

Welcome to [E-XFL.COM](http://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	I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	12
Program Memory Size	32KB (32K 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg32e1mtg">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg32e1mtg</a>

# Contents

Section Number	Title	Page
<b>Chapter 1</b>		
<b>Device Overview</b>		
1.1	Devices in the MC9S08SG32 Series .....	21
1.2	MCU Block Diagram .....	22
1.3	System Clock Distribution .....	24
<b>Chapter 2</b>		
<b>Pins and Connections</b>		
2.1	Device Pin Assignment .....	25
2.2	Recommended System Connections .....	27
2.2.1	Power .....	27
2.2.2	Oscillator (XOSC) .....	28
2.2.3	$\overline{\text{RESET}}$ .....	28
2.2.4	Background / Mode Select (BKGD/MS) .....	29
2.2.5	General-Purpose I/O and Peripheral Ports .....	29
<b>Chapter 3</b>		
<b>Modes of Operation</b>		
3.1	Introduction .....	33
3.2	Features .....	33
3.3	Run Mode .....	33
3.4	Active Background Mode .....	33
3.5	Wait Mode .....	34
3.6	Stop Modes .....	34
3.6.1	Stop3 Mode .....	35
3.6.2	Stop2 Mode .....	36
3.6.3	On-Chip Peripheral Modules in Stop Modes .....	36
<b>Chapter 4</b>		
<b>Memory</b>		
4.1	MC9S08SG32 Series Memory Map .....	39
4.2	Reset and Interrupt Vector Assignments .....	40
4.3	Register Addresses and Bit Assignments .....	41
4.4	RAM .....	48
4.5	FLASH .....	48
4.5.1	Features .....	49
4.5.2	Program and Erase Times .....	49
4.5.3	Program and Erase Command Execution .....	50

# Chapter 1

## Device Overview

The MC9S08SG32 devices are members of the low-cost, high-performance HCS08 family of 8-bit microcontroller units (MCUs). The MC9S08SG32 Series high-temperature devices have been qualified to meet or exceed AEC Grade 0 requirements to allow them to operate up to 150 °C T<sub>A</sub>. All MCUs in the family use the enhanced HCS08 core and are available with a variety of modules, memory sizes, memory types, and package types.

### 1.1 Devices in the MC9S08SG32 Series

Table 1-1 summarizes the feature set available in the MC9S08SG32 series of MCUs.

**Table 1-1. MC9S08SG32 Series Features by MCU and Package**

Feature	MC9S08SG32			MC9S08SG16		
FLASH size (bytes)	32768			16384		
RAM size (bytes)	1024			1024		
Pin quantity	28	20	16	28	20	16
ACMP	yes			yes		
ADC channels	16	12	8	16	12	8
DBG	yes			yes		
ICS	yes			yes		
IIC	yes			yes		
MTIM	yes			yes		
Pin Interrupts	8			8		
Pin I/O	22	16	12	22	16	12
RTC	yes			yes		
SCI	yes			yes		
SPI	yes			yes		
TPM1 channels	yes			yes		
TPM2 channels	yes			yes		
XOSC	yes			yes		



### 4.5.5 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. Before any command can be processed, write a 1 to FACCERR in FSTAT to clear the access error flag (FACCERR).

- Writing to a FLASH address before the internal FLASH clock frequency has been set by writing to the FCDIV register
- Writing to a FLASH address while FCBEF is not set (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address
- Writing any command code other than the five allowed codes (0x05, 0x20, 0x25, 0x40, or 0x41) to FCMD
- Writing any FLASH control register other than the write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD
- The MCU enters stop mode while a program or erase command is in progress (The command is aborted.)
- Writing the byte program, burst program, or page erase command code (0x20, 0x25, or 0x40) with a background debug command while the MCU is secured (The background debug controller can only do blank check and mass erase commands when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command

### 4.5.6 FLASH Block Protection

The block protection feature prevents the protected region of FLASH from program or erase changes. Block protection is controlled through the FLASH protection register (FPROT). When enabled, block protection begins at any 512 byte boundary below the last address of FLASH, 0xFFFF. (See [Section 4.7.4, “FLASH Protection Register \(FPROT and NVPROT\)”](#)).

After exit from reset, FPROT is loaded with the contents of the NVPROT location, which is in the nonvolatile register block of the FLASH memory. FPROT cannot be changed directly from application software so a runaway program cannot alter the block protection settings. Because NVPROT is within the last 512 bytes of FLASH, if any amount of memory is protected, NVPROT is itself protected and cannot be altered (intentionally or unintentionally) by the application software. FPROT can be written through background debug commands, which allows a way to erase and reprogram a protected FLASH memory.

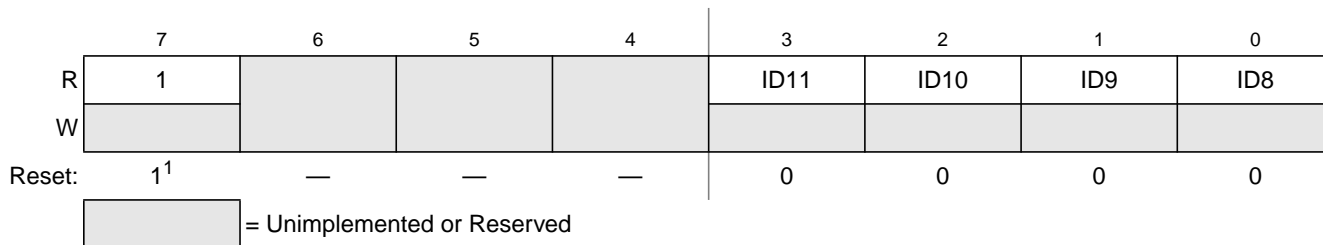
The block protection mechanism is illustrated in [Figure 4-4](#). The FPS bits are used as the upper bits of the last address of unprotected memory. This address is formed by concatenating FPS7:FPS1 with logic 1 bits as shown. For example, to protect the last 1536 bytes of memory (addresses 0xFA00 through 0xFFFF), the FPS bits must be set to 1111 100, which results in the value 0xF9FF as the last address of unprotected

