



Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	16MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	4
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08qd2vpc">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08qd2vpc</a>

the CPU executes a STOP instruction, the MCU will not enter either of the stop modes and an illegal opcode reset is forced. The stop modes are selected by setting the appropriate bits in SPMSC2.

HCS08 devices that are designed for low voltage operation (1.8V to 3.6V) also include stop1 mode. The MC9S08QD4 series does not include stop1 mode.

Table 3-1 summarizes the behavior of the MCU in each of the stop modes.

**Table 3-1. Stop Mode Behavior**

Mode	PPDC	CPU, Digital Peripherals, Flash	RAM	ICS	ADC1	Regulator	I/O Pins	RTI
Stop2	1	Off	Standby	Off	Disabled	Standby	States held	Optionally on
Stop3	0	Standby	Standby	Off <sup>1</sup>	Optionally on	Standby	States held	Optionally on

<sup>1</sup> ICS can be configured to run in stop3. Please see the ICS registers.

### 3.6.1 Stop2 Mode

The stop2 mode provides very low standby power consumption and maintains the contents of RAM and the current state of all of the I/O pins. To enter stop2, the user must execute a STOP instruction with stop2 selected (PPDC = 1) and stop mode enabled (STOPE = 1). In addition, the LVD must not be enabled to operate in stop (LVDSE = 0 or LVDE = 0). If the LVD is enabled in stop, then the MCU enters stop3 upon the execution of the STOP instruction regardless of the state of PPDC.

Before entering stop2 mode, the user must save the contents of the I/O port registers, as well as any other memory-mapped registers which they want to restore after exit of stop2, to locations in RAM. Upon exit of stop2, these values can be restored by user software before pin latches are opened.

When the MCU is in stop2 mode, all internal circuits that are powered from the voltage regulator are turned off, except for the RAM. The voltage regulator is in a low-power standby state, as is the ADC. Upon entry into stop2, the states of the I/O pins are latched. The states are held while in stop2 mode and after exiting stop2 mode until a logic 1 is written to PPDACK in SPMSC2.

Exit from stop2 is done by asserting either of the wake-up pins:  $\overline{\text{RESET}}$  or IRQ, or by an RTI interrupt. IRQ is always an active low input when the MCU is in stop2, regardless of how it was configured before entering stop2.

#### NOTE

Although this IRQ pin is automatically configured as active low input, the pullup associated with the IRQ pin is not automatically enabled. Therefore, if an external pullup is not used, the internal pullup must be enabled by setting IRQPE in IRQSC.

Upon wake-up from stop2 mode, the MCU will start up as from a power-on reset (POR) except pin states remain latched. The CPU will take the reset vector. The system and all peripherals will be in their default reset states and must be initialized.

## 5.8 Reset, Interrupt, and System Control Registers and Control Bits

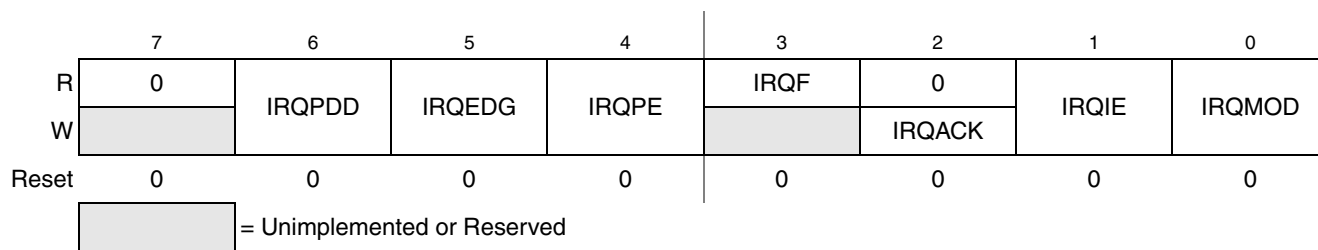
One 8-bit register in the direct page register space and eight 8-bit registers in the high-page register space are related to reset and interrupt systems.

Refer to the direct-page register summary in [Chapter 3, “Modes of Operation,”](#) for the absolute address assignments for all 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.

Some control bits in the SOPT1, SOPT2 and SPMSC2 registers are related to modes of operation. Although brief descriptions of these bits are provided here, the related functions are discussed in greater detail in [Chapter 3, “Modes of Operation.”](#)

### 5.8.1 Interrupt Pin Request Status and Control Register (IRQSC)

This direct page register includes status and control bits which are used to configure the IRQ function, report status, and acknowledge IRQ events.



