

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 | 48MHz |
| Connectivity | I ² C, LINbus, SCI, SPI, USB |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 51 |
| Program Memory Size | 60KB (60K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 12x12b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-QFP |
| Supplier Device Package | 64-QFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08jm60cqh |

[“Analog-to-Digital Converter \(S08ADC12V1\),”](#) for more information regarding the use of MCGERCLK with these modules.

- MCGFFCLK — This clock source is divided by 2 to generate FFCLK after being synchronized to the BUSCLK. It can be selected as clock source for the TPM modules. The frequency of the MCGFFCLK is determined by the settings of the MCG. See the [Section 12.4.7, “Fixed Frequency Clock,”](#) for details.
- LPO clock— This clock is generated from an internal Low Power Oscillator that is completely independent of the MCG module. The LPO clock can be selected as the clock source to the RTC or COP modules. See [Chapter 13, “Real-Time Counter \(S08RTCV1\),”](#) and [Section 5.4, “Computer Operating Properly \(COP\) Watchdog,”](#) for details on using the LPO clock with these modules.
- TPMCLK — TPMCLK is the optional external clock source for the TPM modules. The TPMCLK must be limited to 1/4th the frequency of the BUSCLK for synchronization. See [Chapter 16, “Timer/Pulse-Width Modulator \(S08TPMV3\),”](#) for more details.

Chapter 3

Modes of Operation

3.1 Introduction

The operating modes of the MC9S08JM60 series are described in this chapter. Entry into each mode, exit from each mode, and functionality while in each mode are described.

3.2 Features

- Active background mode for code development
- Wait mode:
 - CPU halts operation to conserve power
 - System clocks running
 - Full voltage regulation is maintained
- Stop modes: CPU and bus clocks stopped
 - Stop2: Partial power down of internal circuits; RAM and USB RAM contents retained
 - Stop3: All internal circuits powered for fast recovery; RAM, USB RAM, and register contents are retained

3.3 Run Mode

Run is the normal operating mode for the MC9S08JM60 series. This mode is selected upon the MCU exiting reset if the BKGD/MS pin is high. In this mode, the CPU executes code from internal memory with execution beginning at the address fetched from memory at 0xFFFFE:0xFFFF after reset.

3.4 Active Background Mode

The active background mode functions are managed through the background debug controller (BDC) in the HCS08 core. The BDC, together with the on-chip in-circuit emulator (ICE) debug module (DBG), provides the means for analyzing MCU operation during software development.

Active background mode is entered in any of five ways:

- When the BKGD/MS pin is low at the rising edge of reset
- When a BACKGROUND command is received through the BKGD pin
- When a BGND instruction is executed
- When encountering a BDC breakpoint
- When encountering a DBG breakpoint

3.6 Stop Modes

One of two stop modes is entered upon execution of a STOP instruction when STOPE in SOPT1 is set. In any stop mode, the bus and CPU clocks are halted. The MCG module can be configured to leave the reference clocks running. See [Chapter 12, “Multi-Purpose Clock Generator \(S08MCGV1\)”](#) for more information.

[Table 3-1](#) shows all of the control bits that affect stop mode selection and the mode selected under various conditions. The selected mode is entered following the execution of a STOP instruction.

Table 3-1. Stop Mode Selection

| STOPE | ENBDM ¹ | LVDE | LVDSE | PPDC | Stop Mode |
|-------|--------------------|---------------------|-------|------|--|
| 0 | x | x | x | x | Stop modes disabled; illegal opcode reset if STOP instruction executed |
| 1 | 1 | x | x | x | Stop3 with BDM enabled ² |
| 1 | 0 | Both bits must be 1 | x | x | Stop3 with voltage regulator active |
| 1 | 0 | Either bit a 0 | 0 | 0 | Stop3 |
| 1 | 0 | Either bit a 0 | 1 | 1 | Stop2 |

¹ ENBDM is located in the BDCSCR which is only accessible through BDC commands, see [Section 18.4.1.1, “BDC Status and Control Register \(BDCSCR\)”](#).

² When in stop3 mode with BDM enabled, The S_{IDD} will be near R_{IDD} levels because internal clocks are enabled.

3.6.1 Stop3 Mode

Stop3 mode is entered by executing a STOP instruction under the conditions as shown in [Table 3-1](#). The states of all of the internal registers and logic, RAM contents, and I/O pin states are maintained.

Stop3 can be exited by asserting $\overline{\text{RESET}}$, or by an interrupt from one of the following sources: the real-time clock (RTC) interrupt, the USB resume interrupt, LVD, ADC, IRQ, KBI, SCI, or the ACMP.

If stop3 is exited by means of the $\overline{\text{RESET}}$ pin, then the MCU is reset and operation will resume after taking the reset vector. Exit by means of one of the internal interrupt sources results in the MCU taking the appropriate interrupt vector.

3.6.1.1 LVD Enabled in Stop Mode

The LVD system is capable of generating either an interrupt or a reset when the supply voltage drops below the LVD voltage. If the LVD is enabled in stop (LVDE and LVDSE bits in SPMSC1 both set) at the time the CPU executes a STOP instruction, then the voltage regulator remains active during stop mode. If the user attempts to enter stop2 with the LVD enabled for stop, the MCU will enter stop3 instead.

For the ADC to operate, the LVD must be left enabled when entering stop3. For the ACMP to operate when ACGBS in ACMPSC is set, the LVD must be left enabled when entering stop3.

