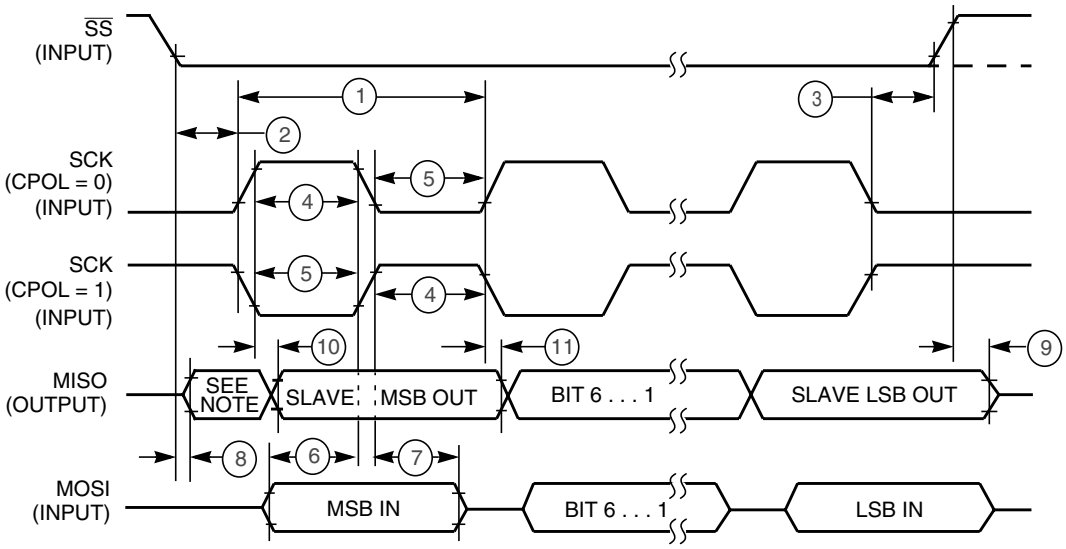
4 This is the minimum pulse width that is guaranteed to pass through the pin synchronization circuitry. Shorter pulses may or may not be recognized. In stop mode, the synchronizer is bypassed so shorter pulses can be recognized in that case.

5 Timing is shown with respect to 20% V_{DD} and 80% V_{DD} levels. Temperature range –40°C to 125°C.



NOTE:
1. Not defined but normally MSB of character just received

Figure A-18. SPI Slave Timing (CPHA = 0)



NOTE:
1. Not defined but normally LSB of character just received

Figure A-19. SPI Slave Timing (CPHA = 1)