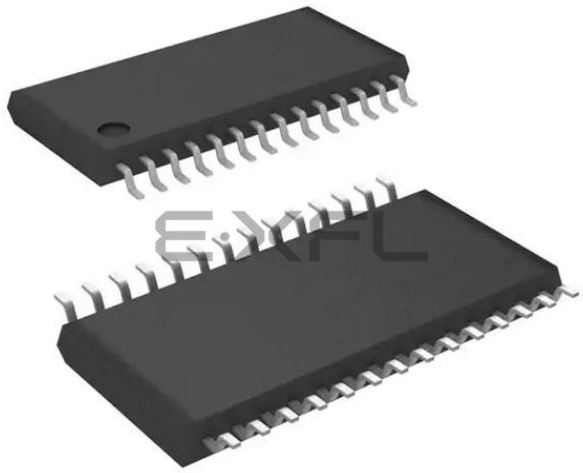


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

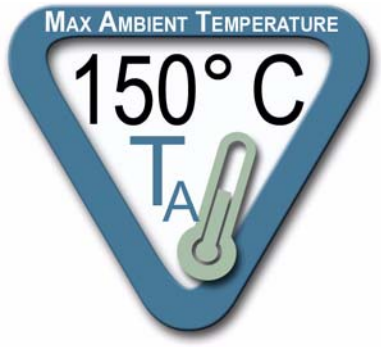
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, LINbus, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 22 |
| Program Memory Size | 16KB (16K 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 16x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-TSSOP (0.173", 4.40mm Width) |
| Supplier Device Package | 28-TSSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg16e1ctl |

# MC9S08SG32
# MC9S08SG16
# Data Sheet

***Now Includes High-Temperature (up to 150 °C) Devices!***




***HCS08***
***Microcontrollers***

MC9S08SG32
Rev. 8
5/2010

freescale.com

freescale™
semiconductor

## 1.2 MCU Block Diagram

The block diagram in Figure 1-1 shows the structure of the MC9S08SG32 Series MCU.