For the XOSC to operate with an external reference when RANGE in MCGC2 is set, the LVD must be left enabled when entering stop3.

writing to PPDACK, the pins will be controlled by their associated port control registers when the I/O latches are opened.

3.6.3 On-Chip Peripheral Modules in Stop Modes

When the MCU enters any stop mode, system clocks to the internal peripheral modules are stopped. Even in the exception case (ENBDM = 1), where clocks to the background debug logic continue to operate, clocks to the peripheral systems are halted to reduce power consumption. Refer to [Section 3.6.2, “Stop2 Mode,”](#) and [Section 3.6.1, “Stop3 Mode,”](#) for specific information on system behavior in stop modes.

Table 3-2. Stop Mode Behavior

| Peripheral | Mode | |
|---------------------------|----------------------------|----------------------------|
| | Stop2 | Stop3 |
| CPU | Off | Standby |
| RAM | Standby | Standby |
| Flash | Off | Standby |
| Parallel Port Registers | Off | Standby |
| ADC | Off | Optionally On ¹ |
| ACMP | Off | Optionally On ² |
| MCG | Off | Optionally On ³ |
| IIC | Off | Standby |
| RTC | Optionally on ⁴ | Optionally on ⁴ |
| SCI | Off | Standby |
| SPI | Off | Standby |
| TPM | Off | Standby |
| System Voltage Regulator | Off | Standby |
| XOSC | Off | Optionally On ⁵ |
| I/O Pins | States Held | States Held |
| USB (SIE and Transceiver) | Off | Optionally On ⁶ |
| USB 3.3-V Regulator | Off | Standby |
| USB RAM | Standby | Standby |

¹ Requires the asynchronous ADC clock and LVD to be enabled, else in standby.

² If ACGBS in ACMPSC is set, LVD must be enabled, else in standby.

³ IRCLKEN and IREFSTEN set in MCGC1, else in standby.

⁴ RTCPS[3:0] in RTCSC does not equal 0 before entering stop, else off.

⁵ ERCLKEN and EREFSTEN set in MCGC2, else in standby. For high frequency range (RANGE in MCGC2 set) requires the LVD to also be enabled in stop3.

⁶ USBEN in CTL is set and USBPHYEN in USBCTL0 is set, else off.

bootloader is protected, it remains intact even if MCU power is lost in the middle of an erase and reprogram operation.

4.5.7 Vector Redirection

Whenever any block protection is enabled, the reset and interrupt vectors will be protected. Vector redirection allows users to modify interrupt vector information without unprotecting bootloader and reset vector space. Vector redirection is enabled by programming the FNORED bit in the NVOPT register located at address 0xFFBF to zero. For redirection to occur, at least some portion but not all of the flash memory must be block protected by programming the NVPROT register located at address 0xFFBD. All of the interrupt vectors (memory locations 0xFFC0–0xFFFD) are redirected, though the reset vector (0xFFFFE:FFFF) is not.

For example, if 512 bytes of flash are protected, the protected address region is from 0xFE00 through 0xFFFF. The interrupt vectors (0xFFC0–0xFFFD) are redirected to the locations 0xFDC0–0xFDFD. Now, if a TPM1 overflow interrupt is taken for instance, the values in the locations 0xFDE0:FDE1 are used for the vector instead of the values in the locations 0xFFE0:FFE1. This allows the user to reprogram the unprotected portion of the flash with new program code including new interrupt vector values while leaving the protected area, which includes the default vector locations, unchanged.

4.6 Security

The MC9S08JM60 Series includes circuitry to prevent unauthorized access to the contents of flash and RAM memory. When security is engaged, flash and RAM are considered secure resources. Direct-page registers, high-page registers, and the background debug controller are considered unsecured resources. Programs executing within secure memory have normal access to any MCU memory locations and resources. Attempts to access a secure memory location with a program executing from an unsecured memory space or through the background debug interface are blocked (writes are ignored and reads return all 0s).

Security is engaged or disengaged based on the state of two nonvolatile register bits (SEC01:SEC00) in the FOPT register. During reset, the contents of the nonvolatile location NVOPT are copied from flash into the working FOPT register in high-page register space. A user engages security by programming the NVOPT location which can be done at the same time the flash memory is programmed. The 1:0 state disengages security and the other three combinations engage security. Notice the erased state (1:1) makes the MCU secure. During development, whenever the flash is erased, it is good practice to immediately program the SEC00 bit to 0 in NVOPT so SEC01:SEC00 = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate background debug controller can still be used for background memory access commands, but the MCU cannot enter active background mode except by holding BKGD/MS low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there

6.5.7 Port D I/O Registers (PTDD and PTDDD)

Port D parallel I/O function is controlled by the registers listed below.

