

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	I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
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 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg32e1ctj">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg32e1ctj</a>

Table 1. MC9S08SG32 Rev. 1 Addendum

Location	Description																																																																																																																																
Chapter “Electrical Characteristics”/Section “Thermal Characteristics”/Table A-3. Thermal Characteristics/Page 293	<p>Update Table A-3. Thermal Characteristics as follows:</p> <ul style="list-style-type: none"><li>—Change the value for row “Thermal resistance,Single-layer board/28-pin TSSOP/Airflow @200ft/min.” from 71 to 72 C/W</li><li>—Change the value for 16-pin TSSOP/Thermalresistance</li><li>1.Single layer board / Airflow @ 200ft/min. from 108 to 113 C/W.</li><li>2.Four layer board / Airflow @ 200ft/min. from 78 to 84 C/W.</li><li>• Update parameter 4 of Table “A-3.Thermal Characteristics” .</li></ul> <table><tr><th rowspan="2">#</th><th rowspan="2">C</th><th rowspan="2">Rating</th><th rowspan="2">Symbol</th><th rowspan="2">Value</th><th rowspan="2">Unit</th><th colspan="2">Temp Rated</th></tr><tr><th>Standard</th><th>AEC Grade 0</th></tr><tr><td rowspan="6">1</td><td rowspan="6">—</td><td colspan="5">Operating temperature range (packaged)</td></tr><tr><td>Temperature Code W</td><td rowspan="5">T<sub>A</sub></td><td>–40 to 150</td><td rowspan="5">°C</td><td>—</td><td>♦</td></tr><tr><td>Temperature Code J</td><td>–40 to 140</td><td>—</td><td>♦</td></tr><tr><td>Temperature Code M</td><td>–40 to 125</td><td>♦</td><td>—</td></tr><tr><td>Temperature Code V</td><td>–40 to 105</td><td>♦</td><td>—</td></tr><tr><td>Temperature Code C</td><td>–40 to 85</td><td>♦</td><td>—</td></tr><tr><td rowspan="4">2</td><td rowspan="4">D</td><td colspan="5">Thermal resistance, Single-layer board</td></tr><tr><td></td><td></td><td>Airflow @200 ft/min</td><td>Natural Convection</td><td></td><td></td></tr><tr><td>28-pin TSSOP</td><td rowspan="3">θ<sub>JA</sub></td><td>72</td><td>91</td><td rowspan="3">°C/W</td><td>♦</td><td>♦</td></tr><tr><td>20-pin TSSOP</td><td>94</td><td>114</td><td>♦</td><td>—</td></tr><tr><td>16-pin TSSOP</td><td>113</td><td>133</td><td>♦</td><td>♦</td></tr><tr><td rowspan="4">3</td><td rowspan="4">D</td><td colspan="5">Thermal resistance, Four-layer board</td></tr><tr><td></td><td></td><td>Airflow @200 ft/min</td><td>Natural Convection</td><td></td><td></td></tr><tr><td>28-pin TSSOP</td><td rowspan="3">θ<sub>JA</sub></td><td>51</td><td>58</td><td rowspan="3">°C/W</td><td>♦</td><td>♦</td></tr><tr><td>20-pin TSSOP</td><td>68</td><td>75</td><td>♦</td><td>—</td></tr><tr><td>16-pin TSSOP</td><td>84</td><td>92</td><td>♦</td><td>♦</td></tr><tr><td rowspan="6">4</td><td rowspan="6">D</td><td colspan="5">Maximum junction temperature</td></tr><tr><td>Temperature Code W</td><td rowspan="5">T<sub>J</sub></td><td>155</td><td rowspan="5">°C</td><td>—</td><td>♦</td></tr><tr><td>Temperature Code J</td><td>150</td><td>—</td><td>♦</td></tr><tr><td>Temperature Code M</td><td>135</td><td>♦</td><td>—</td></tr><tr><td>Temperature Code V</td><td>115</td><td>♦</td><td>—</td></tr><tr><td>Temperature Code C</td><td>95</td><td>♦</td><td>—</td></tr></table>	#	C	Rating	Symbol	Value	Unit	Temp Rated		Standard	AEC Grade 0	1	—	Operating temperature range (packaged)					Temperature Code W	T <sub>A</sub>	–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	θ <sub>JA</sub>	72	91	°C/W	♦	♦	20-pin TSSOP	94	114	♦	—	16-pin TSSOP	113	133	♦	♦	3	D	Thermal resistance, Four-layer board							Airflow @200 ft/min	Natural Convection			28-pin TSSOP	θ <sub>JA</sub>	51	58	°C/W	♦	♦	20-pin TSSOP	68	75	♦	—	16-pin TSSOP	84	92	♦	♦	4	D	Maximum junction temperature					Temperature Code W	T <sub>J</sub>	155	°C	—	♦	Temperature Code J	150	—	♦	Temperature Code M	135	♦	—	Temperature Code V	115	♦	—	Temperature Code C	95	♦	—
#	C							Rating	Symbol	Value	Unit			Temp Rated																																																																																																																			
		Standard	AEC Grade 0																																																																																																																														
1	—	Operating temperature range (packaged)																																																																																																																															
		Temperature Code W	T <sub>A</sub>	–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	θ <sub>JA</sub>	72	91	°C/W	♦	♦																																																																																																																									
		20-pin TSSOP		94	114		♦	—																																																																																																																									
16-pin TSSOP	113	133		♦	♦																																																																																																																												
3	D	Thermal resistance, Four-layer board																																																																																																																															
				Airflow @200 ft/min	Natural Convection																																																																																																																												
		28-pin TSSOP	θ <sub>JA</sub>	51	58	°C/W	♦	♦																																																																																																																									
		20-pin TSSOP		68	75		♦	—																																																																																																																									
16-pin TSSOP	84	92		♦	♦																																																																																																																												
4	D	Maximum junction temperature																																																																																																																															
		Temperature Code W	T <sub>J</sub>	155	°C	—	♦																																																																																																																										
		Temperature Code J		150		—	♦																																																																																																																										
		Temperature Code M		135		♦	—																																																																																																																										
		Temperature Code V		115		♦	—																																																																																																																										
		Temperature Code C		95		♦	—																																																																																																																										
Chapter “Electrical Characteristics”/Section “DC Characteristics”/Table A-6. DC Characteristics/Page 298	<p>In the Table “DC Characteristics” add note 11 and 12 for parameter #18.</p> <p>Note 11: Device functionality is guaranteed between the LVD threshold VLVD0 and VDD Min. When VDD is below the minimum operating voltage (VDD Min), the analog parameters for the IO pins, ACMP and ADC, are not guaranteed to meet data sheet performance parameters.</p> <p>Note 12: In addition to LVD, it is recommended to also use the LVW feature. LVW can trigger an interrupt and be used as an indicator to warn that the VDD is dropping,so that the software can take actions accordingly before the VDD drops below VDD Min.</p>																																																																																																																																