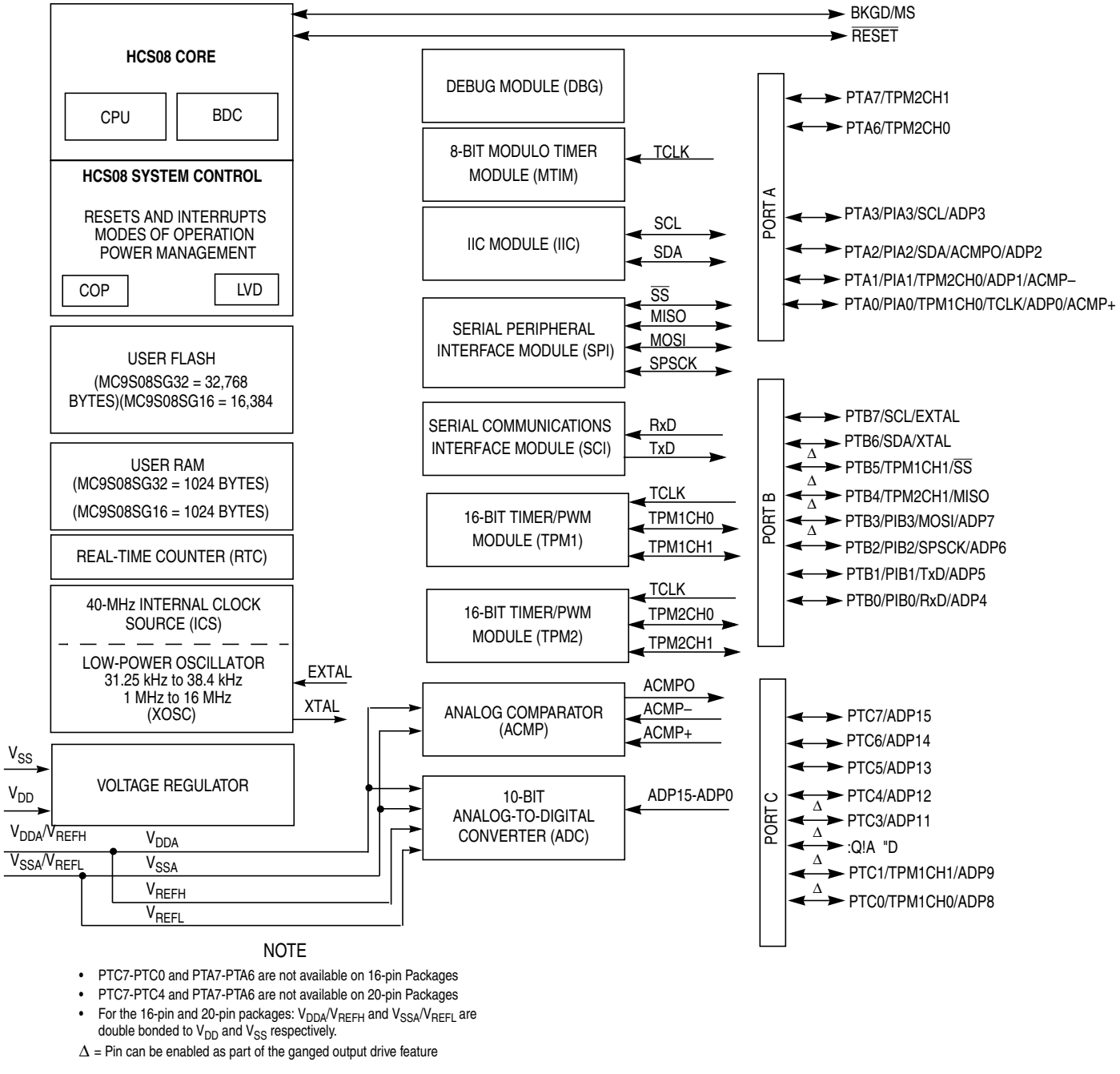



**Figure 1-1. MC9S08SG32 Series Block Diagram**

## 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.

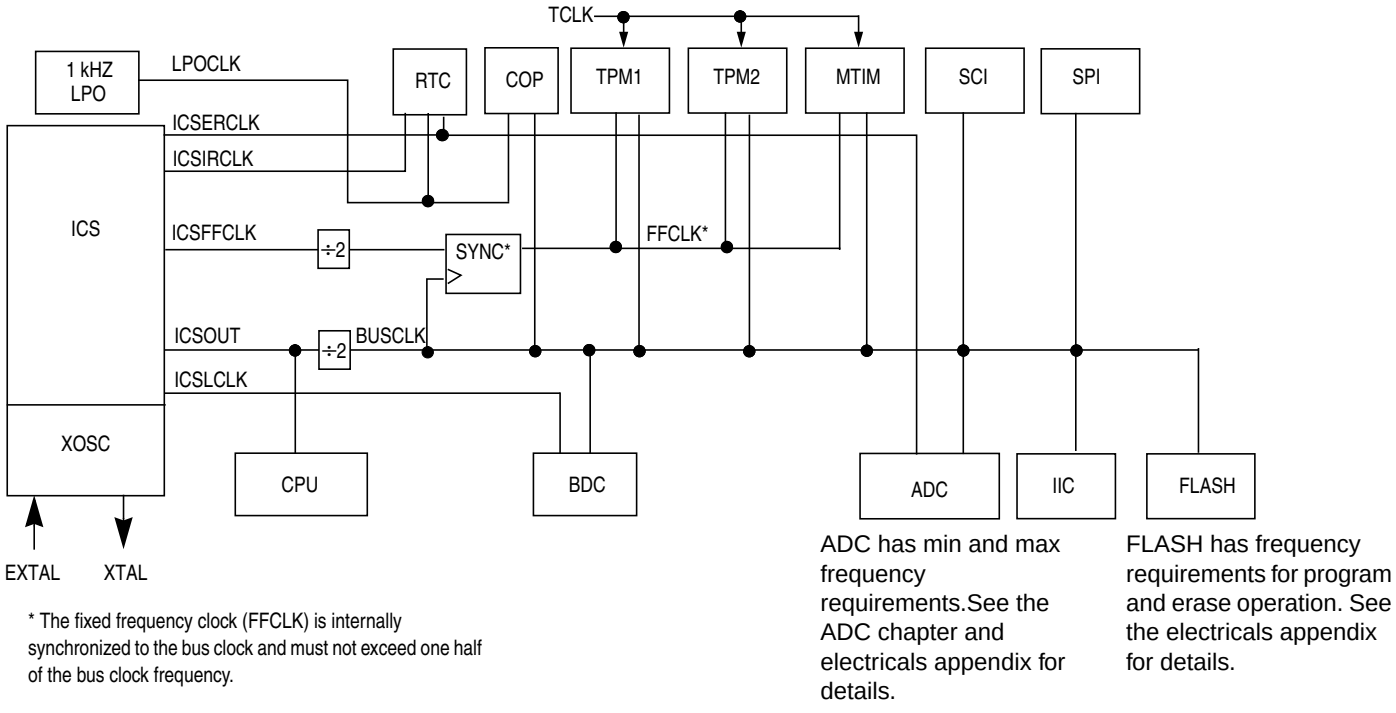



**Figure 1-2. System Clock Distribution Diagram**

## 5.7.6 System Power Management Status and Control 1 Register (SPMSC1)

This high page register contains status and control bits to support the low-voltage detect function, and to enable the bandgap voltage reference for use by the ADC and ACMP modules. This register should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LVWF[1] | 0 | LVWIE | LVDRE[2] | LVDSE[2] | LVDE[2] | 0 | BGBE |
| W | | LVWACK | | | | | | |
| Reset: | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

= Unimplemented or Reserved

1 LVWF will be set in the case when $V_{Supply}$ transitions below the trip point or after reset and $V_{Supply}$ is already below $V_{LVW}$

2 This bit can be written only one time after reset. Additional writes are ignored.

**Figure 5-8. System Power Management Status and Control 1 Register (SPMSC1)**

**Table 5-9. SPMSC1 Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>LVWF | **Low-Voltage Warning Flag** — The LVWF bit indicates the low voltage warning status.<br>0 Low voltage warning is not present.<br>1 Low voltage warning is present or was present. |
| 6<br>LVWACK | **Low-Voltage Warning Acknowledge** — The LVWF bit indicates the low voltage warning status.Writing a 1 to LVWACK clears LVWF to a 0 if a low voltage warning is not present. |
| 5<br>LVWIE | **Low-Voltage Warning Interrupt Enable** — This bit enables hardware interrupt requests for LVWF.<br>0 Hardware interrupt disabled (use polling).<br>1 Request a hardware interrupt when LVWF = 1. |
| 4<br>LVDRE | **Low-Voltage Detect Reset Enable** — This write-once bit enables LVD events to generate a hardware reset (provided LVDE = 1).<br>0 LVD events do not generate hardware resets.<br>1 Force an MCU reset when an enabled low-voltage detect event occurs. |
| 3<br>LVDSE | **Low-Voltage Detect Stop Enable** — Provided LVDE = 1, this write-once bit determines whether the low-voltage detect function operates when the MCU is in stop mode.<br>0 Low-voltage detect disabled during stop mode.<br>1 Low-voltage detect enabled during stop mode. |
| 2<br>LVDE | **Low-Voltage Detect Enable** — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register.<br>0 LVD logic disabled.<br>1 LVD logic enabled. |
| 0<br>BGBE | **Bandgap Buffer Enable** — This bit enables an internal buffer for the bandgap voltage reference for use by the ADC and ACMP modules.<br>0 Bandgap buffer disabled.<br>1 Bandgap buffer enabled. |