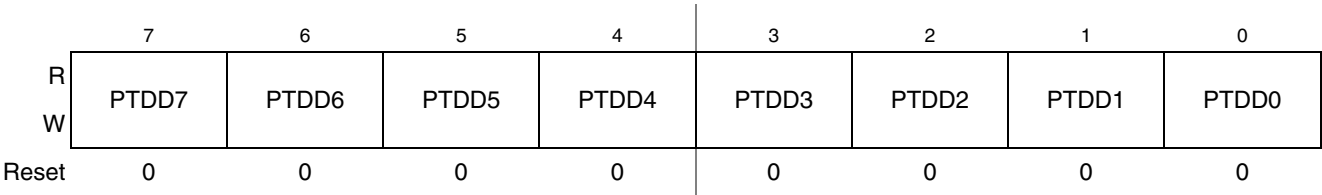


Figure 6-17. Port D Data Register (PTDD)

Table 6-16. PTDD Register Field Descriptions

| Field | Description |
|------------------|--|
| 7:0 PTDD[7:0] | Port D Data Register Bits — For port D pins that are inputs, reads return the logic level on the pin. For port D pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port D pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTDD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled. |

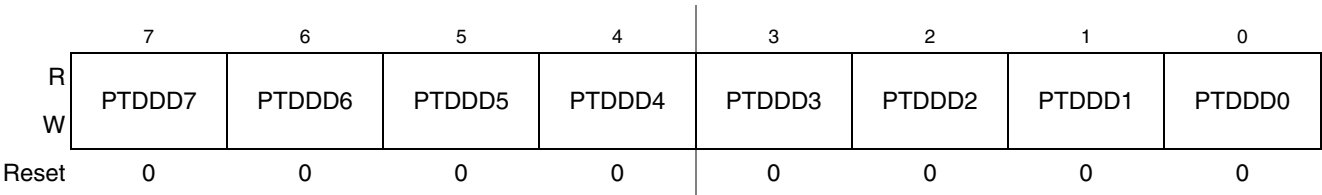


Figure 6-18. Data Direction for Port D (PTDDD)

Table 6-17. PTDDD Register Field Descriptions

| Field | Description |
|-------------------|--|
| 7:0 PTDDD[7:0] | Data Direction for Port D Bits — These read/write bits control the direction of port D pins and what is read for PTDD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port D bit n and PTDD reads return the contents of PTDDn. |

6.5.8 Port D Pin Control Registers (PTDPE, PTDSE, PTDDS)

In addition to the I/O control, port D pins are controlled by the registers listed below.

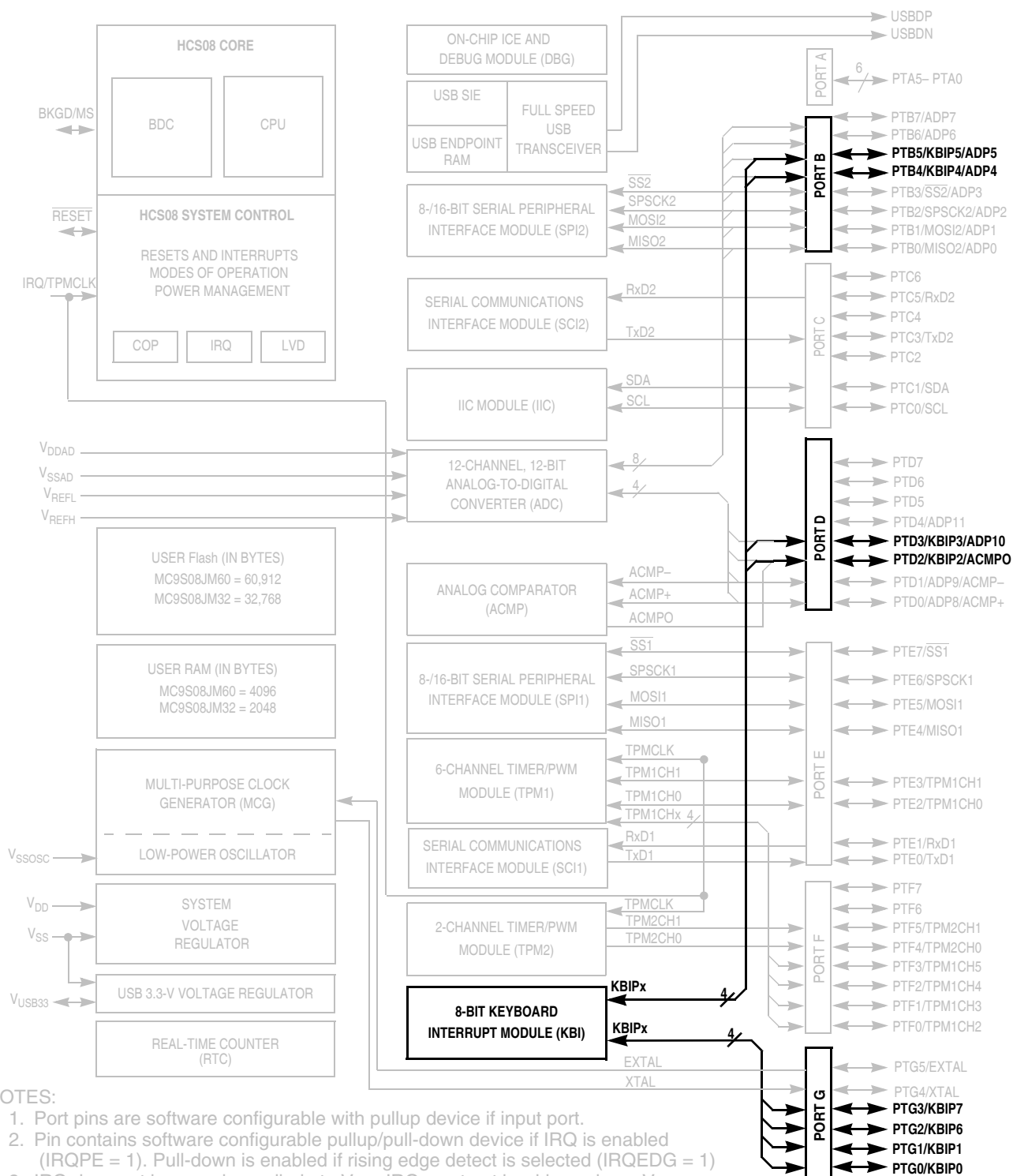


Figure 9-1. MC9S08JM60 Series Block Diagram Highlighting KBI Block and Pins

Table 10-6. ADCCFG Register Field Descriptions (continued)

| Field | Description |
|---------------|---|
| 3:2 MODE | Conversion Mode Selection. MODE bits are used to select between 12-, 10-, or 8-bit operation. See Table 10-8 . |
| 1:0 ADICLK | Input Clock Select. ADICLK bits select the input clock source to generate the internal clock ADCK. See Table 10-9 . |