Section Number	Title	Page
9.5.1	ADC Module Initialization Example .....	144
9.6	Application Information .....	146
9.6.1	External Pins and Routing .....	146
9.6.2	Sources of Error .....	148

## Chapter 10

### Inter-Integrated Circuit (S08IICV2)

10.1	Introduction .....	151
10.1.1	Module Configuration .....	151
10.1.2	Features .....	153
10.1.3	Modes of Operation .....	153
10.1.4	Block Diagram .....	154
10.2	External Signal Description .....	154
10.2.1	SCL — Serial Clock Line .....	154
10.2.2	SDA — Serial Data Line .....	154
10.3	Register Definition .....	154
10.3.1	IIC Address Register (IICA) .....	155
10.3.2	IIC Frequency Divider Register (IICF) .....	155
10.3.3	IIC Control Register (IICC1) .....	158
10.3.4	IIC Status Register (IICS) .....	159
10.3.5	IIC Data I/O Register (IICD) .....	160
10.3.6	IIC Control Register 2 (IICC2) .....	160
10.4	Functional Description .....	161
10.4.1	IIC Protocol .....	161
10.4.2	10-bit Address .....	165
10.4.3	General Call Address .....	166
10.5	Resets .....	166
10.6	Interrupts .....	166
10.6.1	Byte Transfer Interrupt .....	166
10.6.2	Address Detect Interrupt .....	166
10.6.3	Arbitration Lost Interrupt .....	166
10.7	Initialization/Application Information .....	168