### 6.6.1.3 Port A Pull Enable Register (PTAPE)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PTAPE7 | PTAPE6 | R | R | PTAPE3 | PTAPE2 | PTAPE1 | PTAPE0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6-5. Internal Pull Enable for Port A Register (PTAPE)**

**Table 6-4. PTAPE Register Field Descriptions**

| Field | Description |
|---|---|
| 7:5,3:0<br>PTAPE[7:5, 3:0] | **Internal Pull Enable for Port A Bits** — Each of these control bits determines if the internal pull-up or pull-down device is enabled for the associated PTA pin. For port A pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled.<br>0 Internal pull-up/pull-down device disabled for port A bit n.<br>1 Internal pull-up/pull-down device enabled for port A bit n. |
| 5:4<br>Reserved | **Reserved Bits** — These bits are unused on this MCU, writes have no affect and could read as 1s or 0s. |

**NOTE**

Pull-down devices only apply when using pin interrupt functions, when corresponding edge select and pin select functions are configured to detect rising edges.

### 6.6.1.4 Port A Slew Rate Enable Register (PTASE)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PTASE7 | PTASE6 | R | R | PTASE3 | PTASE2 | PTASE1 | PTASE0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6-6. Slew Rate Enable for Port A Register (PTASE)**

**Table 6-5. PTASE Register Field Descriptions**

| Field | Description |
|---|---|
| 7:5,3:0<br>PTASE[7:5, 3:0] | **Output Slew Rate Enable for Port A Bits** — Each of these control bits determines if the output slew rate control is enabled for the associated PTA pin. For port A pins that are configured as inputs, these bits have no effect.<br>0 Output slew rate control disabled for port A bit n.<br>1 Output slew rate control enabled for port A bit n. |
| 5:4<br>Reserved | **Reserved Bits** — These bits are unused on this MCU, writes have no affect and could read as 1s or 0s. |

the intermediate conversion data is lost. In 8-bit mode, there is no interlocking with ADCRH. If the MODE bits are changed, any data in ADCRL becomes invalid.

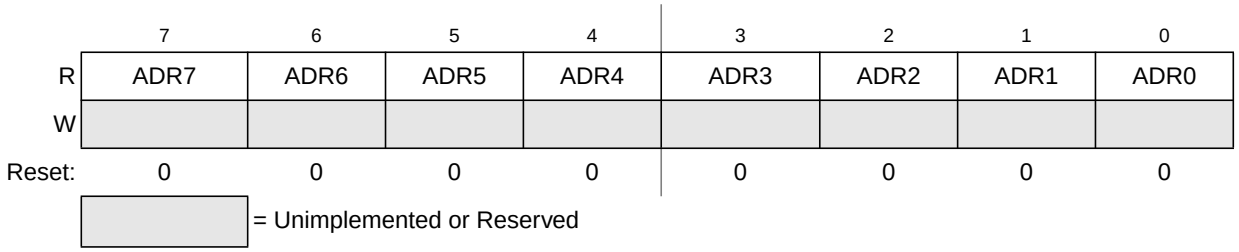



**Figure 9-6. Data Result Low Register (ADCRL)**

## 9.3.5 Compare Value High Register (ADCCVH)

In 10-bit mode, the ADCCVH register holds the upper two bits of the 10-bit compare value (ADCV[9:8]). When the compare function is enabled, these bits are compared to the upper two bits of the result following a conversion in 10-bit mode.

In 8-bit operation, ADCCVH is not used during compare.

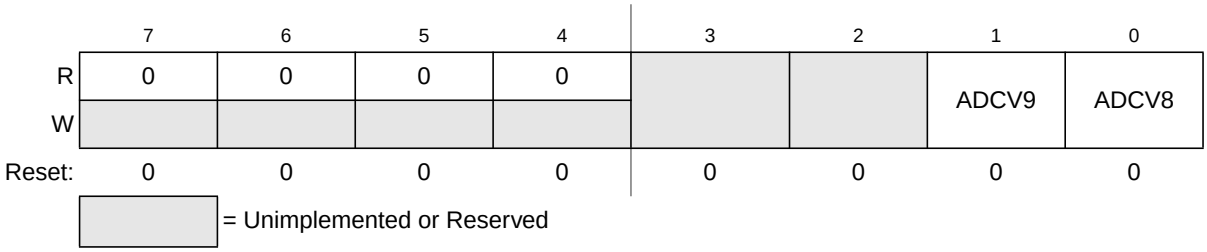



**Figure 9-7. Compare Value High Register (ADCCVH)**

## 9.3.6 Compare Value Low Register (ADCCVL)

The ADCCVL register holds the lower eight bits of the 10-bit compare value or all eight bits of the 8-bit compare value. When the compare function is enabled, bits ADCV[7:0] are compared to the lower eight bits of the result following a conversion in 10-bit or 8-bit mode.