Table 10-7. Clock Divide Select

| ADIV | Divide Ratio | Clock Rate |
|------|--------------|-----------------|
| 00 | 1 | Input clock |
| 01 | 2 | Input clock ÷ 2 |
| 10 | 4 | Input clock ÷ 4 |
| 11 | 8 | Input clock ÷ 8 |

Table 10-8. Conversion Modes

| MODE | Mode Description |
|------|--------------------------|
| 00 | 8-bit conversion (N=8) |
| 01 | 12-bit conversion (N=12) |
| 10 | 10-bit conversion (N=10) |
| 11 | Reserved |

Table 10-9. Input Clock Select

| ADICLK | Selected Clock Source |
|--------|----------------------------|
| 00 | Bus clock |
| 01 | Bus clock divided by 2 |
| 10 | Alternate clock (ALTCLK) |
| 11 | Asynchronous clock (ADACK) |

10.3.8 Pin Control 1 Register (APCTL1)

The pin control registers disable the I/O port control of MCU pins used as analog inputs. APCTL1 is used to control the pins associated with channels 0–7 of the ADC module.

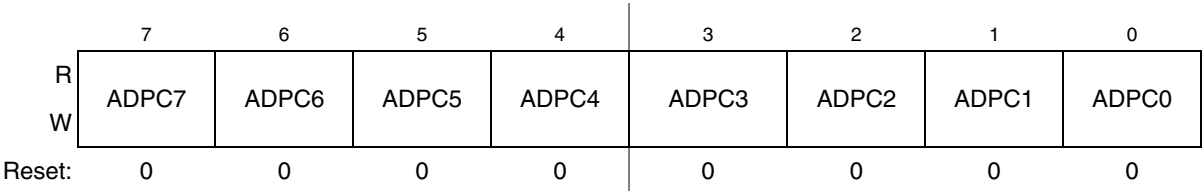


Figure 10-10. Pin Control 1 Register (APCTL1)

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

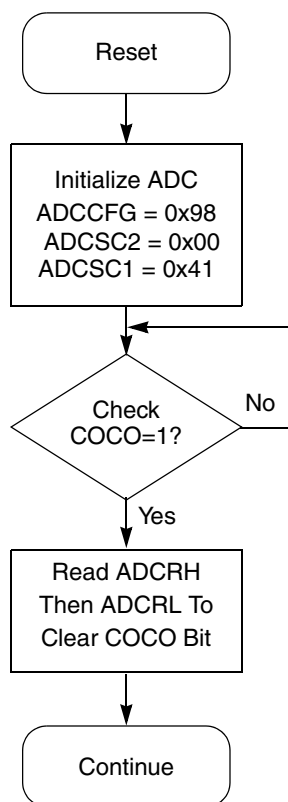


Figure 10-13. Initialization Flowchart for Example

10.6 Application Information

This section contains information for using the ADC module in applications. The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an A/D converter.

10.6.1 External Pins and Routing

The following sections discuss the external pins associated with the ADC module and how they should be used for best results.

12.3 Register Definition

12.3.1 MCG Control Register 1 (MCGC1)

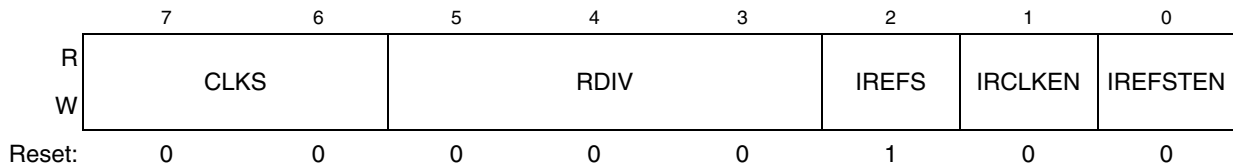


Figure 12-3. MCG Control Register 1 (MCGC1)