## Chapter 11

### Internal Clock Source (S08ICSV2)

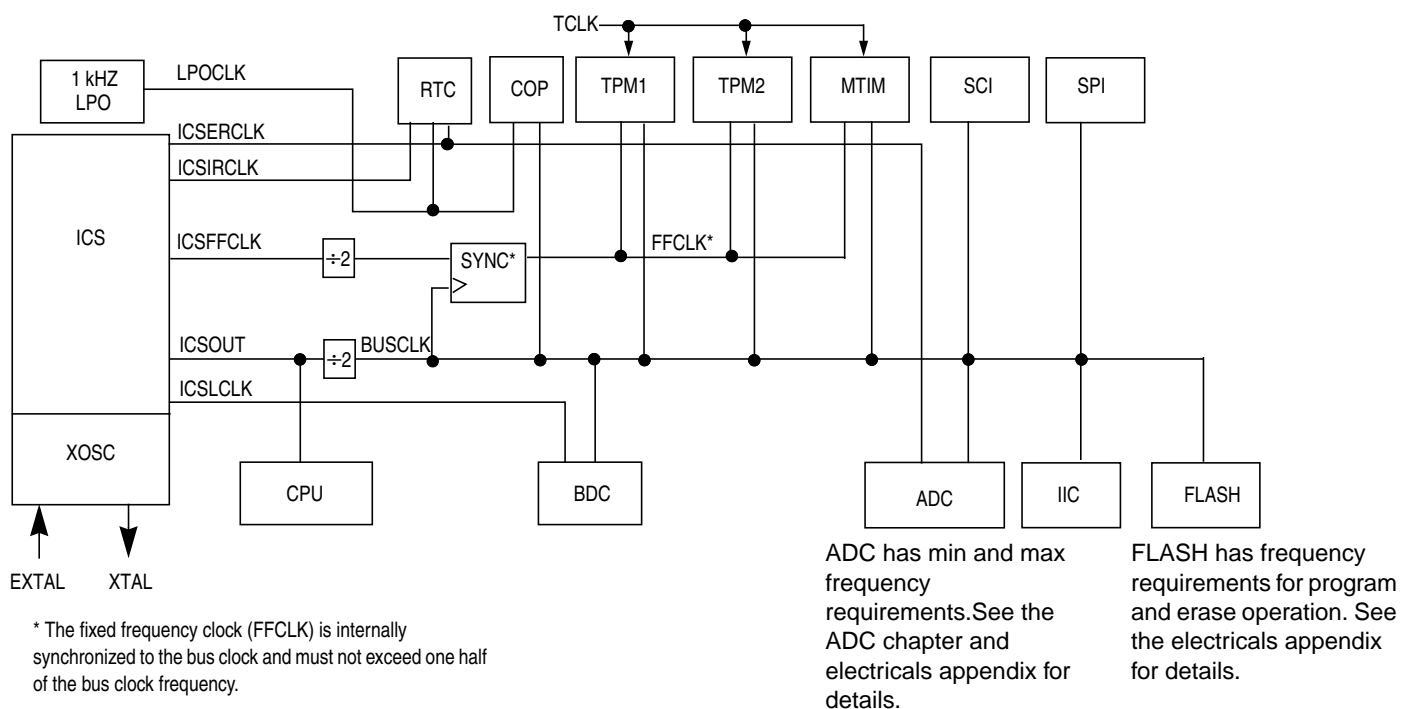
11.1	Introduction .....	171
11.1.1	Module Configuration .....	171
11.1.2	Features .....	173
11.1.3	Block Diagram .....	173
11.1.4	Modes of Operation .....	174
11.2	External Signal Description .....	175
11.3	Register Definition .....	175

## 1.3 System Clock Distribution

Figure 1-2 shows a simplified clock connection diagram. Some modules in the MCU have selectable clock inputs as shown. The clock inputs to the modules indicate the clock(s) that are used to drive the module function.

The following defines the clocks used in this MCU:

- **BUSCLK** — The frequency of the bus is always half of ICSOUT.
- **ICSOUT** — Primary output of the ICS and is twice the bus frequency.
- **ICSLCLK** — Development tools can select this clock source to speed up BDC communications in systems where the bus clock is configured to run at a very slow frequency.
- **ICSERCLK** — External reference clock can be selected as the RTC clock source and as the alternate clock for the ADC module.
- **ICSIRCLK** — Internal reference clock can be selected as the RTC clock source.
- **ICSFFCLK** — Fixed frequency clock can be selected as clock source for the TPM1, TPM2 and MTIM modules.
- **LPOCLK** — Independent 1-kHz clock source that can be selected as the clock source for the COP and RTC modules.
- **TCLK** — External input clock source for TPM1, TPM2 and MTIM and is referenced as TPMCLK in TPM chapters.



**Figure 1-2. System Clock Distribution Diagram**



The COP counter is initialized by the first writes to the SOPT1 and SOPT2 registers after any system reset. Subsequent writes to SOPT1 and SOPT2 have no effect on COP operation. Even if the application will use the reset default settings of COPT, COPCLKS, and COPW bits, the user should write to the write-once SOPT1 and SOPT2 registers during reset initialization to lock in the settings. This will prevent accidental changes if the application program gets lost.

The write to SRS that services (clears) the COP counter should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

If the bus clock source is selected, the COP counter does not increment while the MCU is in background debug mode or while the system is in stop mode. The COP counter resumes when the MCU exits background debug mode or stop mode.

If the 1-kHz clock source is selected, the COP counter is re-initialized to zero upon entry to either background debug mode or stop mode and begins from zero upon exit from background debug mode or stop mode.

## 5.5 Interrupts

Interrupts provide a way to save the current CPU status and registers, execute an interrupt service routine (ISR), and then restore the CPU status so processing resumes where it left off before the interrupt. Other than the software interrupt (SWI), which is a program instruction, interrupts are caused by hardware events such as an edge on a pin interrupt or a timer-overflow event. The debug module can also generate an SWI under certain circumstances.

If an event occurs in an enabled interrupt source, an associated read-only status flag will become set. The CPU will not respond unless the local interrupt enable is a 1 to enable the interrupt and the I bit in the CCR is 0 to allow interrupts. The global interrupt mask (I bit) in the CCR is initially set after reset which prevents all maskable interrupt sources. The user program initializes the stack pointer and performs other system setup before clearing the I bit to allow the CPU to respond to interrupts.