**Figure 5-2. Interrupt Request Status and Control Register (IRQSC)**

**Table 5-3. IRQSC Register Field Descriptions**

Field	Description
6 IRQPDD	<b>Interrupt Request (IRQ) Pull Device Disable</b> — This read/write control bit is used to disable the internal pullup/pulldown device when the IRQ pin is enabled (IRQPE = 1) allowing for an external device to be used. 0 IRQ pull device enabled if IRQPE = 1. 1 IRQ pull device disabled if IRQPE = 1.
5 IRQEDG	<b>Interrupt Request (IRQ) Edge Select</b> — This read/write control bit is used to select the polarity of edges or levels on the IRQ pin that cause IRQF to be set. The IRQMOD control bit determines whether the IRQ pin is sensitive to both edges and levels or only edges. When the IRQ pin is enabled as the IRQ input and is configured to detect rising edges. When IRQEDG = 1 and the internal pull device is enabled, the pullup device is reconfigured as an optional pulldown device. 0 IRQ is falling edge or falling edge/low-level sensitive. 1 IRQ is rising edge or rising edge/high-level sensitive.
4 IRQPE	<b>IRQ Pin Enable</b> — This read/write control bit enables the IRQ pin function. When this bit is set the IRQ pin can be used as an interrupt request. 0 IRQ pin function is disabled. 1 IRQ pin function is enabled.
3 IRQF	<b>IRQ Flag</b> — This read-only status bit indicates when an interrupt request event has occurred. 0 No IRQ request. 1 IRQ event detected.

Table 5-3. IRQSC Register Field Descriptions (continued)

Field	Description
2 IRQACK	<b>IRQ Acknowledge</b> — This write-only bit is used to acknowledge interrupt request events (write 1 to clear IRQF). Writing 0 has no meaning or effect. Reads always return 0. If edge-and-level detection is selected (IRQMOD = 1), IRQF cannot be cleared while the IRQ pin remains at its asserted level.
1 IRQIE	<b>IRQ Interrupt Enable</b> — This read/write control bit determines whether IRQ events generate an interrupt request. 0 Interrupt request when IRQF set is disabled (use polling). 1 Interrupt requested whenever IRQF = 1.
0 IRQMOD	<b>IRQ Detection Mode</b> — This read/write control bit selects either edge-only detection or edge-and-level detection. The IRQEDG control bit determines the polarity of edges and levels that are detected as interrupt request events. See <a href="#">Section 5.5.2.2, “Edge and Level Sensitivity,”</a> for more details. 0 IRQ event on falling edges or rising edges only. 1 IRQ event on falling edges and low levels or on rising edges and high levels.

## 5.8.2 System Reset Status Register (SRS)

This high-page register includes read-only status flags to indicate the source of the most recent reset. When a debug host forces reset by writing 1 to BDFR in the SBDPFR register, all of the status bits in SRS will be cleared. Writing any value to this register address clears the COP watchdog timer without affecting the contents of this register. The reset state of these bits depends on what caused the MCU to reset.

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

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

Figure 5-3. System Reset Status (SRS)

Table 5-4. SRS Register Field Descriptions

Field	Description
7 POR	<b>Power-On Reset</b> — Reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVR) status bit is also set to indicate that the reset occurred while the internal supply was below the LVR threshold. 0 Reset not caused by POR. 1 POR caused reset.
6 PIN	<b>External Reset Pin</b> — Reset was caused by an active-low level on the external reset pin. 0 Reset not caused by external reset pin. 1 Reset came from external reset pin.

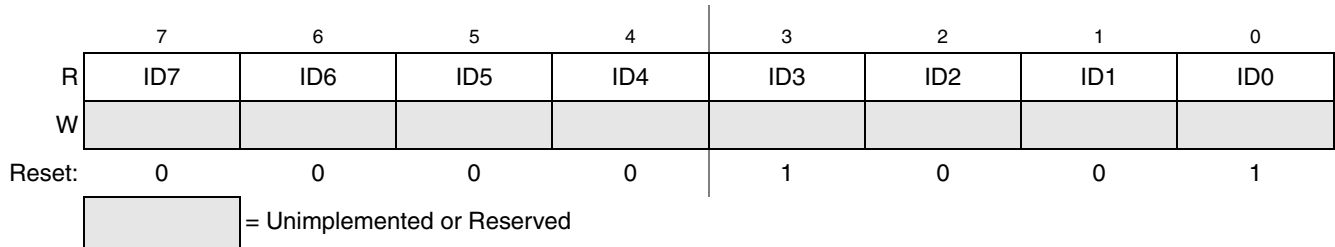


Figure 5-8. System Device Identification Register — Low (SDIDL)