Table 5-2. Vector Summary

Vector Priority	Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description
Lowest  Highest	31	0xFFC0/0xFFC1	—	—	—	—	—
	30	0xFFC2/0xFFC3	Vacmp	ACMP	ACF	ACIE	Analog comparator
	29	0xFFC4/0xFFC5	—	—	—	—	—
	28	0xFFC6/0xFFC7	—	—	—	—	—
	27	0xFFC8/0xFFC9	—	—	—	—	—
	26	0xFFCA/0xFFCB	Vmtim	MTIM	TOF	TOIE	MTIM overflow
	25	0xFFCC/0xFFCD	Vrtc	RTC	RTIF	RTIE	Real-time interrupt
	24	0xFFCE/0xFFCF	Viic	IIC	IICIS	IICIE	IIC control
	23	0xFFD0/0xFFD1	Vadc	ADC	COCO	AIEN	ADC
	22	0xFFD2/0xFFD3	—	—	—	—	—
	21	0xFFD4/0xFFD5	Vportb	Port B	PTBIF	PTBIE	Port B Pins
	20	0xFFD6/0xFFD7	Vporta	Port A	PTAIF	PTAIE	Port A Pins
	19	0xFFD8/0xFFD9	—	—	—	—	—
	18	0xFFDA/0xFFDB	Vscitx	SCI	TDRE, TC	TIE, TCIE	SCI transmit
	17	0xFFDC/0xFFDD	Vscirx	SCI	IDLE, RDRF, LBDKIDIF, RXEDGIF	ILIE, RIE, LBKDIE, RXEDGIE	SCI receive
	16	0xFFDE/0xFFDF	Vscierr	SCI	OR, NF, FE, PF	ORIE, NFIE, FEIE, PFIE	SCI error
	15	0xFFE0/0xFFE1	Vspi	SPI	SPIF, MODF, SPTEF	SPIE, SPIE, SPTIE	SPI
	14	0xFFE2/0xFFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
	13	0xFFE4/0xFFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
	12	0xFFE6/0xFFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
	11	0xFFE8/0xFFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
	10	0xFFEA/0xFFEB	—	—	—	—	—
	9	0xFFEC/0xFFED	—	—	—	—	—
	8	0xFFEE/0xFFEF	—	—	—	—	—
	7	0xFFFF0/0xFFFF1	—	—	—	—	—
	6	0xFFFF2/0xFFFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1
	5	0xFFFF4/0xFFFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0
	4	0xFFFF6/0xFFFF7	—	—	—	—	—
	3	0xFFFF8/0xFFFF9	Vlvd	System control	LVWF	LVWIE	Low-voltage warning
	2	0xFFFFA/0xFFFFB	—	—	—	—	—
	1	0xFFFFC/0xFFFFD	Vswi	Core	SWI Instruction	—	Software interrupt
0	0xFFFFE/0xFFFFF	Vreset	System control	COP, LVD, RESET pin, Illegal opcode, Illegal address	COPE, LVDRE, —, —, —	Watchdog timer Low-voltage detect External pin Illegal opcode Illegal address	

### 5.7.5 System Device Identification Register (SDIDH, SDIDL)

These high page read-only registers are included so host development systems can identify the HCS08 derivative and revision number. This allows the development software to recognize where specific memory blocks, registers, and control bits are located in a target MCU.

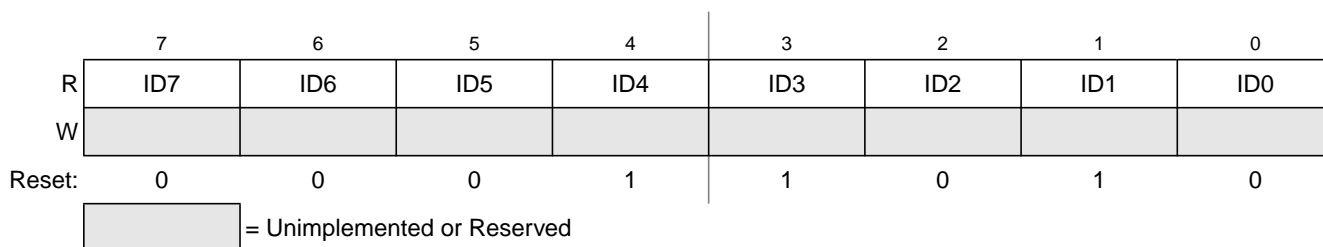


<sup>1</sup> - Bit 7 is a mask option tie off that is used internally to determine that the device is a MC9S08SG32 Series.

**Figure 5-6. System Device Identification Register — High (SDIDH)**

**Table 5-7. SDIDH Register Field Descriptions**

Field	Description
7	Bit 7 will read as a 1 for the MC9S08SG32 Series devices; writes have no effect.
6:4 Reserved	Bits 6:4 are reserved. Reading these bits will result in an indeterminate value; writes have no effect.
3:0 ID[11:8]	<b>Part Identification Number</b> — Each derivative in the HCS08 Family has a unique identification number. The MC9S08SG32 is hard coded to the value 0x01A. See also ID bits in <a href="#">Table 5-8</a> .



**Figure 5-7. System Device Identification Register — Low (SDIDL)**

**Table 5-8. SDIDL Register Field Descriptions**

Field	Description
7:0 ID[7:0]	<b>Part Identification Number</b> — Each derivative in the HCS08 Family has a unique identification number. The MC9S08SG32 is hard coded to the value 0x01A. See also ID bits in <a href="#">Table 5-7</a> .

**Table 9-12. APCTL3 Register Field Descriptions (continued)**

Field	Description
1 ADPC17	<b>ADC Pin Control 17</b> — ADPC17 controls the pin associated with channel AD17. 0 AD17 pin I/O control enabled 1 AD17 pin I/O control disabled
0 ADPC16	<b>ADC Pin Control 16</b> — ADPC16 controls the pin associated with channel AD16. 0 AD16 pin I/O control enabled 1 AD16 pin I/O control disabled

## 9.4 Functional Description

The ADC module is disabled during reset or when the ADCH bits are all high. The module is idle when a conversion has completed and another conversion has not been initiated. When idle, the module is in its lowest power state.