When the CPU receives a qualified interrupt request, it completes the current instruction before responding to the interrupt. The interrupt sequence obeys the same cycle-by-cycle sequence as the SWI instruction and consists of:

- Saving the CPU registers on the stack
- Setting the I bit in the CCR to mask further interrupts
- Fetching the interrupt vector for the highest-priority interrupt that is currently pending
- Filling the instruction queue with the first three bytes of program information starting from the address fetched from the interrupt vector locations

While the CPU is responding to the interrupt, the I bit is automatically set to avoid the possibility of another interrupt interrupting the ISR itself (this is called nesting of interrupts). Normally, the I bit is restored to 0 when the CCR is restored from the value stacked on entry to the ISR. In rare cases, the I bit can be cleared inside an ISR (after clearing the status flag that generated the interrupt) so that other interrupts can be serviced without waiting for the first service routine to finish. This practice is not recommended for anyone

## Chapter 7

# Central Processor Unit (S08CPUV3)

### 7.1 Introduction

This section provides summary information about the registers, addressing modes, and instruction set of the CPU of the HCS08 Family. For a more detailed discussion, refer to the *HCS08 Family Reference Manual, volume 1*, Freescale Semiconductor document order number HCS08RMV1/D.

The HCS08 CPU is fully source- and object-code-compatible with the M68HC08 CPU. Several instructions and enhanced addressing modes were added to improve C compiler efficiency and to support a new background debug system which replaces the monitor mode of earlier M68HC08 microcontrollers (MCU).

#### 7.1.1 Features

Features of the HCS08 CPU include:

- Object code fully upward-compatible with M68HC05 and M68HC08 Families
- All registers and memory are mapped to a single 64-Kbyte address space
- 16-bit stack pointer (any size stack anywhere in 64-Kbyte address space)
- 16-bit index register (H:X) with powerful indexed addressing modes
- 8-bit accumulator (A)
- Many instructions treat X as a second general-purpose 8-bit register
- Seven addressing modes:
  - Inherent — Operands in internal registers
  - Relative — 8-bit signed offset to branch destination
  - Immediate — Operand in next object code byte(s)
  - Direct — Operand in memory at 0x0000–0x00FF
  - Extended — Operand anywhere in 64-Kbyte address space
  - Indexed relative to H:X — Five submodes including auto increment
  - Indexed relative to SP — Improves C efficiency dramatically
- Memory-to-memory data move instructions with four address mode combinations
- Overflow, half-carry, negative, zero, and carry condition codes support conditional branching on the results of signed, unsigned, and binary-coded decimal (BCD) operations
- Efficient bit manipulation instructions
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- STOP and WAIT instructions to invoke low-power operating modes



## 9.1.6 Features

Features of the ADC module include:

- Linear successive approximation algorithm with 10-bit resolution
- Up to 28 analog inputs<sup>1</sup>
- Output formatted in 10- or 8-bit right-justified unsigned format
- Single or continuous conversion (automatic return to idle after single conversion)
- Configurable sample time and conversion speed/power
- Conversion complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in wait or stop3 modes for lower noise operation
- Asynchronous clock source for lower noise operation
- Selectable asynchronous hardware conversion trigger
- Automatic compare with interrupt for less-than, or greater-than or equal-to, programmable value

## 9.1.7 ADC Module Block Diagram

Figure 9-2 provides a block diagram of the ADC module

1. Number of analog inputs varies according to the device and may be from external or internal sources. Refer to the introduction section to this chapter for AD0–AD27 channel input assignments.

2. Update status and control register 2 (ADCSC2) to select the conversion trigger (hardware or software) and compare function options, if enabled.
3. Update status and control register 1 (ADCSC1) to select whether conversions will be continuous or completed only once, and to enable or disable conversion complete interrupts. The input channel on which conversions will be performed is also selected here.

### 9.5.1.2 Pseudo-Code Example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low power with a long sample time on input channel 1, where the internal ADCK clock is derived from the bus clock divided by 1.

**ADCCFG = 0x98 (%10011000)**

Bit 7	ADLPC	1	Configures for low power (lowers maximum clock speed)
Bit 6:5	ADIV	00	Sets the ADCK to the input clock ÷ 1
Bit 4	ADLSMP	1	Configures for long sample time
Bit 3:2	MODE	10	Sets mode at 10-bit conversions
Bit 1:0	ADICLK	00	Selects bus clock as input clock source

**ADCSC2 = 0x00 (%00000000)**

Bit 7	ADACT	0	Flag indicates if a conversion is in progress
Bit 6	ADTRG	0	Software trigger selected
Bit 5	ACFE	0	Compare function disabled
Bit 4	ACFGT	0	Not used in this example
Bit 3:2		00	Reserved, always reads zero
Bit 1:0		00	Reserved for Freescale's internal use; always write zero

**ADCSC1 = 0x41 (%01000001)**

Bit 7	COCO	0	Read-only flag which is set when a conversion completes
Bit 6	AIEN	1	Conversion complete interrupt enabled
Bit 5	ADCO	0	One conversion only (continuous conversions disabled)
Bit 4:0	ADCH	00001	Input channel 1 selected as ADC input channel

**ADCRH/L = 0xxx**

Holds results of conversion. Read high byte (ADCRH) before low byte (ADCRL) so that conversion data cannot be overwritten with data from the next conversion.