Table 12-1. MCG Control Register 1 Field Descriptions

| Field | Description |
|---------------|--|
| 7:6 CLKS | Clock Source Select — Selects the system clock source. 00 Encoding 0 — Output of FLL or PLL is selected. 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Reserved, defaults to 00. |
| 5:3 RDIV | Reference Divider — Selects the amount to divide down the reference clock selected by the IREFS bit. If the FLL is selected, the resulting frequency must be in the range 31.25 kHz to 39.0625 kHz. If the PLL is selected, the resulting frequency must be in the range 1 MHz to 2 MHz. 000 Encoding 0 — Divides reference clock by 1 (reset default) 001 Encoding 1 — Divides reference clock by 2 010 Encoding 2 — Divides reference clock by 4 011 Encoding 3 — Divides reference clock by 8 100 Encoding 4 — Divides reference clock by 16 101 Encoding 5 — Divides reference clock by 32 110 Encoding 6 — Divides reference clock by 64 111 Encoding 7 — Divides reference clock by 128 |
| 2 IREFS | Internal Reference Select — Selects the reference clock source. 1 Internal reference clock selected 0 External reference clock selected |
| 1 IRCLKEN | Internal Reference Clock Enable — Enables the internal reference clock for use as MCGIRCLK. 1 MCGIRCLK active 0 MCGIRCLK inactive |
| 0 IREFSTEN | Internal Reference Stop Enable — Controls whether or not the internal reference clock remains enabled when the MCG enters stop mode. 1 Internal reference clock stays enabled in stop if IRCLKEN is set or if MCG is in FEI, FBI, or BLPI mode before entering stop 0 Internal reference clock is disabled in stop |

12.3.4 MCG Status and Control Register (MCGSC)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|------|-------|--------|-------|---|---------|-------|
| R | LOLS | LOCK | PLLST | IREFST | CLKST | | OSCINIT | FTRIM |
| W | | | | | | | | |
| POR: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | U |

Figure 12-6. MCG Status and Control Register (MCGSC)

Table 12-4. MCG Status and Control Register Field Descriptions

| Field | Description |
|--------------|--|
| 7 LOLS | <p>Loss of Lock Status — This bit is a sticky indication of lock status for the FLL or PLL. LOLS is set when lock detection is enabled and after acquiring lock, the FLL or PLL output frequency has fallen outside the lock exit frequency tolerance, D_{unl}. LOLIE determines whether an interrupt request is made when set. LOLS is cleared by reset or by writing a logic 1 to LOLS when LOLS is set. Writing a logic 0 to LOLS has no effect.</p> <p>0 FLL or PLL has not lost lock since LOLS was last cleared. 1 FLL or PLL has lost lock since LOLS was last cleared.</p> |
| 6 LOCK | <p>Lock Status — Indicates whether the FLL or PLL has acquired lock. Lock detection is disabled when both the FLL and PLL are disabled. If the lock status bit is set then changing the value of any of the following bits IREFS, PLLS, RDIV[2:0], TRIM[7:0] (if in FEI or FBI modes), or VDIV[3:0] (if in PBE or PEE modes), will cause the lock status bit to clear and stay cleared until the FLL or PLL has reacquired lock. Stop mode entry will also cause the lock status bit to clear and stay cleared until the FLL or PLL has reacquired lock. Entry into BLPI or BLPE mode will also cause the lock status bit to clear and stay cleared until the MCG has exited these modes and the FLL or PLL has reacquired lock.</p> <p>0 FLL or PLL is currently unlocked. 1 FLL or PLL is currently locked.</p> |
| 5 PLLST | <p>PLL Select Status — The PLLST bit indicates the current source for the PLLS clock. The PLLST bit does not update immediately after a write to the PLLS bit due to internal synchronization between clock domains.</p> <p>0 Source of PLLS clock is FLL clock. 1 Source of PLLS clock is PLL clock.</p> |
| 4 IREFST | <p>Internal Reference Status — The IREFST bit indicates the current source for the reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains.</p> <p>0 Source of reference clock is external reference clock (oscillator or external clock source as determined by the IREFS bit in the MCGC2 register). 1 Source of reference clock is internal reference clock.</p> |
| 3:2 CLKST | <p>Clock Mode Status — The CLKST bits indicate the current clock mode. The CLKST bits do not update immediately after a write to the CLKS bits due to internal synchronization between clock domains.</p> <p>00 Encoding 0 — Output of FLL is selected. 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Output of PLL is selected.</p> |

external crystal and a maximum reference divider factor of 128, the resulting frequency of the reference clock for the FLL is 62.5 kHz (greater than the 39.0625 kHz maximum allowed).

Care must be taken in the software to minimize the amount of time spent in this state where the FLL is operating in this condition.

The following code sequence describes how to move from FEI mode to PEE mode until the 8 MHz crystal reference frequency is set to achieve a bus frequency of 8 MHz. Because the MCG is in FEI mode out of reset, this example also shows how to initialize the MCG for PEE mode out of reset. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, FEI must transition to FBE mode:
 - a) MCGC2 = 0x36 (%00110110)
 - BDIV (bits 7 and 6) set to %00, or divide-by-1
 - RANGE (bit 5) set to 1 because the frequency of 8 MHz is within the high frequency range
 - HGO (bit 4) set to 1 to configure external oscillator for high gain operation
 - EREFS (bit 2) set to 1, because a crystal is being used
 - ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active
 - b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.
 - c) Block Interrupts (If applicable by setting the interrupt bit in the CCR).
 - d) MCGC1 = 0xB8 (%10111000)
 - CLKS (bits 7 and 6) set to %10 in order to select external reference clock as system clock source
 - RDIV (bits 5-3) set to %111, or divide-by-128.

NOTE

8 MHz / 128 = 62.5 kHz which is greater than the 31.25 kHz to 39.0625 kHz range required by the FLL. Therefore after the transition to FBE is complete, software must progress through to BLPE mode immediately by setting the LP bit in MCGC2.