The ADC can perform an analog-to-digital conversion on any of the software selectable channels. The selected channel voltage is converted by a successive approximation algorithm into an 11-bit digital result. In 8-bit mode, the selected channel voltage is converted by a successive approximation algorithm into a 9-bit digital result.

When the conversion is completed, the result is placed in the data registers (ADCRH and ADCRL). In 10-bit mode, the result is rounded to 10 bits and placed in ADCRH and ADCRL. In 8-bit mode, the result is rounded to 8 bits and placed in ADCRL. The conversion complete flag (COCO) is then set and an interrupt is generated if the conversion complete interrupt has been enabled (AIEN = 1).

The ADC module has the capability of automatically comparing the result of a conversion with the contents of its compare registers. The compare function is enabled by setting the ACFE bit and operates with any of the conversion modes and configurations.

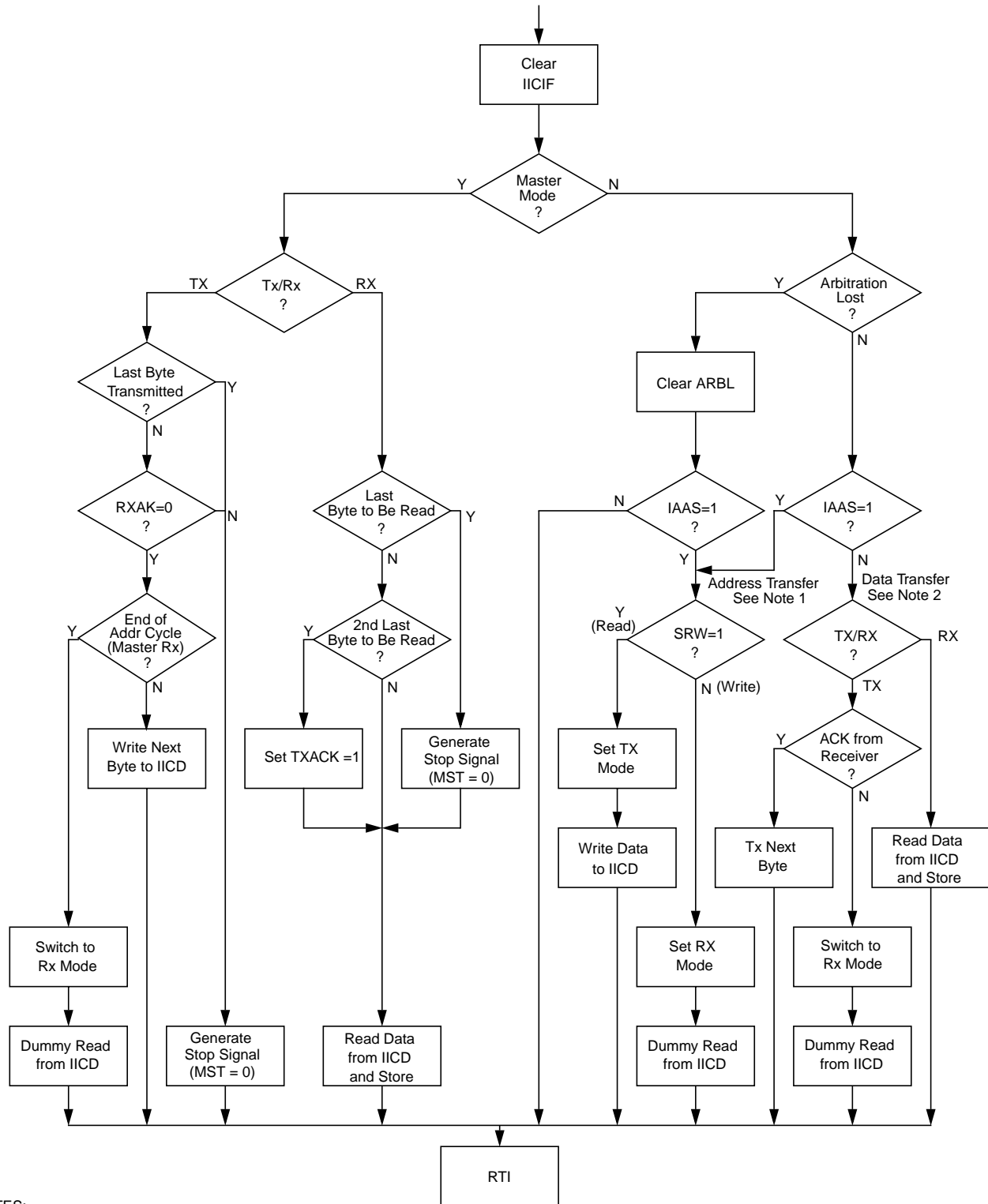
### 9.4.1 Clock Select and Divide Control

One of four clock sources can be selected as the clock source for the ADC module. This clock source is then divided by a configurable value to generate the input clock to the converter (ADCK). The clock is selected from one of the following sources by means of the ADICLK bits.