**ADCCVH/L = 0xxx**

Holds compare value when compare function enabled

**APCTL1=0x02**

AD1 pin I/O control disabled. All other AD pins remain general purpose I/O pins

**APCTL2=0x00**

All other AD pins remain general purpose I/O pins



## 12.1.2 Features

Timer system features include:

- 8-bit up-counter
  - Free-running or 8-bit modulo limit
  - Software controllable interrupt on overflow
  - Counter reset bit (TRST)
  - Counter stop bit (TSTP)
- Four software selectable clock sources for input to prescaler:
  - System bus clock — rising edge
  - Fixed frequency clock (XCLK) — rising edge
  - External clock source on the TCLK pin — rising edge
  - External clock source on the TCLK pin — falling edge
- Nine selectable clock prescale values:
  - Clock source divide by 1, 2, 4, 8, 16, 32, 64, 128, or 256

## 12.1.3 Modes of Operation

This section defines the MTIM's operation in stop, wait and background debug modes.

### 12.1.3.1 MTIM in Wait Mode

The MTIM continues to run in wait mode if enabled before executing the WAIT instruction. Therefore, the MTIM can be used to bring the MCU out of wait mode if the timer overflow interrupt is enabled. For lowest possible current consumption, the MTIM should be stopped by software if not needed as an interrupt source during wait mode.

### 12.1.3.2 MTIM in Stop Modes

The MTIM is disabled in all stop modes, regardless of the settings before executing the STOP instruction. Therefore, the MTIM cannot be used as a wake up source from stop modes.

Waking from stop2 mode, the MTIM will be put into its reset state. If stop3 is exited with a reset, the MTIM will be put into its reset state. If stop3 is exited with an interrupt, the MTIM continues from the state it was in when stop3 was entered. If the counter was active upon entering stop3, the count will resume from the current value.

### 12.1.3.3 MTIM in Active Background Mode

The MTIM suspends all counting until the microcontroller returns to normal user operating mode. Counting resumes from the suspended value as long as an MTIM reset did not occur (TRST written to a 1 or MTIMMOD written).

Figure 14-3 shows the receiver portion of the SCI.