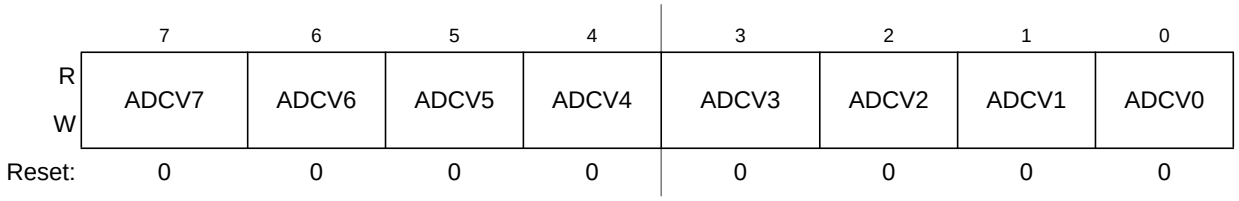



**Figure 9-8. Compare Value Low Register (ADCCVL)**

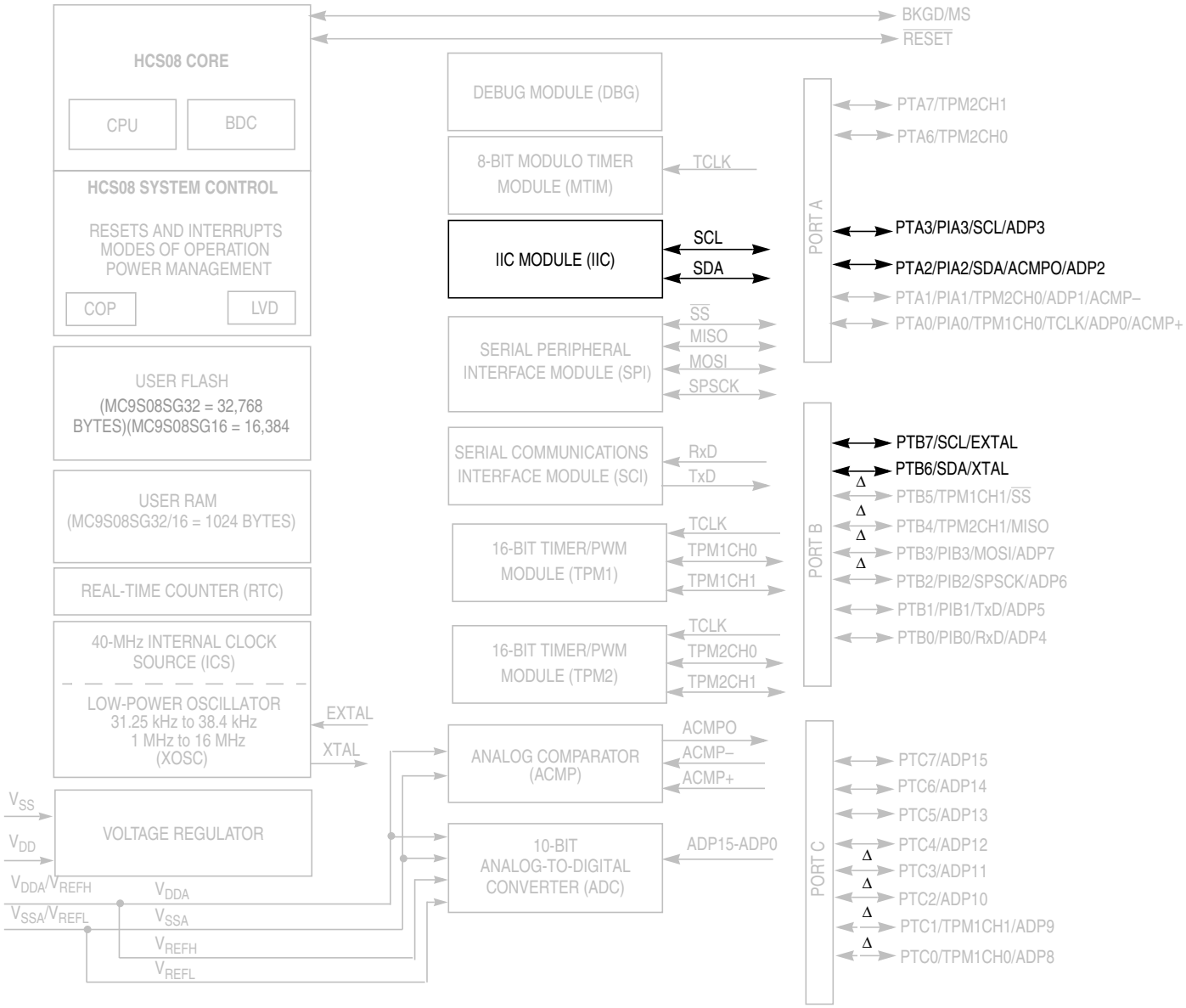
## 9.3.7 Configuration Register (ADCCFG)

ADCCFG selects the mode of operation, clock source, clock divide, and configures for low power and long sample time.

converter yields the lower code (and vice-versa). However, even 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 increases with noise. This error may be reduced by repeatedly sampling the input and averaging the result. Additionally the techniques discussed in Section 9.6.2.3 reduces 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 never converted for any input value.

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




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.