- The bus clock, which is equal to the frequency at which software is executed. This is the default selection following reset.
- The bus clock divided by two. For higher bus clock rates, this allows a maximum divide by 16 of the bus clock.
- ALTCLK, as defined for this MCU (See module section introduction).
- The asynchronous clock (ADACK). This clock is generated from a clock source within the ADC module. When selected as the clock source, this clock remains active while the MCU is in wait or stop3 mode and allows conversions in these modes for lower noise operation.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC does not perform according to specifications. If the available clocks

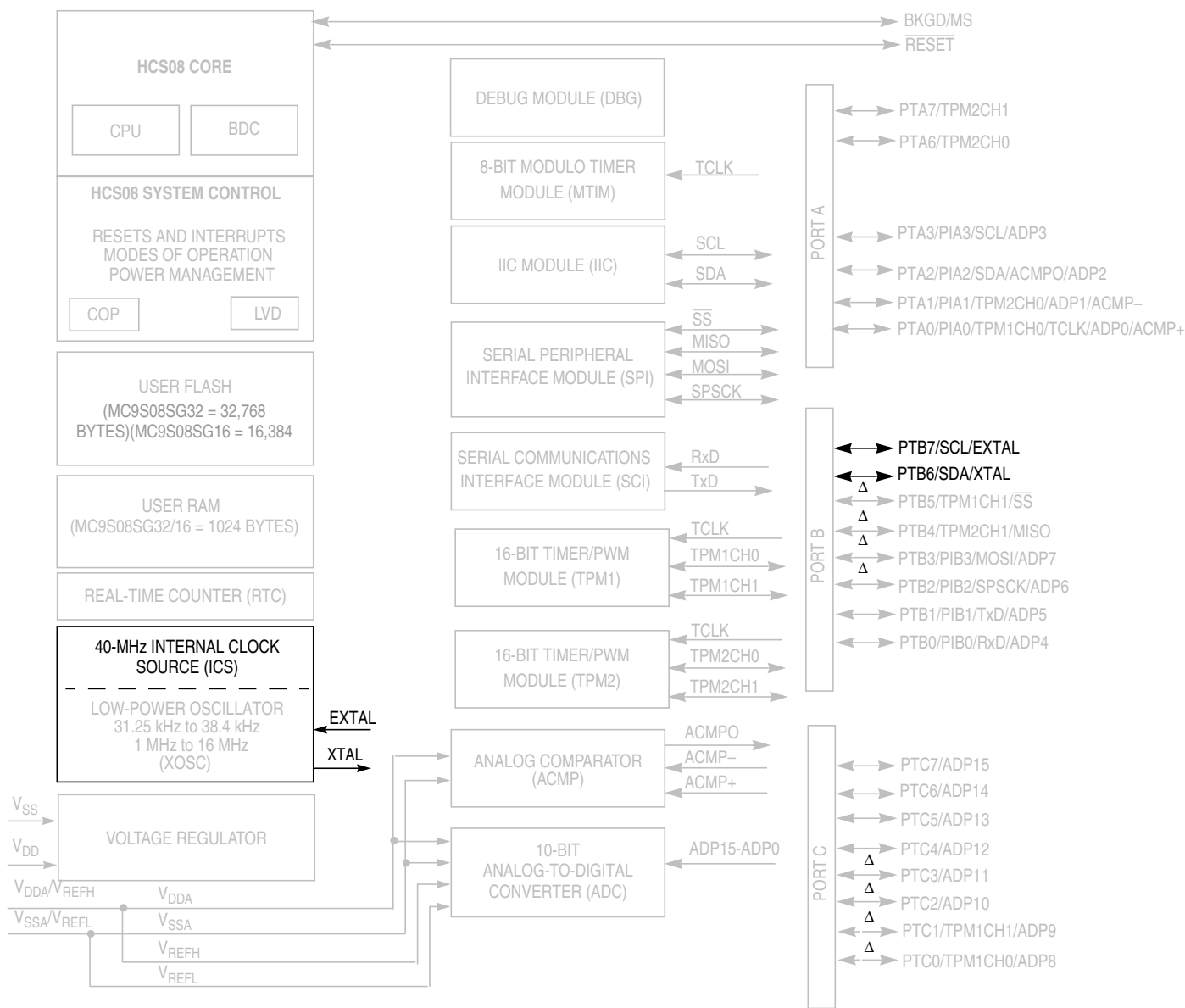




NOTES:

1. If general call is enabled, a check must be done to determine whether the received address was a general call address (0x00). If the received address was a general call address, then the general call must be handled by user software.
2. When 10-bit addressing is used to address a slave, the slave sees an interrupt following the first byte of the extended address. User software must ensure that for this interrupt, the contents of IICD are ignored and not treated as a valid data transfer

Figure 10-12. Typical IIC Interrupt Routine



NOTE

- PTC7-PTC0 and PTA7-PTA6 are not available on 16-pin Packages
  - PTC7-PTC4 and PTA7-PTA6 are not available on 20-pin Packages
  - For the 16-pin and 20-pin packages:  $V_{DDA}/V_{REFH}$  and  $V_{SSA}/V_{REFL}$  are double bonded to  $V_{DD}$  and  $V_{SS}$  respectively.
- $\Delta$  = Pin can be enabled as part of the ganged output drive feature

Figure 11-1. MC9S08SG32 Series Block Diagram Highlighting ICS Block and Pins

The CLKS bits can also be changed at anytime, but the RDIV bits must be changed simultaneously so that the resulting frequency stays in the range of 31.25 kHz to 39.0625 kHz. The actual switch to the newly selected clock will not occur until after a few full cycles of the new clock. If the newly selected clock is not available, the previous clock will remain selected.

### 11.4.3 Bus Frequency Divider

The BDIV bits can be changed at anytime and the actual switch to the new frequency will occur immediately.

### 11.4.4 Low Power Bit Usage

The low power bit (LP) is provided to allow the FLL to be disabled and thus conserve power when it is not being used. However, in some applications it may be desirable to enable the FLL and allow it to lock for maximum accuracy before switching to an FLL engaged mode. Do this by writing the LP bit to 0.

### 11.4.5 Internal Reference Clock

When IRCLKEN is set the internal reference clock signal will be presented as ICSIRCLK, which can be used as an additional clock source. The ICSIRCLK frequency can be re-targeted by trimming the period of the internal reference clock. This can be done by writing a new value to the TRIM bits in the ICSTRM register. Writing a larger value will slow down the ICSIRCLK frequency, and writing a smaller value to the ICSTRM register will speed up the ICSIRCLK frequency. The TRIM bits will effect the ICSOUT frequency if the ICS is in FLL engaged internal (FEI), FLL bypassed internal (FBI), or FLL bypassed internal low power (FBILP) mode. The TRIM and FTRIM value will not be affected by a reset.