- IREFS (bit 2) cleared to 0, selecting the external reference clock
- e) Loop until IREFST (bit 4) in MCGSC is 0, indicating the external reference is the current source for the reference clock
 - f) Loop until CLKST (bits 3 and 2) in MCGSC are %10, indicating that the external reference clock is selected to feed MCGOUT
2. Then, FBE mode transitions into BLPE mode:
 - a) MCGC2 = 0x3E (%00111110)
 - LP (bit 3) in MCGC2 to 1 (BLPE mode entered)

NOTE

There must be no extra steps (including interrupts) between steps 1d and 2a.

- b) Enable Interrupts (if applicable by clearing the interrupt bit in the CCR).

When the SPI is configured as a master, data queued in the transmit data buffer is transmitted immediately after the previous transmission has completed.

The SPI transmit buffer empty flag (SPTEF) in the SPIxS register indicates when the transmit data buffer is ready to accept new data. SPIxS must be read when SPTEF is set before writing to the SPI data registers, or the write will be ignored.

Data may be read from SPIxDH:SPIxDL any time after SPRF is set and before another transfer is finished. Failure to read the data out of the receive data buffer before a new transfer ends causes a receive overrun condition and the data from the new transfer is lost.

In 8-bit mode, only SPIxDL is available. Reads of SPIxDH will return all 0s. Writes to SPIxDH will be ignored.

In 16-bit mode, reading either byte (SPIxDH or SPIxDL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. Writing to either byte (SPIxDH or SPIxDL) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the transmit data buffer.

15.3.6 SPI Match Registers (SPIxMH:SPIxML)

These read/write registers contain the hardware compare value, which sets the SPI match flag (SPMF) when the value received in the SPI receive data buffer equals the value in the SPIxMH:SPIxML registers.

In 8-bit mode, only SPIxML is available. Reads of SPIxMH will return all 0s. Writes to SPIxMH will be ignored.

In 16-bit mode, reading either byte (SPIxMH or SPIxML) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. Writing to either byte (SPIxMH or SPIxML) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent value into the SPI match registers.

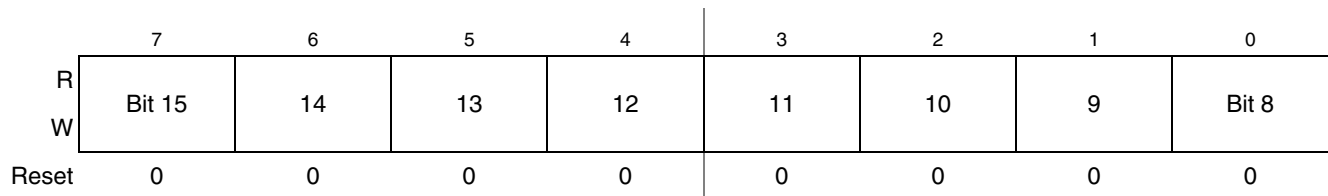


Figure 15-11. SPI Match Register High (SPIxMH)

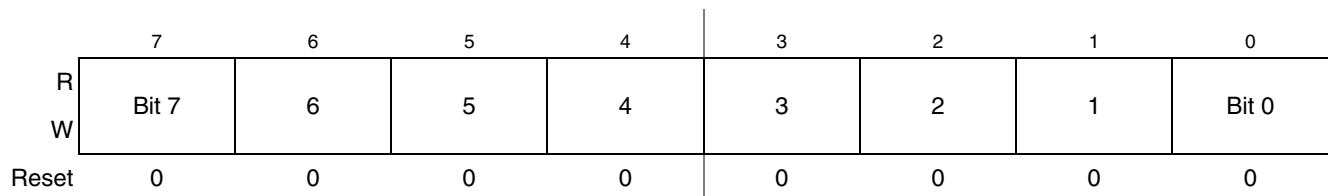


Figure 15-12. SPI Match Register Low (SPIxML)

16.2.1.1 EXTCLK — External Clock Source

Control bits in the timer status and control register allow the user to select nothing (timer disable), the bus-rate clock (the normal default source), a crystal-related clock, or an external clock as the clock which drives the TPM prescaler and subsequently the 16-bit TPM counter. The external clock source is synchronized in the TPM. The bus clock clocks the synchronizer; the frequency of the external source must be no more than one-fourth the frequency of the bus-rate clock, to meet Nyquist criteria and allowing for jitter.

The external clock signal shares the same pin as a channel I/O pin, so the channel pin will not be usable for channel I/O function when selected as the external clock source. It is the user's responsibility to avoid such settings. If this pin is used as an external clock source (CLKSB:CLKSA = 1:1), the channel can still be used in output compare mode as a software timer (ELSnB:ELSnA = 0:0).

16.2.1.2 TPMxCHn — TPM Channel n I/O Pin(s)

Each TPM channel is associated with an I/O pin on the MCU. The function of this pin depends on the channel configuration. The TPM pins share with general purpose I/O pins, where each pin has a port data register bit, and a data direction control bit, and the port has optional passive pullups which may be enabled whenever a port pin is acting as an input.

The TPM channel does not control the I/O pin when (ELSnB:ELSnA = 0:0) or when (CLKSB:CLKSA = 0:0) so it normally reverts to general purpose I/O control. When CPWMS = 1 (and ELSnB:ELSnA not = 0:0), all channels within the TPM are configured for center-aligned PWM and the TPMxCHn pins are all controlled by the TPM system. When CPWMS=0, the MSnB:MSnA control bits determine whether the channel is configured for input capture, output compare, or edge-aligned PWM.