Table 5-9. SDIDL Register Field Descriptions

Field	Description
7:0 ID[7:0]	<b>Part Identification Number</b> — Each derivative in the HCS08 family has a unique identification number. The MC9S08QD4 series is hard coded to the value 0x011. See also ID bits in <a href="#">Table 5-8</a> .

### 5.8.7 System Real-Time Interrupt Status and Control Register (SRTISC)

This high-page register contains status and control bits for the RTI.

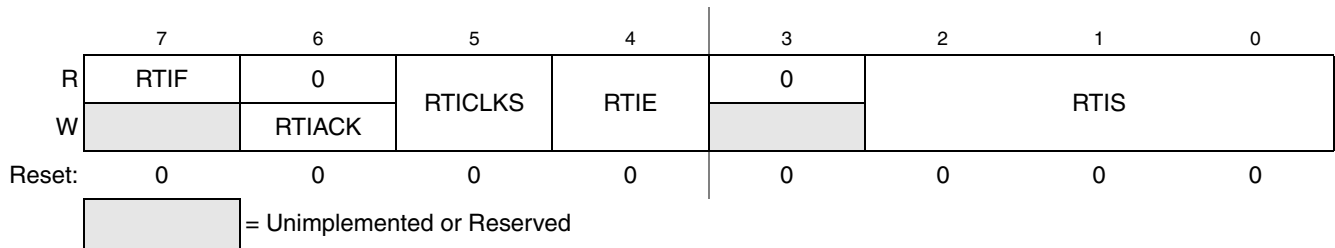


Figure 5-9. System RTI Status and Control Register (SRTISC)

Table 5-10. SRTISC Register Field Descriptions

Field	Description
7 RTIF	<b>Real-Time Interrupt Flag</b> — This read-only status bit indicates the periodic wakeup timer has timed out. 0 Periodic wakeup timer not timed out. 1 Periodic wakeup timer timed out.
6 RTIACK	<b>Real-Time Interrupt Acknowledge</b> — This write-only bit is used to acknowledge real-time interrupt request (write 1 to clear RTIF). Writing 0 has no meaning or effect. Reads always return 0.
5 RTICLKs	<b>Real-Time Interrupt Clock Select</b> — This read/write bit selects the clock source for the real-time interrupt. 0 Real-time interrupt request clock source is internal 1 kHz oscillator. 1 Real-time interrupt request clock source is 32 kHz ICS clock.
4 RTIE	<b>Real-Time Interrupt Enable</b> — This read-write bit enables real-time interrupts. 0 Real-time interrupts disabled. 1 Real-time interrupts enabled.
2:0 RTIS	<b>Real-Time Interrupt Delay Selects</b> — These read/write bits select the period for the RTI. See <a href="#">Table 5-11</a>

Table 5-12. SPMSC1 Register Field Descriptions (continued)

Field	Description
5 LVDIE	<b>Low-Voltage Detect Interrupt Enable</b> — This bit enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVDF = 1.
4 LVDRE	<b>Low-Voltage Detect Reset Enable</b> — This write-once bit enables LVDF events to generate a hardware reset (provided LVDE = 1). 0 LVDF does not generate hardware resets. 1 Force an MCU reset when LVDF = 1.
3 LVDSE	<b>Low-Voltage Detect Stop Enable</b> — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	<b>Low-Voltage Detect Enable</b> — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	<b>Bandgap Buffer Enable</b> — This bit enables an internal buffer for the bandgap voltage reference for use by the ADC module on one of its internal channels. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

### 5.8.9 System Power Management Status and Control 2 Register (SPMSC2)

This high-page register contains status and control bits to configure the stop mode behavior of the MCU. See [Section 3.6, “Stop Modes,”](#) for more information on stop modes.