Δ = Pin can be enabled as part of the ganged output drive feature

**Figure 10-1. MC9S08SG32 Series Block Diagram Highlighting IIC Block and Pins**

### 14.1.1 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
  - Active edge on receive pin
  - Break detect supporting LIN
- 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 generation / 11-bit break character detection
- Selectable transmitter output polarity

### 14.1.2 Modes of Operation

See Section 14.3, “Functional Description,” For details concerning SCI operation in these modes:

- 8- and 9-bit data modes
- Stop mode operation
- Loop mode
- Single-wire mode

**Table 14-3. SCIC1 Field Descriptions (continued)**

| Field | Description |
|---|---|
| 3<br>WAKE | **Receiver Wakeup Method Select** — Refer to Section 14.3.3.2, “Receiver Wakeup Operation” for more information.<br>0 Idle-line wakeup.<br>1 Address-mark wakeup. |
| 2<br>ILT | **Idle Line Type Select** — Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to Section 14.3.3.2.1, “Idle-Line Wakeup” for more information.<br>0 Idle character bit count starts after start bit.<br>1 Idle character bit count starts after stop bit. |
| 1<br>PE | **Parity Enable** — Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit.<br>0 No hardware parity generation or checking.<br>1 Parity enabled. |
| 0<br>PT | **Parity Type** — Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even.<br>0 Even parity.<br>1 Odd parity. |

## 14.2.3 SCI Control Register 2 (SCIC2)

This register can be read or written at any time.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 14-7. SCI Control Register 2 (SCIC2)**

**Table 14-4. SCIC2 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TIE | **Transmit Interrupt Enable (for TDRE)**<br>0 Hardware interrupts from TDRE disabled (use polling).<br>1 Hardware interrupt requested when TDRE flag is 1. |
| 6<br>TCIE | **Transmission Complete Interrupt Enable (for TC)**<br>0 Hardware interrupts from TC disabled (use polling).<br>1 Hardware interrupt requested when TC flag is 1. |
| 5<br>RIE | **Receiver Interrupt Enable (for RDRF)**<br>0 Hardware interrupts from RDRF disabled (use polling).<br>1 Hardware interrupt requested when RDRF flag is 1. |
| 4<br>ILIE | **Idle Line Interrupt Enable (for IDLE)**<br>0 Hardware interrupts from IDLE disabled (use polling).<br>1 Hardware interrupt requested when IDLE flag is 1. |

**Table 15-3. SPIC2 Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>MODFEN | **Master Mode-Fault Function Enable** — 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).<br>0 Mode fault function disabled, master $\overline{SS}$ pin reverts to general-purpose I/O not controlled by SPI<br>1 Mode fault function enabled, master $\overline{SS}$ pin acts as the mode fault input or the slave select output |
| 3<br>BIDIROE | **Bidirectional Mode Output Enable** — 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.<br>0 Output driver disabled so SPI data I/O pin acts as an input<br>1 SPI I/O pin enabled as an output |
| 1<br>SPISWAI | **SPI Stop in Wait Mode**<br>0 SPI clocks continue to operate in wait mode<br>1 SPI clocks stop when the MCU enters wait mode |
| 0<br>SPC0 | **SPI Pin Control 0** — 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.<br>0 SPI uses separate pins for data input and data output<br>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.

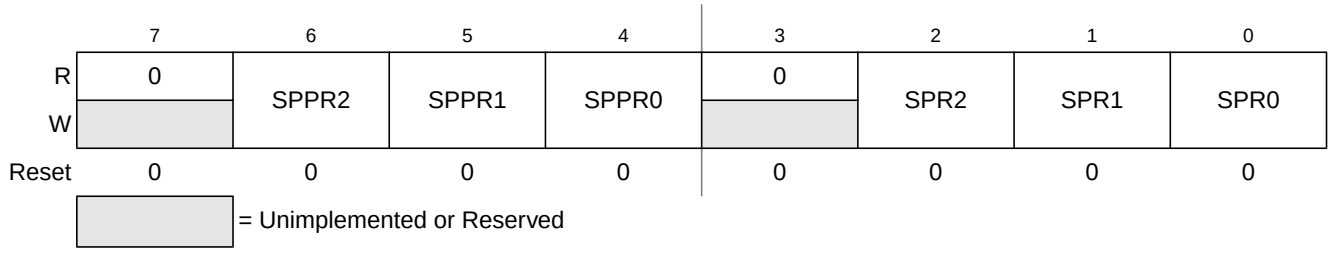



**Figure 15-7. SPI Baud Rate Register (SPIBR)**

**Table 15-4. SPIBR Register Field Descriptions**

| Field | Description |
|---|---|
| 6:4<br>SPPR[2:0] | **SPI Baud Rate Prescale Divisor** — 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<br>SPR[2:0] | **SPI Baud Rate Divisor** — 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)**

pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{SS}$ OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master $\overline{SS}$ output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The $\overline{SS}$ IN waveform applies to the slave select input of a slave.