When a channel is configured for input capture (CPWMS=0, MSnB:MSnA = 0:0 and ELSnB:ELSnA not = 0:0), the TPMxCHn pin is forced to act as an edge-sensitive input to the TPM. ELSnB:ELSnA control bits determine what polarity edge or edges will trigger input-capture events. A synchronizer based on the bus clock is used to synchronize input edges to the bus clock. This implies the minimum pulse width—that can be reliably detected—on an input capture pin is four bus clock periods (with ideal clock pulses as near as two bus clocks can be detected). TPM uses this pin as an input capture input to override the port data and data direction controls for the same pin.

When a channel is configured for output compare (CPWMS=0, MSnB:MSnA = 0:1 and ELSnB:ELSnA not = 0:0), the associated data direction control is overridden, the TPMxCHn pin is considered an output controlled by the TPM, and the ELSnB:ELSnA control bits determine how the pin is controlled. The remaining three combinations of ELSnB:ELSnA determine whether the TPMxCHn pin is toggled, cleared, or set each time the 16-bit channel value register matches the timer counter.

When the output compare toggle mode is initially selected, the previous value on the pin is driven out until the next output compare event—then the pin is toggled.

Table 17-7. REV Field Descriptions

| Field | Description |
|-----------------|--|
| 8–0 REV[7:0] | Revision — Revision number of the USB module. |

17.3.5 Interrupt Status Register (INTSTAT)

The INTSTAT contains bits for each of the interrupt source within the USB module. Each of these bits is qualified with its respective interrupt enable bits (see the interrupt enable register). All bits of the register are logically OR'ed together to form a single interrupt source for the microcontroller. Once an interrupt bit has been set, it may only be cleared by writing a 1 to the respective interrupt bit. This register will contain the value of 0x00 after a reset.

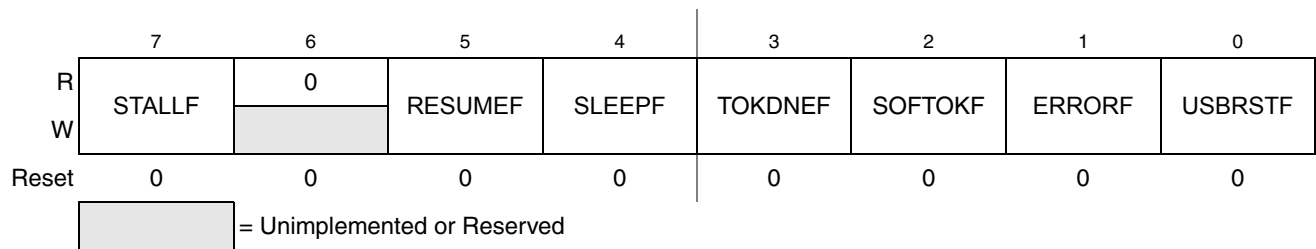


Figure 17-8. Interrupt Status Register (INTSTAT)

Table 17-9. INTSTAT Field Descriptions

| Field | Description |
|--------------|--|
| 7 STALLF | Stall Flag — The stall interrupt is used in device mode. In device mode the stall flag is asserted when a STALL handshake is sent by the serial interface engine (SIE). 0 A STALL handshake has not been sent 1 A STALL handshake has been sent |
| 5 RESUMEF | Resume Flag — This bit is set 2.5 μ s after clocks to the USB module have restarted following resume signaling. It can be used to indicate remote wakeup signaling on the USB bus. This interrupt is enabled only when the USB module is about to enter suspend mode (usually when SLEEPF interrupt detected). 0 No RESUME observed 1 RESUME detected (K-state is observed on the USBDP/USBDN signals for 2.5 μ s) |
| 4 SLEEPF | Sleep Flag — This bit is set if the USB module has detected a constant idle on the USB bus for 3 ms, indicating that the USB module will go into suspend mode. The sleep timer is reset by activity on the USB bus. 0 No constant idle state of 3 ms has been detected on the USB bus 1 A constant idle state of 3 ms has been detected on the USB bus |
| 3 TOKDNEF | Token Complete Flag — This bit is set when the current transaction is completed. The firmware must immediately read the STAT register to determine the endpoint and BD information. Clearing this bit (by setting it to 1) causes the STAT register to be cleared or the STAT FIFO holding register to be loaded into the STAT register. 0 No tokens being processed are complete 1 Current token being processed is complete |
| 2 SOFTOKF | SOF Token Flag — This bit is set if the USB module has received a start of frame (SOF) token. 0 The USB module has not received an SOF token 1 The USB module has received an SOF token |

17.4.2 Buffer Descriptor Table (BDT)

To efficiently manage USB endpoint communications, the USB module implements a buffer descriptor table (BDT) comprised of buffer descriptors (BD) in the local USB RAM. The BD entries provide status or control information for a corresponding endpoint. The BD entries also provide an address to the endpoint's buffer. A single BD for an endpoint direction requires 3-bytes. A detailed description of the BDT format is provided in the next sections.

The software API intelligently manages buffers for the USB module by updating the BDT when needed. This allows the USB module to efficiently handle data transmission and reception, while the microcontroller performs communication overhead processing and other function dependent applications.

Because the buffers are shared between the microcontroller and the USB module, a simple semaphore mechanism is used to distinguish who is allowed to update the BDT and buffers in buffer memory. A semaphore bit, the OWN bit, is cleared to 0 when the BD entry is owned by the microcontroller. The microcontroller is allowed read and write access to the BD entry and the data buffer when the OWN bit is 0. When the OWN bit is set to 1, the BD entry and the data buffer are owned by the USB module. The USB module now has full read and write access and the microcontroller must not modify the BD or its corresponding data buffer.