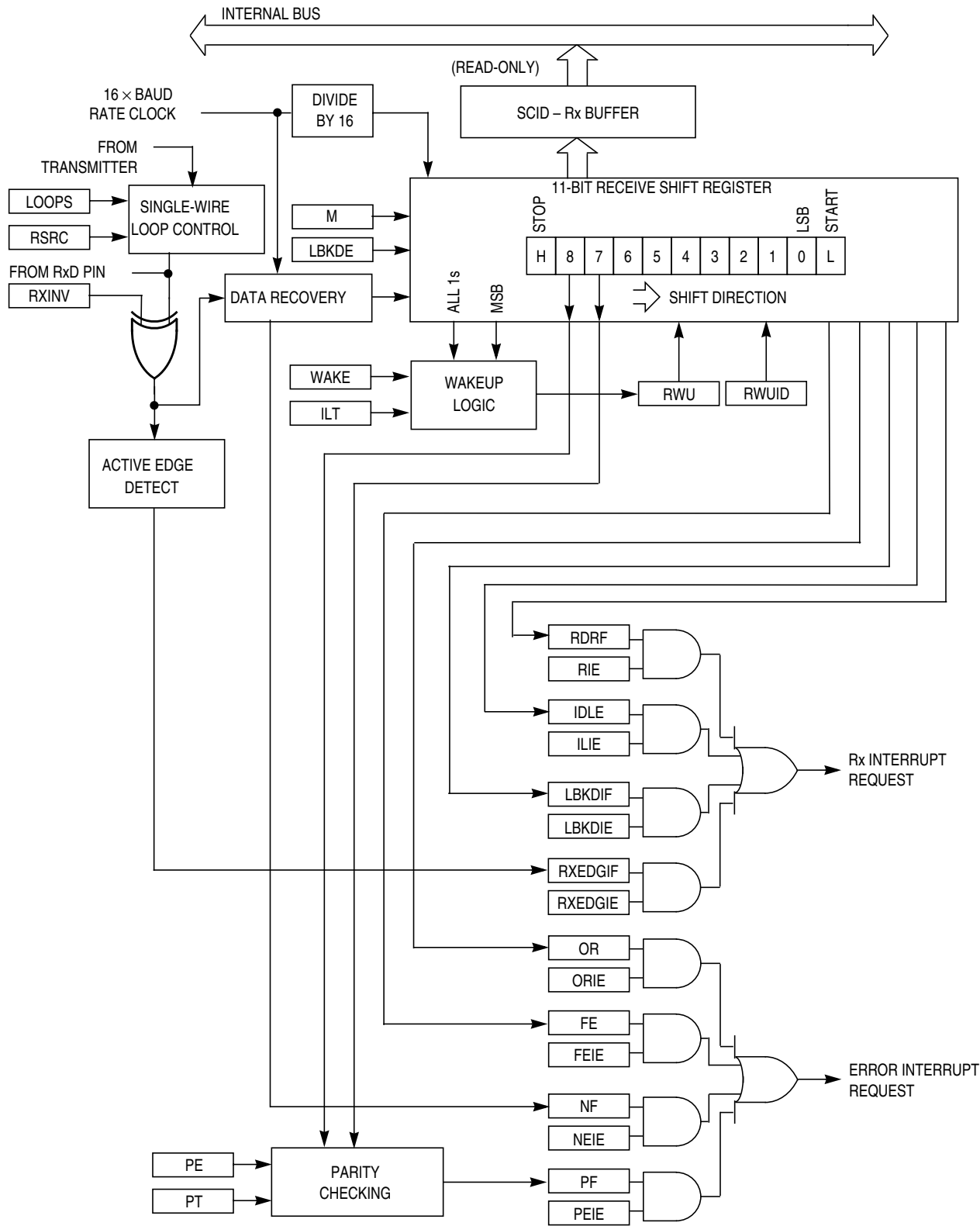


Figure 14-3. SCI Receiver Block Diagram

### 14.3.5.2 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted.

In stop2 mode, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note, because the clocks are halted, the SCI module will resume operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

### 14.3.5.3 Loop Mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general-purpose port I/O pin.

### 14.3.5.4 Single-Wire Operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general-purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

Table 15-3. SPIC2 Register Field Descriptions

Field	Description
4 MODFEN	<b>Master Mode-Fault Function Enable</b> — When the SPI is configured for slave mode, this bit has no meaning or effect. (The $\overline{SS}$ pin is the slave select input.) In master mode, this bit determines how the $\overline{SS}$ pin is used (refer to Table 15-2 for more details). 0 Mode fault function disabled, master $\overline{SS}$ pin reverts to general-purpose I/O not controlled by SPI 1 Mode fault function enabled, master $\overline{SS}$ pin acts as the mode fault input or the slave select output
3 BIDIROE	<b>Bidirectional Mode Output Enable</b> — When bidirectional mode is enabled by SPI pin control 0 (SPC0) = 1, BIDIROE determines whether the SPI data output driver is enabled to the single bidirectional SPI I/O pin. Depending on whether the SPI is configured as a master or a slave, it uses either the MOSI (MOMI) or MISO (SISO) pin, respectively, as the single SPI data I/O pin. When SPC0 = 0, BIDIROE has no meaning or effect. 0 Output driver disabled so SPI data I/O pin acts as an input 1 SPI I/O pin enabled as an output
1 SPISWAI	<b>SPI Stop in Wait Mode</b> 0 SPI clocks continue to operate in wait mode 1 SPI clocks stop when the MCU enters wait mode
0 SPC0	<b>SPI Pin Control 0</b> — The SPC0 bit chooses single-wire bidirectional mode. If MSTR = 0 (slave mode), the SPI uses the MISO (SISO) pin for bidirectional SPI data transfers. If MSTR = 1 (master mode), the SPI uses the MOSI (MOMI) pin for bidirectional SPI data transfers. When SPC0 = 1, BIDIROE is used to enable or disable the output driver for the single bidirectional SPI I/O pin. 0 SPI uses separate pins for data input and data output 1 SPI configured for single-wire bidirectional operation

### 15.4.3 SPI Baud Rate Register (SPIBR)

This register is used to set the prescaler and bit rate divisor for an SPI master. This register may be read or written at any time.

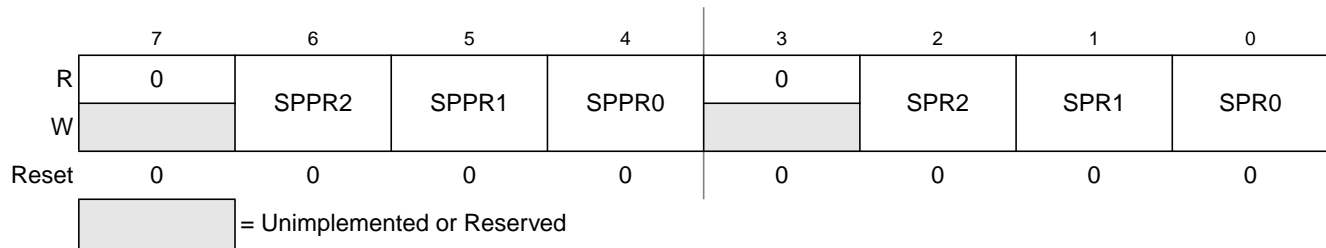


Figure 15-7. SPI Baud Rate Register (SPIBR)

Table 15-4. SPIBR Register Field Descriptions

Field	Description
6:4 SPPR[2:0]	<b>SPI Baud Rate Prescale Divisor</b> — This 3-bit field selects one of eight divisors for the SPI baud rate prescaler as shown in Table 15-5. The input to this prescaler is the bus rate clock (BUSCLK). The output of this prescaler drives the input of the SPI baud rate divider (see Figure 15-4).
2:0 SPR[2:0]	<b>SPI Baud Rate Divisor</b> — This 3-bit field selects one of eight divisors for the SPI baud rate divider as shown in Table 15-6. The input to this divider comes from the SPI baud rate prescaler (see Figure 15-4). The output of this divider is the SPI bit rate clock for master mode.