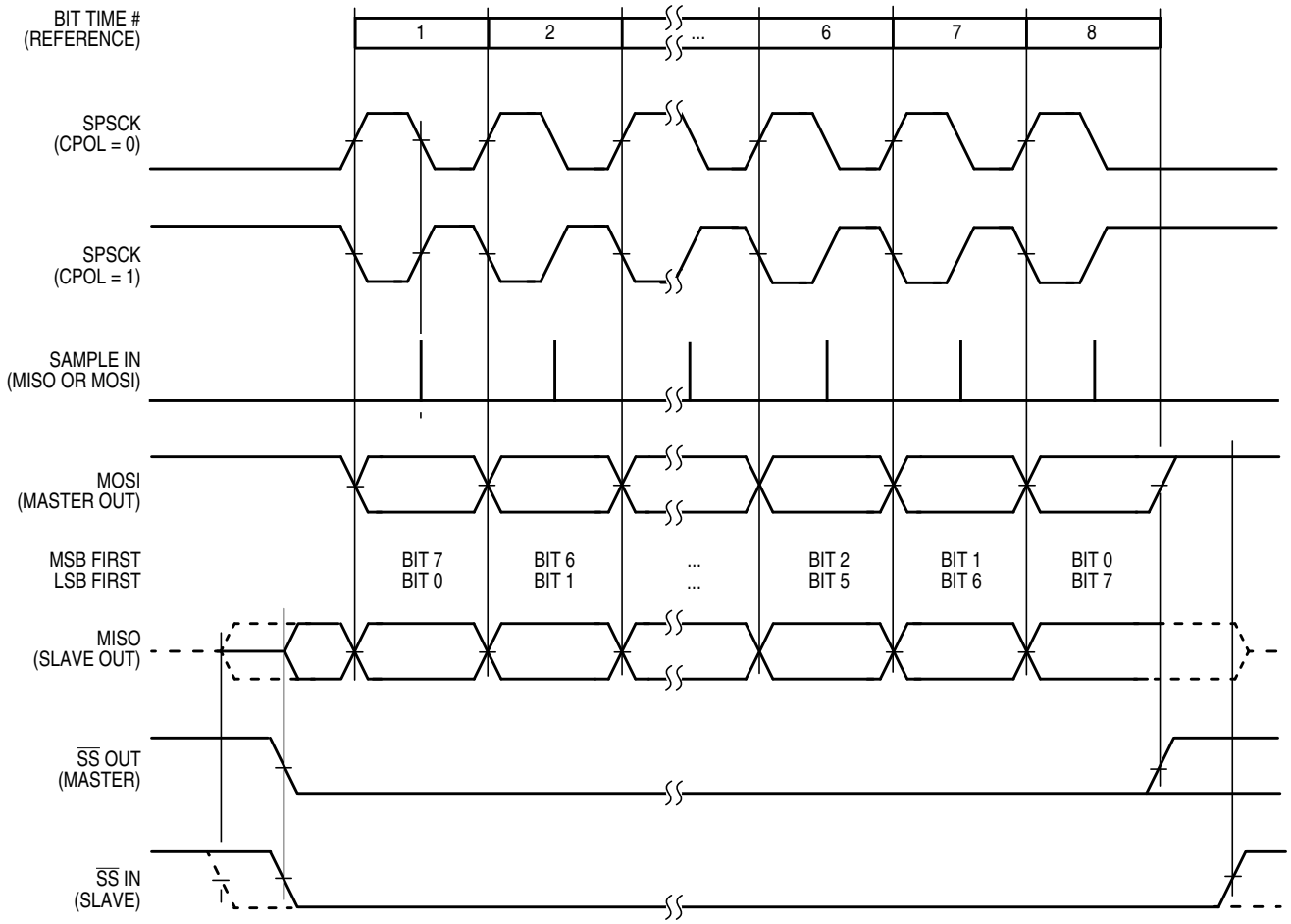



**Figure 15-10. SPI Clock Formats (CPHA = 1)**

When CPHA = 1, the slave begins to drive its MISO output when $\overline{SS}$ goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CHPA = 1, the slave's $\overline{SS}$ input is not required to go to its inactive high level between transfers.

Figure 15-11 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected ($\overline{SS}$ IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| W | Any write to TPMxCNTL clears the 16-bit counter | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 16-9. TPM Counter Register Low (TPMxCNTL)**

When BDM is active, the timer counter is frozen (this is the value that will be read by user); the coherency mechanism is frozen such that the buffer latches remain in the state they were in when the BDM became active, even if one or both counter halves are read while BDM is active. This assures that if the user was in the middle of reading a 16-bit register when BDM became active, it will read the appropriate value from the other half of the 16-bit value after returning to normal execution.

In BDM mode, writing any value to TPMxSC, TPMxCNTH or TPMxCNTL registers resets the read coherency mechanism of the TPMxCNTH:L registers, regardless of the data involved in the write.

## 16.3.3 TPM Counter Modulo Registers (TPMxMODH:TPMxMODL)

The read/write TPM modulo registers contain the modulo value for the TPM counter. After the TPM counter reaches the modulo value, the TPM counter resumes counting from 0x0000 at the next clock, and the overflow flag (TOF) becomes set. Writing to TPMxMODH or TPMxMODL inhibits the TOF bit and overflow interrupts until the other byte is written. Reset sets the TPM counter modulo registers to 0x0000 which results in a free running timer counter (modulo disabled).

Writing to either byte (TPMxMODH or TPMxMODL) latches the value into a buffer and the registers are updated with the value of their write buffer according to the value of CLKSB:CLKSA bits, so:

- If (CLKSB:CLKSA = 0:0), then the registers are updated when the second byte is written
- If (CLKSB:CLKSA not = 0:0), then the registers are updated after both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL - 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter, the update is made when the TPM counter changes from 0xFFFE to 0xFFFF

The latching mechanism may be manually reset by writing to the TPMxSC address (whether BDM is active or not).

When BDM is active, the coherency mechanism is frozen (unless reset by writing to TPMxSC register) such that the buffer latches remain in the state they were in when the BDM became active, even if one or both halves of the modulo register are written while BDM is active. Any write to the modulo registers bypasses the buffer latches and directly writes to the modulo register while BDM is active.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 16-10. TPM Counter Modulo Register High (TPMxMODH)**