17.4.2.1 Multiple Buffer Descriptor Table Entries for a Single Endpoint

Every endpoint direction requires at least one three-byte Buffer Descriptor entry. Thus, endpoint 0, a bidirectional control endpoint, requires one BDT entry for the IN direction, and one for the OUT direction.

Using two BD entries also allows for double-buffering. Double-buffering BDs allows the USB module to easily transfer data at the maximum throughput provided by the USB module. Double buffering allows the MCU to process one BD while the USB module is processing the other BD.

To facilitate double-buffering, two buffer descriptor (BD) entries are needed for each endpoint direction. One BD entry is the EVEN BD and the other is the ODD BD.

17.4.2.2 Addressing Buffer Descriptor Table Entries

The BDT addressing is hardwired into the module. The BDT occupies the first portion of the USB RAM. To access endpoint data via the USB or MCU, the addressing mechanism of the buffer descriptor table must be understood.

All enabled IN and OUT endpoint BD entries are indexed into the BDT to allow easy access via the USB module or the MCU. The figure below shows the USB RAM organization. The figure shows that the first entries in the USB RAM are dedicated to storage of the BDT entries - i.e. the first 30 bytes of the USB RAM (0x00 to 0x1D) are used to implement the BDT.

18.2.3 BDC Commands

BDC commands are sent serially from a host computer to the BKGD pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. Active background mode commands require that the target MCU is currently in the active background mode while non-intrusive commands may be issued at any time whether the target MCU is in active background mode or running a user application program.

[Table 18-1](#) shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

Coding Structure Nomenclature

This nomenclature is used in [Table 18-1](#) to describe the coding structure of the BDC commands.

| | |
|------|--|
| | Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first) |
| / | = separates parts of the command |
| d | = delay 16 target BDC clock cycles |
| AAAA | = a 16-bit address in the host-to-target direction |
| RD | = 8 bits of read data in the target-to-host direction |
| WD | = 8 bits of write data in the host-to-target direction |
| RD16 | = 16 bits of read data in the target-to-host direction |
| WD16 | = 16 bits of write data in the host-to-target direction |
| SS | = the contents of BDCSCR in the target-to-host direction (STATUS) |
| CC | = 8 bits of write data for BDCSCR in the host-to-target direction (CONTROL) |
| RBKP | = 16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register) |
| WBKP | = 16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register) |

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

18.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

Table 18-2. BDCSCR Register Field Descriptions (continued)

| Field | Description |
|----------|--|
| 2 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. 0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active) 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 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.) 0 Memory access did not conflict with a wait or stop instruction 1 Memory access command failed because the CPU entered wait or stop mode |
| 0 DVF | Data Valid Failure Status — This status bit is not used in the MC9S08JM60 Series because it does not have any slow access memory. 0 Memory access did not conflict with a slow memory access 1 Memory access command failed because CPU was not finished with a slow memory access |


18.4.1.2 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 18.2.4, “BDC Hardware Breakpoint.”](#)

18.4.2 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.

| | | | | | | | | |
|-------|---|---|---|---|---|---|---|-------------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | BDFR ¹ |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

 = Unimplemented or Reserved

¹ BDFR is writable only through serial background mode debug commands, not from user programs.

Figure 18-6. System Background Debug Force Reset Register (SBDFR)

Appendix B

Ordering Information and Mechanical Drawings

B.1 Ordering Information

This section contains ordering numbers for MC9S08JM60 series devices. See below for an example of the device numbering system.

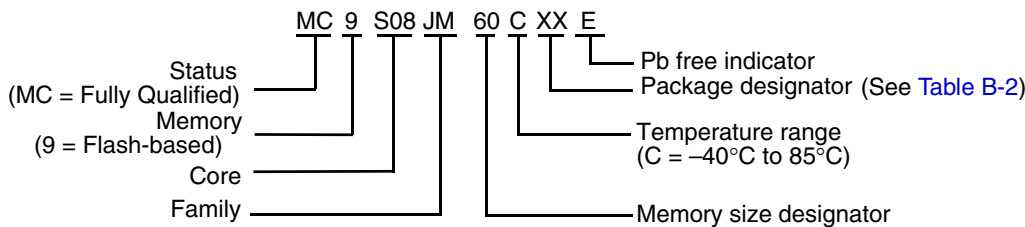
Table B-1. Device Numbering System

| Device Number ¹ | Memory | | Available Packages ² |
|----------------------------|--------|------|--|
| | Flash | RAM | Type |
| MC9S08JM60 | 60,912 | 4096 | 64-pin LQFP 64-pin QFP 48-pin QFN 44-pin LQFP |
| MC9S08JM32 | 32,768 | 2048 | |

¹ See [Table 1-1](#) for a complete description of modules included on each device.

² See [Table B-2](#) for package information.

B.2 Orderable Part Numbering System



B.3 Mechanical Drawings

The following pages contain mechanical specifications for MC9S08JM60 series package options. See [Table B-2](#) for the document numbers that correspond to each package type.

Table B-2. Package Information

| Pin Count | Type | Designator | Document No. |
|-----------|------|------------|--------------|
| 44 | LQFP | LD | 98ASS23225W |
| 48 | QFN | GT | 98ARH99048A |
| 64 | LQFP | LH | 98ASS23234W |
| 64 | QFP | QH | 98ASB42844B |