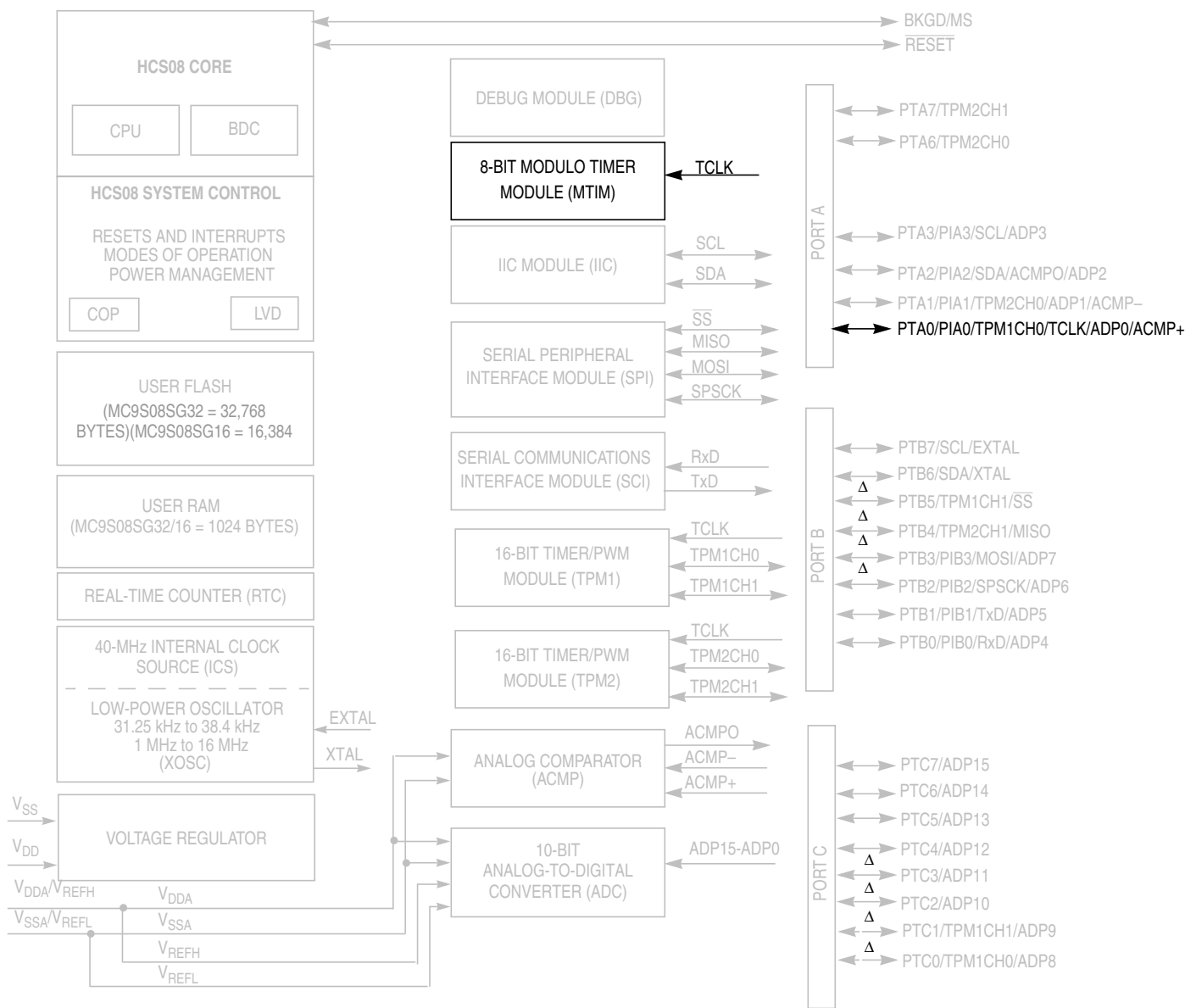
Until ICSIRCLK is trimmed, programming low reference divider (RDIV) factors may result in ICSOUT frequencies that exceed the maximum chip-level frequency and violate the chip-level clock timing specifications (see the [Device Overview](#) chapter).

If IREFSTEN is set and the IRCLKEN bit is written to 1, the internal reference clock will keep running during stop mode in order to provide a fast recovery upon exiting stop.

All MCU devices are factory programmed with a trim value in a reserved memory location (NVTRIM:NVFTRIM). This value can be copied to the ICSTRM register during reset initialization. The factory trim value includes the FTRIM bit. For finer precision, the user can trim the internal oscillator in the application to take in account small differences between the factory test setup and actual application conditions.

### 11.4.6 Optional External Reference Clock

The ICS module can support an external reference clock with frequencies between 31.25 kHz to 5 MHz in all modes. When the ERCLKEN is set, the external reference clock signal will be presented as ICSECLK, which can be used as an additional clock source. When IREFS = 1, the external reference clock will not be used by the FLL and will only be used as ICSECLK. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support (see the [Device Overview](#) chapter).



**NOTE**

- PTC7-PTC0 and PTA7-PTA6 are not available on 16-pin Packages
  - PTC7-PTC4 and PTA7-PTA6 are not available on 20-pin Packages
  - For the 16-pin and 20-pin packages:  $V_{DDA}/V_{REFH}$  and  $V_{SSA}/V_{REFL}$  are double bonded to  $V_{DD}$  and  $V_{SS}$  respectively.
- $\Delta$  = Pin can be enabled as part of the ganged output drive feature

**Figure 12-1. MC9S08SG32 Series Block Diagram Highlighting MTIM Block and Pins**

# Chapter 14

## Serial Communications Interface (S08SCIV4)

### 14.1 Introduction

[Figure 14-1](#) shows the MC9S08SG32 Series block diagram with the SCI module highlighted.