**Table 17-2. BDCSCR Register Field Descriptions (continued)**

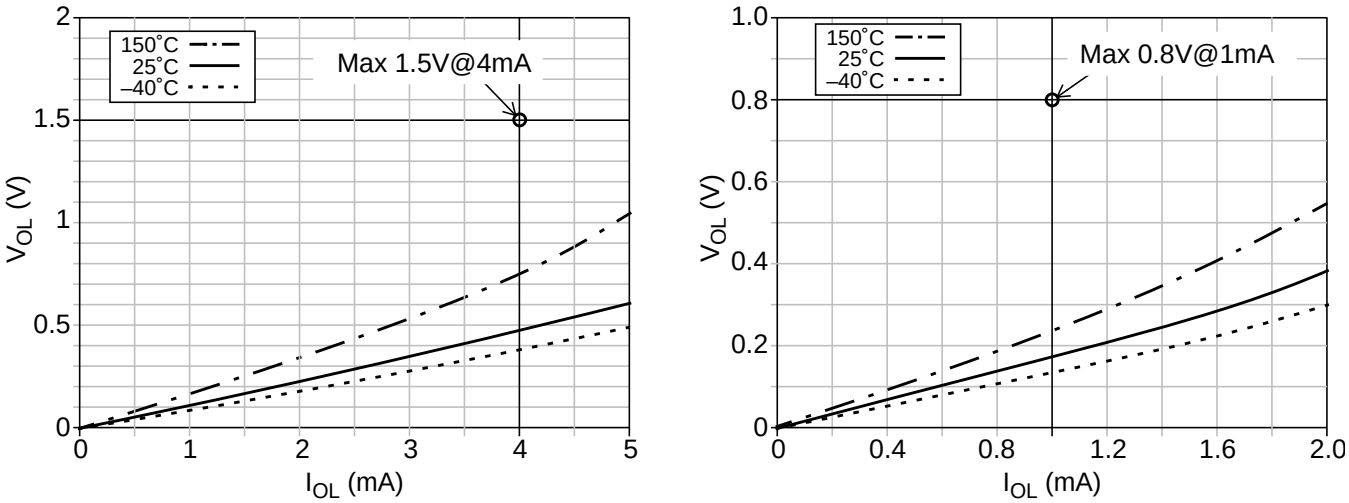
| Field | Description |
|---|---|
| 2<br>WS | **Wait or Stop Status** — When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host should issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands.<br>0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active)<br>1 Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode |
| 1<br>WSF | **Wait or Stop Failure Status** — This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.)<br>0 Memory access did not conflict with a wait or stop instruction<br>1 Memory access command failed because the CPU entered wait or stop mode |
| 0<br>DVF | **Data Valid Failure Status** — This status bit is not used in the MC9S08SG32 Series because it does not have any slow access memory.<br>0 Memory access did not conflict with a slow memory access<br>1 Memory access command failed because CPU was not finished with a slow memory access |

#### 17.4.1.2 BDC Breakpoint Match Register (BDCBKPT)

This 16-bit register holds the address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register but is not accessible to user programs because it is not located in the normal memory map of the MCU. Breakpoints are normally set while the target MCU is in active background mode before running the user application program. For additional information about setup and use of the hardware breakpoint logic in the BDC, refer to Section 17.2.4, “BDC Hardware Breakpoint.”

### 17.4.2 System Background Debug Force Reset Register (SBDFR)

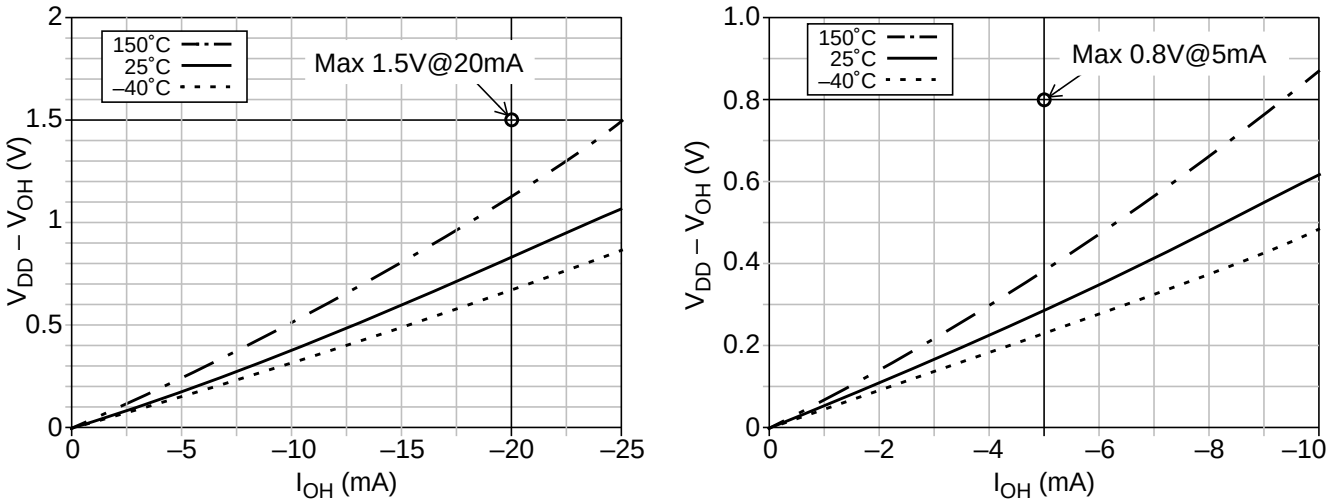
This register contains a single write-only control bit. A serial background mode command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.