	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV	LVWV	PPDF	0		PPDC <sup>1</sup>
W		LVWACK				PPDACK		
POR:	0 <sup>2</sup>	0	0	0	0	0	0	0
LVDR:	0 <sup>2</sup>	0	U	U	0	0	0	0
Other Reset	0 <sup>2</sup>	0	U	U	0	0	0	0

= Unimplemented or Reserved
 U = Unaffected by reset

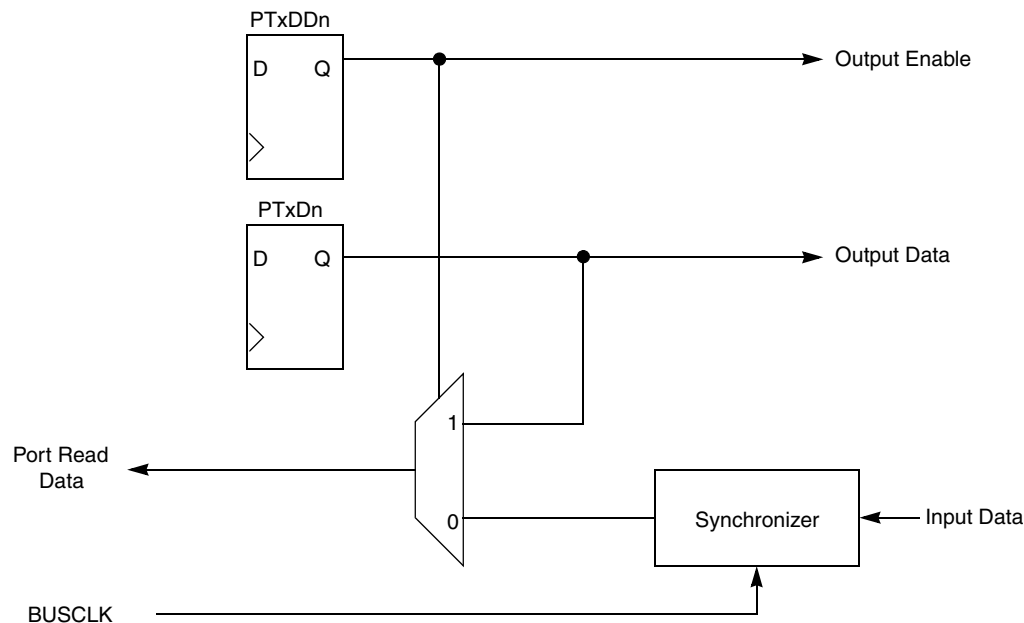
<sup>1</sup> This bit can be written only one time after reset. Additional writes are ignored.

<sup>2</sup> LVWF will be set in the case when  $V_{\text{supply}}$  transitions below the trip point or after reset and  $V_{\text{supply}}$  is already below  $V_{\text{LVW}}$ .

Figure 5-11. System Power Management Status and Control 2 Register (SPMSC2)

**Table 5-13. SPMSC2 Register Field Descriptions**

Field	Description
7 LVWF	<b>Low-Voltage Warning Flag</b> — The LVWF bit indicates the low voltage warning status. 0 Low voltage warning <b>not</b> preset. 1 Low voltage warning is present or was present.
6 LVWACK	<b>Low-Voltage Warning Acknowledge</b> — The LVWACK bit is the low-voltage warning acknowledge. Writing a 1 to LVWACK clears LVWF to a 0 if a low voltage warning is not present.
5 LVDV	<b>Low-Voltage Detect Voltage Select</b> — The LVDV bit selects the LVD trip point voltage ( $V_{LVD}$ ). 0 Low trip point selected ( $V_{LVD} = V_{LVDL}$ ). 1 High trip point selected ( $V_{LVD} = V_{LV DH}$ ).
4 LVWV	<b>Low-Voltage Warning Voltage Select</b> — The LVWV bit selects the LVW trip point voltage ( $V_{LVW}$ ). 0 Low trip point selected ( $V_{LVW} = V_{LVWL}$ ). 1 High trip point selected ( $V_{LVW} = V_{LVWH}$ ).
3 PPDF	<b>Partial Power Down Flag</b> — The PPDF bit indicates that the MCU has exited the stop2 mode. 0 Not stop2 mode recovery. 1 Stop2 mode recovery.
2 PPDACK	<b>Partial Power Down Acknowledge</b> — Writing a 1 to PPDACK clears the PPDF bit.
0 PPDC	<b>Partial Power Down Control</b> — The write-once PPDC bit controls whether stop2 or stop3 mode is selected. 0 Stop3 mode enabled. 1 Stop2, partial power down, mode enabled.