**Table 14-5. SCIS1 Field Descriptions**

Field	Description
7 TDRE	<p><b>Transmit Data Register Empty Flag</b> — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIS1 with TDRE = 1 and then write to the SCI data register (SCID).</p> <p>0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.</p>
6 TC	<p><b>Transmission Complete Flag</b> — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted.</p> <p>0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).</p> <p>TC is cleared automatically by reading SCIS1 with TC = 1 and then doing one of the following three things:</p> <ul style="list-style-type: none"> <li>• Write to the SCI data register (SCID) to transmit new data</li> <li>• Queue a preamble by changing TE from 0 to 1</li> <li>• Queue a break character by writing 1 to SBK in SCIC2</li> </ul>
5 RDRF	<p><b>Receive Data Register Full Flag</b> — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCID). To clear RDRF, read SCIS1 with RDRF = 1 and then read the SCI data register (SCID).</p> <p>0 Receive data register empty. 1 Receive data register full.</p>
4 IDLE	<p><b>Idle Line Flag</b> — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line.</p> <p>To clear IDLE, read SCIS1 with IDLE = 1 and then read the SCI data register (SCID). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period.</p> <p>0 No idle line detected. 1 Idle line was detected.</p>
3 OR	<p><b>Receiver Overrun Flag</b> — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCID yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCID. To clear OR, read SCIS1 with OR = 1 and then read the SCI data register (SCID).</p> <p>0 No overrun. 1 Receive overrun (new SCI data lost).</p>
2 NF	<p><b>Noise Flag</b> — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIS1 and then read the SCI data register (SCID).</p> <p>0 No noise detected. 1 Noise detected in the received character in SCID.</p>

**Table 14-7. SCIC3 Field Descriptions (continued)**

Field	Description
4 TXINV <sup>1</sup>	<b>Transmit Data Inversion</b> — Setting this bit reverses the polarity of the transmitted data output. 0 Transmit data not inverted 1 Transmit data inverted
3 ORIE	<b>Overrun Interrupt Enable</b> — This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled (use polling). 1 Hardware interrupt requested when OR = 1.
2 NEIE	<b>Noise Error Interrupt Enable</b> — This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled (use polling). 1 Hardware interrupt requested when NF = 1.
1 FEIE	<b>Framing Error Interrupt Enable</b> — This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled (use polling). 1 Hardware interrupt requested when FE = 1.
0 PEIE	<b>Parity Error Interrupt Enable</b> — This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled (use polling). 1 Hardware interrupt requested when PF = 1.

<sup>1</sup> Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle.

## 14.2.7 SCI Data Register (SCID)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

**Figure 14-11. SCI Data Register (SCID)**

## 14.3 Functional Description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

### 14.3.1 Baud Rate Generation

As shown in [Figure 14-12](#), the clock source for the SCI baud rate generator is the bus-rate clock.