**Table 15-5. SPI Baud Rate Prescaler Divisor**

SPPR2:SPPR1:SPPR0	Prescaler Divisor
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

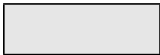
**Table 15-6. SPI Baud Rate Divisor**

SPR2:SPR1:SPR0	Rate Divisor
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

### 15.4.4 SPI Status Register (SPIS)

This register has three read-only status bits. Bits 6, 3, 2, 1, and 0 are not implemented and always read 0. Writes have no meaning or effect.

	7	6	5	4	3	2	1	0
R	SPRF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

 = Unimplemented or Reserved

**Figure 15-8. SPI Status Register (SPIS)**



## 15.5 Functional Description

An SPI transfer is initiated by checking for the SPI transmit buffer empty flag (SPTEF = 1) and then writing a byte of data to the SPI data register (SPID) in the master SPI device. When the SPI shift register is available, this byte of data is moved from the transmit data buffer to the shifter, SPTEF is set to indicate there is room in the buffer to queue another transmit character if desired, and the SPI serial transfer starts.

During the SPI transfer, data is sampled (read) on the MISO pin at one SPSCCK edge and shifted, changing the bit value on the MOSI pin, one-half SPSCCK cycle later. After eight SPSCCK cycles, the data that was in the shift register of the master has been shifted out the MOSI pin to the slave while eight bits of data were shifted in the MISO pin into the master's shift register. At the end of this transfer, the received data byte is moved from the shifter into the receive data buffer and SPRF is set to indicate the data can be read by reading SPID. If another byte of data is waiting in the transmit buffer at the end of a transfer, it is moved into the shifter, SPTEF is set, and a new transfer is started.

Normally, SPI data is transferred most significant bit (MSB) first. If the least significant bit first enable (LSBFE) bit is set, SPI data is shifted LSB first.

When the SPI is configured as a slave, its  $\overline{SS}$  pin must be driven low before a transfer starts and  $\overline{SS}$  must stay low throughout the transfer. If a clock format where CPHA = 0 is selected,  $\overline{SS}$  must be driven to a logic 1 between successive transfers. If CPHA = 1,  $\overline{SS}$  may remain low between successive transfers. See [Section 15.5.1, "SPI Clock Formats"](#) for more details.

Because the transmitter and receiver are double buffered, a second byte, in addition to the byte currently being shifted out, can be queued into the transmit data buffer, and a previously received character can be in the receive data buffer while a new character is being shifted in. The SPTEF flag indicates when the transmit buffer has room for a new character. The SPRF flag indicates when a received character is available in the receive data buffer. The received character must be read out of the receive buffer (read SPID) before the next transfer is finished or a receive overrun error results.

In the case of a receive overrun, the new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for such an overrun condition so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

### 15.5.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

[Figure 15-10](#) shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCCK edge and bit 8 ending one-half SPSCCK cycle after the sixteenth SPSCCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output



BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the frozen TPM counter value.

- This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxSC, TPMxCNTH or TPMxCNTL. Instead, in these conditions the TPM v2 does not clear this read coherency mechanism.
3. Read of TPMxCnVH:L registers ([Section 16.3.5, “TPM Channel Value Registers \(TPMxCnVH:TPMxCnVL\)”](#))
- In TPM v3, any read of TPMxCnVH:L registers during BDM mode returns the value of the TPMxCnVH:L register. In TPM v2, if only one byte of the TPMxCnVH:L registers was read before the BDM mode became active, then any read of TPMxCnVH:L registers during BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the value in the TPMxCnVH:L registers.
  - This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxCnSC. Instead, in this condition the TPM v2 does not clear this read coherency mechanism.
4. Write to TPMxCnVH:L registers
- Input Capture Mode ([Section 16.4.2.1, “Input Capture Mode”](#))  
In this mode the TPM v3 does not allow the writes to TPMxCnVH:L registers. Instead, the TPM v2 allows these writes.
  - Output Compare Mode ([Section 16.4.2.2, “Output Compare Mode”](#))  
In this mode and if (CLKSB:CLKSA not = 0:0), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer at the next change of the TPM counter (end of the prescaler counting) after the second byte is written. Instead, the TPM v2 always updates these registers when their second byte is written.  
The following procedure can be used in the TPM v3 to verify if the TPMxCnVH:L registers were updated with the new value that was written to these registers (value in their write buffer).  
...  
write the new value to TPMxCnVH:L;  
read TPMxCnVH and TPMxCnVL registers;  
while (the read value of TPMxCnVH:L is different from the new value written to TPMxCnVH:L)  
begin  
  read again TPMxCnVH and TPMxCnVL;  
end  
...  
In this point, the TPMxCnVH:L registers were updated, so the program can continue and, for example, write to TPMxC0SC without cancelling the previous write to TPMxCnVH:L registers.
  - Edge-Aligned PWM ([Section 16.4.2.3, “Edge-Aligned PWM Mode”](#))  
In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the