**Figure 6-1. Parallel I/O Block Diagram**

The data direction control bit (PTxDDn) determines whether the output buffer for the associated pin is enabled, and also controls the source for port data register reads. The input buffer for the associated pin is always enabled unless the pin is enabled as an analog function or is an output-only pin.

When a shared digital function is enabled for a pin, the output buffer is controlled by the shared function. However, the data direction register bit will continue to control the source for reads of the port data register.

When a shared analog function is enabled for a pin, both the input and output buffers are disabled. A value of 0 is read for any port data bit where the bit is an input (PTxDDn = 0) and the input buffer is disabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both the digital and analog functions are enabled, the analog function controls the pin.

It is a good programming practice to write to the port data register before changing the direction of a port pin to become an output. This ensures that the pin will not be driven momentarily with an old data value that happened to be in the port data register.

## 6.2 Pin Control — Pullup, Slew Rate and Drive Strength

Associated with the parallel I/O ports is a set of registers located in the high-page register space that operate independently of the parallel I/O registers. These registers are used to control pullups, slew rate and drive strength for the pins.

## 6.3 Pin Behavior in Stop Modes

Pin behavior following execution of a STOP instruction depends on the stop mode that is entered. An explanation of pin behavior for the various stop modes follows:



- In stop1 mode, all internal registers including parallel I/O control and data registers are powered off. Each of the pins assumes its default reset state (output buffer and internal pullup disabled). Upon exit from stop1, all pins must be re-configured the same as if the MCU had been reset.
- Stop2 mode is a partial power-down mode, whereby latches maintain the pin state as before the STOP instruction was executed. CPU register status and the state of I/O registers must be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user must examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, I/O data previously stored in RAM, before the STOP instruction was executed, peripherals previously enabled will require being initialized and restored to their pre-stop condition. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access of pins is now permitted again in the user's application program.
- In stop3 mode, all pin states are maintained because internal logic stays powered up. Upon recovery, all pin functions are the same as before entering stop3.

## 6.4 Parallel I/O Registers

### 6.4.1 Port A Registers

This section provides information about the registers associated with the parallel I/O ports.

Refer to tables in [Chapter 4, “Memory Map and Register Definition,”](#) for the absolute address assignments for all parallel I/O. This section refers to registers and control bits only by their names. A Freescale Semiconductor-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

#### 6.4.1.1 Port A Data (PTAD)

	7	6	5	4	3	2	1	0
R	0	0	PTAD5 <sup>1</sup>	PTAD4 <sup>2</sup>	PTAD3	PTAD2	PTAD1	PTAD0
W								
Reset:	0	0	0	0	0	0	0	0

<sup>1</sup> Reads of bit PTAD5 always return the pin value of PTA5, regardless of the value stored in bit PTADD5.

<sup>2</sup> Reads of bit PTAD4 always return the contents of PTAD4, regardless of the value stored in bit PTADD4.

**Figure 6-2. Port A Data Register (PTAD)**

## Chapter 7

# Central Processor Unit (S08CPUV2)

### 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

### 7.4.5 BGND Instruction

The BGND instruction is new to the HCS08 compared to the M68HC08. BGND would not be used in normal user programs because it forces the CPU to stop processing user instructions and enter the active background mode. The only way to resume execution of the user program is through reset or by a host debug system issuing a GO, TRACE1, or TAGGO serial command through the background debug interface.

Software-based breakpoints can be set by replacing an opcode at the desired breakpoint address with the BGND opcode. When the program reaches this breakpoint address, the CPU is forced to active background mode rather than continuing the user program.

Table 7-2. Instruction Set Summary (Sheet 3 of 9)