a) $V_{DD}$ = 5V, Low Drive

b) $V_{DD}$ = 3V, Low Drive

**Figure A-2. Typical $V_{OL}$ vs $I_{OL}$, Low Drive Strength**




a) $V_{DD}$ = 5V, High Drive

b) $V_{DD}$ = 3V, High Drive

**Figure A-3. Typical $V_{DD} - V_{OH}$ vs $I_{OH}$, High Drive Strength**

**Table A-12. ADC Characteristics (continued)**

| # | Characteristic | Conditions | C | Symb | Min | Typ[1] | Max | Unit | Temp Rated | | Comment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | Standard | AEC Grade 0 | |
| | Full-scale error | 28-pin packages only | | | | | | | | | |
| | | 10-bit mode | T | $E_{FS}$ | 0 | ±0.5 | ±1 | LSB[2] | ♦ | ♦ | |
| | | 8-bit mode | | | 0 | ±0.5 | ±0.5 | LSB[2] | ♦ | ♦ | |
| | | 20-pin packages | | | | | | | | | |
| | | 10-bit mode | T | $E_{FS}$ | 0 | ±1.0 | ±1.5 | LSB[2] | ♦ | — | |
| | | 8-bit mode | | | 0 | ±0.5 | ±0.5 | LSB[2] | ♦ | — | |
| | | 16-pin packages | | | | | | | | | |
| | | 10-bit mode | T | $E_{FS}$ | 0 | ±1.0 | ±1.5 | LSB[2] | ♦ | ♦ | |
| | | 8-bit mode | | | 0 | ±0.5 | ±0.5 | LSB[2] | ♦ | ♦ | |
| | Quantization error | 10-bit mode | D | $E_Q$ | — | — | ±0.5 | LSB[2] | ♦ | ♦ | |
| | | 8-bit mode | | | — | — | ±0.5 | LSB[2] | ♦ | ♦ | |
| | Input leakage error | 10-bit mode | D | $E_{IL}$ | 0 | ±0.2 | ±2.5 | LSB[2] | ♦ | ♦ | Pad leakage[3] * $R_{AS}$ |
| | | 8-bit mode | | | 0 | ±0.1 | ±1 | LSB[2] | ♦ | ♦ | |
| | Temp sensor slope | -40°C to 25°C | D | m | — | 3.266 | — | mV/°C | ♦ | ♦ | |
| | | 25°C to 125°C | | | — | 3.638 | — | mV/°C | ♦ | ♦ | |
| | Temp sensor voltage | 25°C | D | $V_{TEMP25}$ | — | 1.396 | — | V | ♦ | ♦ | |

1 Typical values assume $V_{DD}$ = 5.0 V, Temp = 25°C, $f_{ADCK}$ = 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.

### B.1.1 Device Numbering Scheme

This device uses a smart numbering system. Refer to the following diagram to understand what each element of the device number represents.

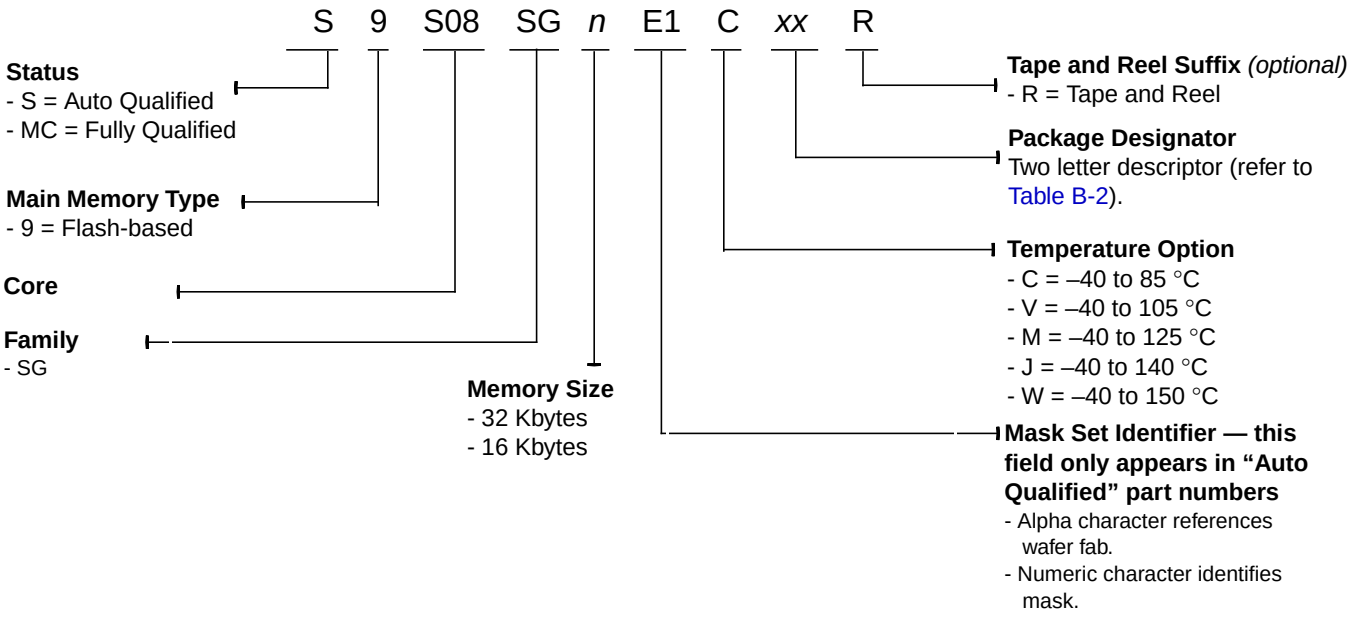



**Figure B-1. MC9S08SG32 Device Numbering Scheme**

## B.2 Package Information and Mechanical Drawings

Table B-2 provides the available package types and their document numbers. The latest package outline/mechanical drawings are available on the MC9S08SG32 Series Product Summary pages at http://www.freescale.com.

To view the latest drawing, either:

- Click on the appropriate link in Table B-2, or
- Open a browser to the Freescale® website (http://www.freescale.com), and enter the appropriate document number (from Table B-2) in the "Enter Keyword" search box at the top of the page.

## *How to Reach Us:*

**Home Page:**
www.freescale.com

**E-mail:**
support@freescale.com

**USA/Europe or Locations Not Listed:**
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

**Europe, Middle East, and Africa:**
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

**Japan:**
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

**Asia/Pacific:**
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

***For Literature Requests Only:***
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 1-303-675-2140
Fax: 1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com



MC9S08SG32
Rev. 8, 5/2010

freescale™
semiconductor