### 17.3.6 Hardware Breakpoints

The BRKEN control bit in the DBGCR register may be set to 1 to allow any of the trigger conditions described in [Section 17.3.5, “Trigger Modes,”](#) to be used to generate a hardware breakpoint request to the CPU. TAG in DBGCR controls whether the breakpoint request will be treated as a tag-type breakpoint or a force-type breakpoint. A tag breakpoint causes the current opcode to be marked as it enters the instruction queue. If a tagged opcode reaches the end of the pipe, the CPU executes a BGND instruction to go to active background mode rather than executing the tagged opcode. A force-type breakpoint causes the CPU to finish the current instruction and then go to active background mode.

If the background mode has not been enabled (ENBDM = 1) by a serial WRITE\_CONTROL command through the BKGD pin, the CPU will execute an SWI instruction instead of going to active background mode.

## 17.4 Register Definition

This section contains the descriptions of the BDC and DBG registers and control bits.

Refer to the high-page register summary in the device overview chapter of this data sheet for the absolute address assignments for all DBG registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

### 17.4.1 BDC Registers and Control Bits

The BDC has two registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the background debug controller.
- The BDC breakpoint match register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in active background mode. (This prevents the ambiguous condition of the control bit forbidding active background mode while the MCU is already in active background mode.) Also, the four status bits (BDMACT, WS, WSF, and DVF) are read-only status indicators and can never be written by the WRITE\_CONTROL serial BDC command. The clock switch (CLKSW) control bit may be read or written at any time.

### 17.4.3.9 Debug Status Register (DBGS)

This is a read-only status register.

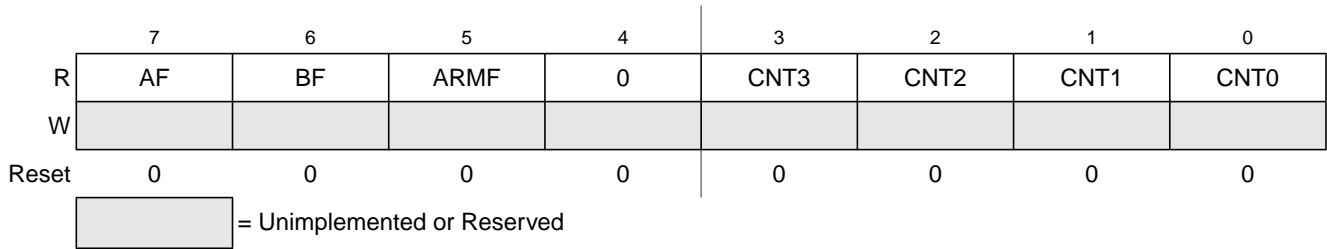


Figure 17-9. Debug Status Register (DBGS)

Table 17-6. DBGS Register Field Descriptions

Field	Description
7 AF	<b>Trigger Match A Flag</b> — AF is cleared at the start of a debug run and indicates whether a trigger match A condition was met since arming. 0 Comparator A has not matched 1 Comparator A match
6 BF	<b>Trigger Match B Flag</b> — BF is cleared at the start of a debug run and indicates whether a trigger match B condition was met since arming. 0 Comparator B has not matched 1 Comparator B match
5 ARMF	<b>Arm Flag</b> — While DBGEN = 1, this status bit is a read-only image of ARM in DBGCR. This bit is set by writing 1 to the ARM control bit in DBGCR (while DBGEN = 1) and is automatically cleared at the end of a debug run. A debug run is completed when the FIFO is full (begin trace) or when a trigger event is detected (end trace). A debug run can also be ended manually by writing 0 to ARM or DBGEN in DBGCR. 0 Debugger not armed 1 Debugger armed
3:0 CNT[3:0]	<b>FIFO Valid Count</b> — These bits are cleared at the start of a debug run and indicate the number of words of valid data in the FIFO at the end of a debug run. The value in CNT does not decrement as data is read out of the FIFO. The external debug host is responsible for keeping track of the count as information is read out of the FIFO. 0000 Number of valid words in FIFO = No valid data 0001 Number of valid words in FIFO = 1 0010 Number of valid words in FIFO = 2 0011 Number of valid words in FIFO = 3 0100 Number of valid words in FIFO = 4 0101 Number of valid words in FIFO = 5 0110 Number of valid words in FIFO = 6 0111 Number of valid words in FIFO = 7 1000 Number of valid words in FIFO = 8