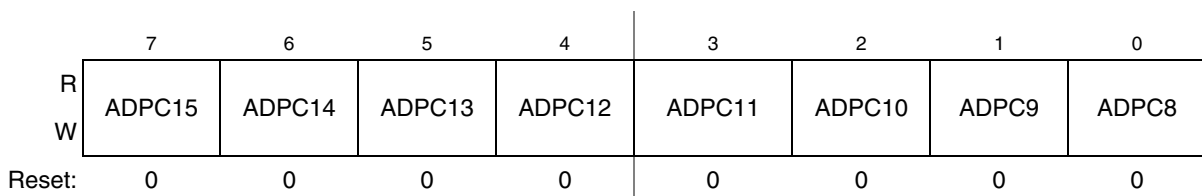
Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR				
						VH	I	N	Z	C
BRA <i>rel</i>	Branch Always (if I = 1)	REL	20 rr	3	ppp	--	--	--	--	--
BRCLR <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Clear (if (Mn) = 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 dd rr 03 dd rr 05 dd rr 07 dd rr 09 dd rr 0B dd rr 0D dd rr 0F dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	--	--	--	--	↑
BRN <i>rel</i>	Branch Never (if I = 0)	REL	21 rr	3	ppp	--	--	--	--	--
BRSET <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Set (if (Mn) = 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 dd rr 02 dd rr 04 dd rr 06 dd rr 08 dd rr 0A dd rr 0C dd rr 0E dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	--	--	--	--	↑
BSET <i>n,opr8a</i>	Set Bit <i>n</i> in Memory (Mn ← 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 dd 12 dd 14 dd 16 dd 18 dd 1A dd 1C dd 1E dd	5 5 5 5 5 5 5 5	rffwpp rffwpp rffwpp rffwpp rffwpp rffwpp rffwpp rffwpp	--	--	--	--	--
BSR <i>rel</i>	Branch to Subroutine PC ← (PC) + \$0002 push (PCL); SP ← (SP) – \$0001 push (PCH); SP ← (SP) – \$0001 PC ← (PC) + <i>rel</i>	REL	AD rr	5	ssppp	--	--	--	--	--
CBEQ <i>opr8a,rel</i> CBEQA # <i>opr8i,rel</i> CBEQX # <i>opr8i,rel</i> CBEQ <i>opr8,X+,rel</i> CBEQ <i>,X+,rel</i> CBEQ <i>opr8,SP,rel</i>	Compare and... Branch if (A) = (M) Branch if (A) = (M) Branch if (X) = (M) Branch if (A) = (M) Branch if (A) = (M) Branch if (A) = (M)	DIR IMM IMM IX1+ IX+ SP1	31 dd rr 41 ii rr 51 ii rr 61 ff rr 71 rr 9E 61 ff rr	5 4 4 5 5 6	rpppp pppp pppp rpppp rffppp prpppp	--	--	--	--	--
CLC	Clear Carry Bit (C ← 0)	INH	98	1	p	--	--	--	--	0
CLI	Clear Interrupt Mask Bit (I ← 0)	INH	9A	1	p	--	0	--	--	--
CLR <i>opr8a</i> CLRA CLR X CLR H CLR <i>opr8,X</i> CLR <i>,X</i> CLR <i>opr8,SP</i>	Clear M ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	DIR INH INH INH IX1 IX SP1	3F dd 4F 5F 8C 6F ff 7F 9E 6F ff	5 1 1 1 5 4 6	rffwpp p p p rffwpp rffwpp prffwpp	0	--	--	0	1

**Table 8-9. APCTL1 Register Field Descriptions (continued)**

Field	Description
1 ADPC1	<b>ADC Pin Control 1</b> — ADPC1 is used to control the pin associated with channel AD1. 0 AD1 pin I/O control enabled 1 AD1 pin I/O control disabled
0 ADPC0	<b>ADC Pin Control 0</b> — ADPC0 is used to control the pin associated with channel AD0. 0 AD0 pin I/O control enabled 1 AD0 pin I/O control disabled

### 8.3.9 Pin Control 2 Register (APCTL2)

APCTL2 is used to control channels 8–15 of the ADC module.


**Figure 8-12. Pin Control 2 Register (APCTL2)**
**Table 8-10. APCTL2 Register Field Descriptions**

Field	Description
7 ADPC15	<b>ADC Pin Control 15</b> — ADPC15 is used to control the pin associated with channel AD15. 0 AD15 pin I/O control enabled 1 AD15 pin I/O control disabled
6 ADPC14	<b>ADC Pin Control 14</b> — ADPC14 is used to control the pin associated with channel AD14. 0 AD14 pin I/O control enabled 1 AD14 pin I/O control disabled
5 ADPC13	<b>ADC Pin Control 13</b> — ADPC13 is used to control the pin associated with channel AD13. 0 AD13 pin I/O control enabled 1 AD13 pin I/O control disabled
4 ADPC12	<b>ADC Pin Control 12</b> — ADPC12 is used to control the pin associated with channel AD12. 0 AD12 pin I/O control enabled 1 AD12 pin I/O control disabled
3 ADPC11	<b>ADC Pin Control 11</b> — ADPC11 is used to control the pin associated with channel AD11. 0 AD11 pin I/O control enabled 1 AD11 pin I/O control disabled
2 ADPC10	<b>ADC Pin Control 10</b> — ADPC10 is used to control the pin associated with channel AD10. 0 AD10 pin I/O control enabled 1 AD10 pin I/O control disabled



### 9.1.3.5 FLL Bypassed External (FBE)

In FLL bypassed external mode, the FLL is enabled and controlled by an external reference clock, but is bypassed. The ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is supplied from the FLL.