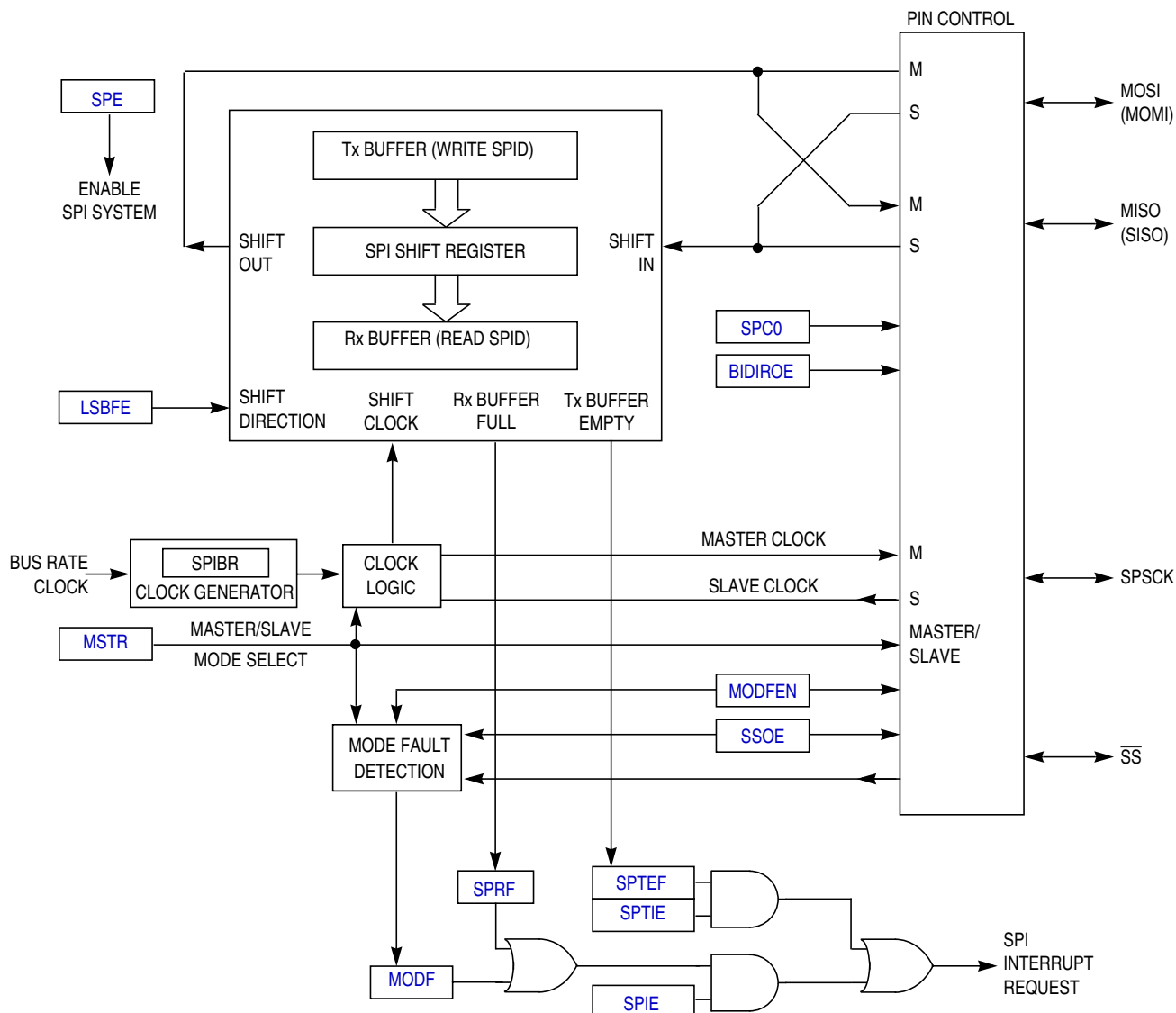


Figure 15-3. SPI Module Block Diagram

### 15.1.3 SPI Baud Rate Generation

As shown in [Figure 15-4](#), the clock source for the SPI baud rate generator is the bus clock. The three prescale bits (SPPR2:SPPR1:SPPR0) choose a prescale divisor of 1, 2, 3, 4, 5, 6, 7, or 8. The three rate select bits (SPR2:SPR1:SPR0) divide the output of the prescaler stage by 2, 4, 8, 16, 32, 64, 128, or 256 to get the internal SPI master mode bit-rate clock.



...

- configure the channel pin as output port pin and set the output pin;
- configure the channel to generate the EPWM signal but keep ELSnB:ELSnA as 00;
- configure the other registers (TPMxMODH, TPMxMODL, TPMxCnVH, TPMxCnVL, ...);
- configure CLKSB:CLKSA bits (TPM v3 starts to generate the high-true EPWM signal, however TPM does not control the channel pin, so the EPWM signal is not available);
- wait until the TOF is set (or use the TOF interrupt);
- enable the channel output by configuring ELSnB:ELSnA bits (now EPWM signal is available);

...

Figure 17-4 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

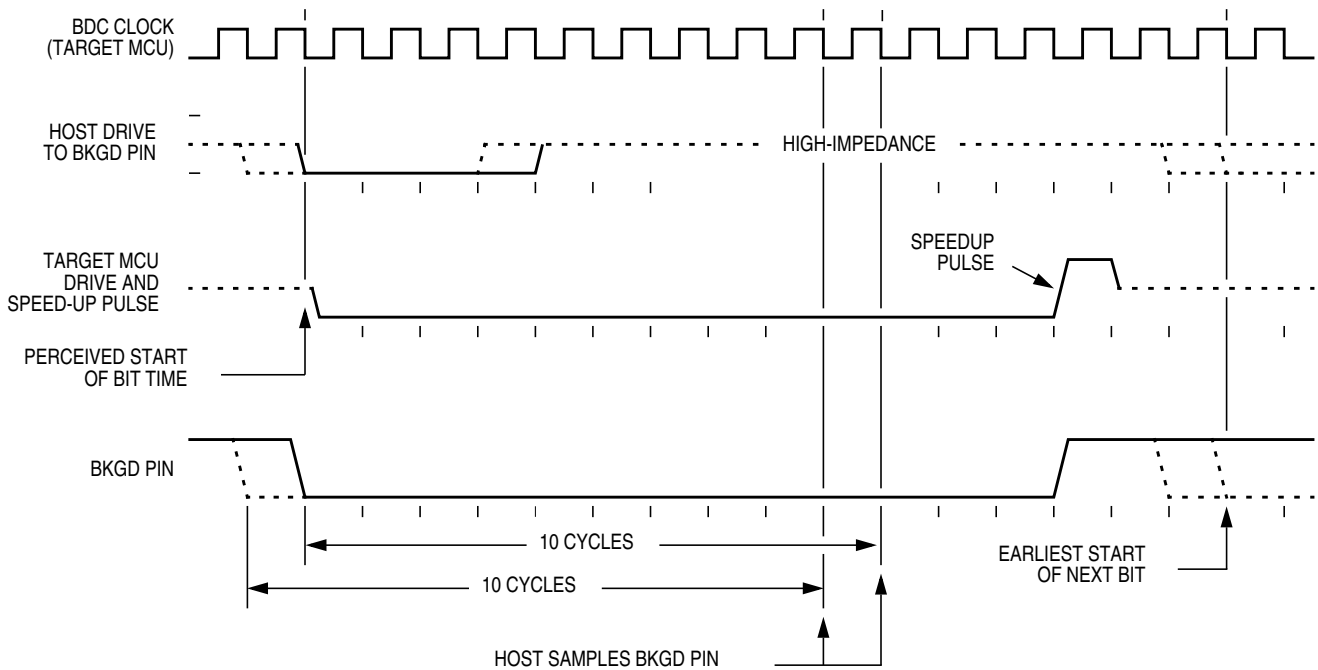


Figure 17-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

Table 17-1. BDC Command Summary

Command Mnemonic	Active BDM/ Non-intrusive	Coding Structure	Description
SYNC	Non-intrusive	n/a <sup>1</sup>	Request a timed reference pulse to determine target BDC communication speed
ACK_ENABLE	Non-intrusive	D5/d	Enable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
ACK_DISABLE	Non-intrusive	D6/d	Disable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
BACKGROUND	Non-intrusive	90/d	Enter active background mode if enabled (ignore if ENBDM bit equals 0)
READ_STATUS	Non-intrusive	E4/SS	Read BDC status from BDCSCR
WRITE_CONTROL	Non-intrusive	C4/CC	Write BDC controls in BDCSCR
READ_BYTE	Non-intrusive	E0/AAAA/d/RD	Read a byte from target memory
READ_BYTE_WS	Non-intrusive	E1/AAAA/d/SS/RD	Read a byte and report status
READ_LAST	Non-intrusive	E8/SS/RD	Re-read byte from address just read and report status
WRITE_BYTE	Non-intrusive	C0/AAAA/WD/d	Write a byte to target memory
WRITE_BYTE_WS	Non-intrusive	C1/AAAA/WD/d/SS	Write a byte and report status
READ_BKPT	Non-intrusive	E2/RBKP	Read BDCBKPT breakpoint register
WRITE_BKPT	Non-intrusive	C2/WBKP	Write BDCBKPT breakpoint register
GO	Active BDM	08/d	Go to execute the user application program starting at the address currently in the PC
TRACE1	Active BDM	10/d	Trace 1 user instruction at the address in the PC, then return to active background mode
TAGGO	Active BDM	18/d	Same as GO but enable external tagging (HCS08 devices have no external tagging pin)
READ_A	Active BDM	68/d/RD	Read accumulator (A)
READ_CCR	Active BDM	69/d/RD	Read condition code register (CCR)
READ_PC	Active BDM	6B/d/RD16	Read program counter (PC)
READ_HX	Active BDM	6C/d/RD16	Read H and X register pair (H:X)
READ_SP	Active BDM	6F/d/RD16	Read stack pointer (SP)
READ_NEXT	Active BDM	70/d/RD	Increment H:X by one then read memory byte located at H:X
READ_NEXT_WS	Active BDM	71/d/SS/RD	Increment H:X by one then read memory byte located at H:X. Report status and data.
WRITE_A	Active BDM	48/WD/d	Write accumulator (A)
WRITE_CCR	Active BDM	49/WD/d	Write condition code register (CCR)
WRITE_PC	Active BDM	4B/WD16/d	Write program counter (PC)
WRITE_HX	Active BDM	4C/WD16/d	Write H and X register pair (H:X)
WRITE_SP	Active BDM	4F/WD16/d	Write stack pointer (SP)
WRITE_NEXT	Active BDM	50/WD/d	Increment H:X by one, then write memory byte located at H:X
WRITE_NEXT_WS	Active BDM	51/WD/d/SS	Increment H:X by one, then write memory byte located at H:X. Also report status.

<sup>1</sup> The SYNC command is a special operation that does not have a command code.

## 17.3 On-Chip Debug System (DBG)

Because HCS08 devices do not have external address and data buses, the most important functions of an in-circuit emulator have been built onto the chip with the MCU. The debug system consists of an 8-stage FIFO that can store address or data bus information, and a flexible trigger system to decide when to capture bus information and what information to capture. The system relies on the single-wire background debug system to access debug control registers and to read results out of the eight stage FIFO.

The debug module includes control and status registers that are accessible in the user's memory map. These registers are located in the high register space to avoid using valuable direct page memory space.

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in [Section 17.3.6, "Hardware Breakpoints."](#)

### 17.3.1 Comparators A and B

Two 16-bit comparators (A and B) can optionally be qualified with the R/W signal and an opcode tracking circuit. Separate control bits allow you to ignore R/W for each comparator. The opcode tracking circuitry optionally allows you to specify that a trigger will occur only if the opcode at the specified address is actually executed as opposed to only being read from memory into the instruction queue. The comparators are also capable of magnitude comparisons to support the inside range and outside range trigger modes. Comparators are disabled temporarily during all BDC accesses.

The A comparator is always associated with the 16-bit CPU address. The B comparator compares to the CPU address or the 8-bit CPU data bus, depending on the trigger mode selected. Because the CPU data bus is separated into a read data bus and a write data bus, the RWAEN and RWA control bits have an additional purpose, in full address plus data comparisons they are used to decide which of these buses to use in the comparator B data bus comparisons. If RWAEN = 1 (enabled) and RWA = 0 (write), the CPU's write data bus is used. Otherwise, the CPU's read data bus is used.

The currently selected trigger mode determines what the debugger logic does when a comparator detects a qualified match condition. A match can cause:

- Generation of a breakpoint to the CPU
- Storage of data bus values into the FIFO
- Starting to store change-of-flow addresses into the FIFO (begin type trace)
- Stopping the storage of change-of-flow addresses into the FIFO (end type trace)

### 17.3.2 Bus Capture Information and FIFO Operation

The usual way to use the FIFO is to setup the trigger mode and other control options, then arm the debugger. When the FIFO has filled or the debugger has stopped storing data into the FIFO, you would read the information out of it in the order it was stored into the FIFO. Status bits indicate the number of words of valid information that are in the FIFO as data is stored into it. If a trace run is manually halted by writing 0 to ARM before the FIFO is full (CNT = 1:0:0:0), the information is shifted by one position and

## A.4 Thermal Characteristics

This section provides information about operating temperature range, power dissipation, and package thermal resistance. Power dissipation on I/O pins is usually small compared to the power dissipation in on-chip logic and voltage regulator circuits, and it is user-determined rather than being controlled by the MCU design. To take  $P_{I/O}$  into account in power calculations, determine the difference between actual pin voltage and  $V_{SS}$  or  $V_{DD}$  and multiply by the pin current for each I/O pin. Except in cases of unusually high pin current (heavy loads), the difference between pin voltage and  $V_{SS}$  or  $V_{DD}$  will be very small.

**Table A-3. Thermal Characteristics**

#	C	Rating	Symbol	Value	Unit	Temp Rated			
						Standard	AEC Grade 0		
1	—	Operating temperature range (packaged)							
		Temperature Code W	$T_A$	–40 to 150	°C	—	◆		
		Temperature Code J		–40 to 140		—	◆		
		Temperature Code M		–40 to 125		◆	—		
		Temperature Code V		–40 to 105		◆	—		
Temperature Code C	–40 to 85	◆		—					
2	D	Thermal resistance, Single-layer board							
				Airflow @200 ft/min	Natural Convection				
		28-pin TSSOP	$\theta_{JA}$	71	91	°C/W	◆	◆	
		20-pin TSSOP		94	114		◆	—	
16-pin TSSOP	108	133		◆	◆				
3	D	Thermal resistance, Four-layer board							
				Airflow @200 ft/min	Natural Convection				
		28-pin TSSOP	$\theta_{JA}$	51	58	°C/W	◆	◆	
		20-pin TSSOP		68	75		◆	—	
16-pin TSSOP	78	92		◆	◆				
4	D	Maximum junction temperature	$T_J$	135	°C	◆	—		
				155		—	◆		