### 9.1.3.6 FLL Bypassed External Low Power (FBELP)

In FLL bypassed external low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is not available.

### 9.1.3.7 Stop (STOP)

In stop mode the FLL is disabled and the internal or external reference clocks can be selected to be enabled or disabled. The BDC clock is not available and the ICS does not provide an MCU clock source.

## 9.1.4 Block Diagram

Figure 9-2 is the ICS block diagram.

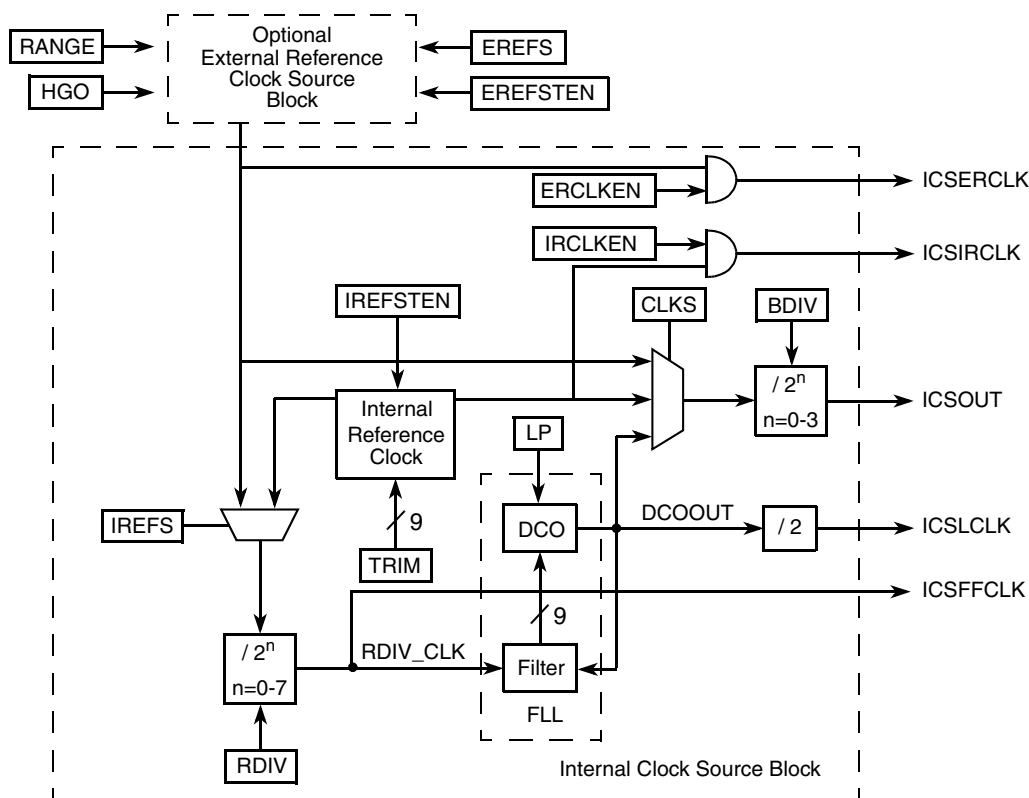
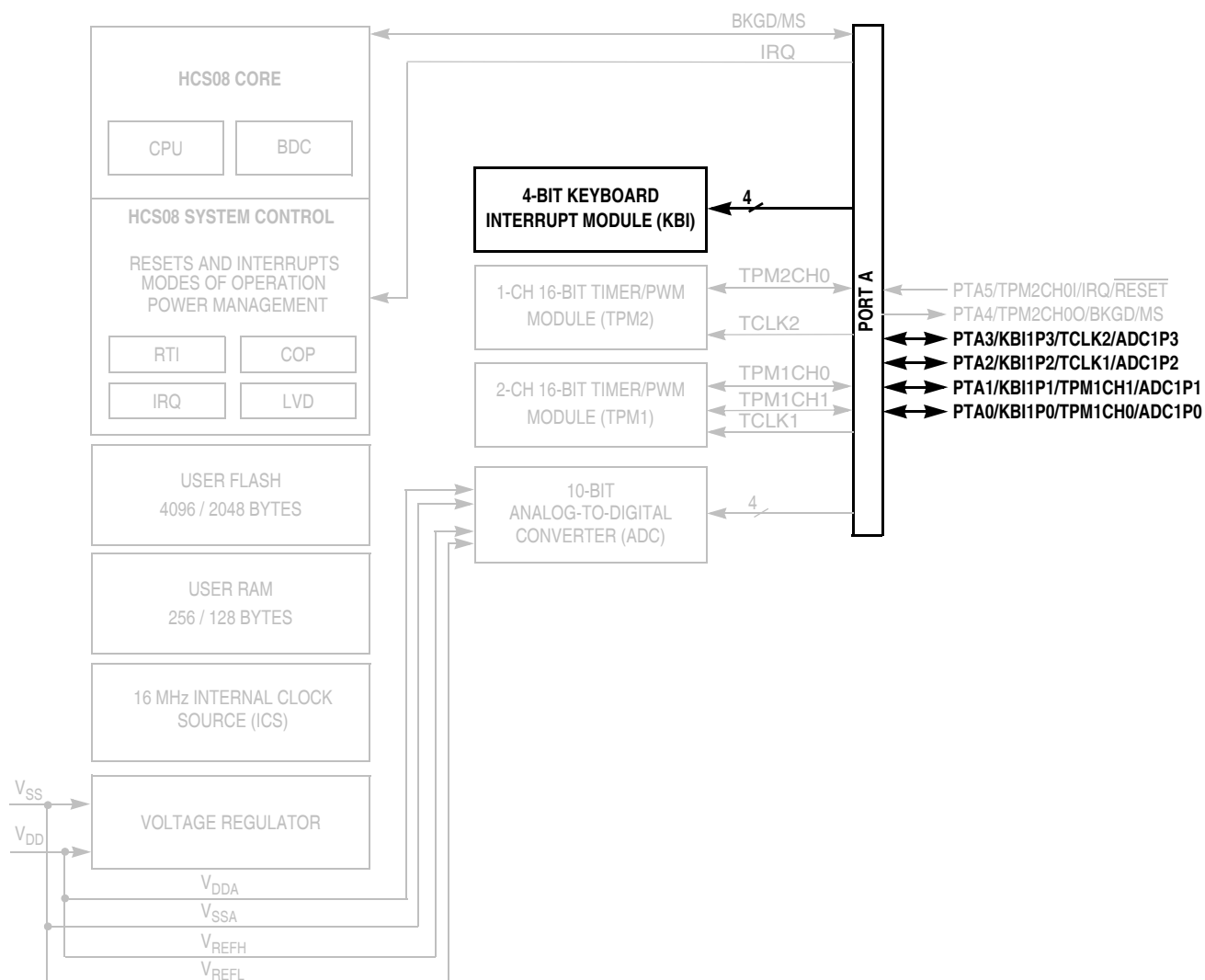


Figure 9-2. Internal Clock Source (ICS) Block Diagram



## NOTES:

- <sup>1</sup> Port pins are software configurable with pullup device if input port.
- <sup>2</sup> Port pins are software configurable for output drive strength.
- <sup>3</sup> Port pins are software configurable for output slew rate control.
- <sup>4</sup> IRQ contains a software configurable (IRQPDD) pullup/pulldown device if PTA5 enabled as IRQ pin function (IRQPE = 1).
- <sup>5</sup> RESET contains integrated pullup device if PTA5 enabled as reset pin function (RSTPE = 1).
- <sup>6</sup> PTA5 does not contain a clamp diode to  $V_{DD}$  and must not be driven above  $V_{DD}$ . The voltage measured on this pin when internal pullup is enabled may be as low as  $V_{DD} - 0.7$  V. The internal gates connected to this pin are pulled to  $V_{DD}$ .
- <sup>7</sup> PTA4 contains integrated pullup device if BKGD enabled (BKGDPE = 1).
- <sup>8</sup> When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

**Figure 10-1. MC9S08QD4 Series Block Diagram Highlighting KBI Block and Pins**



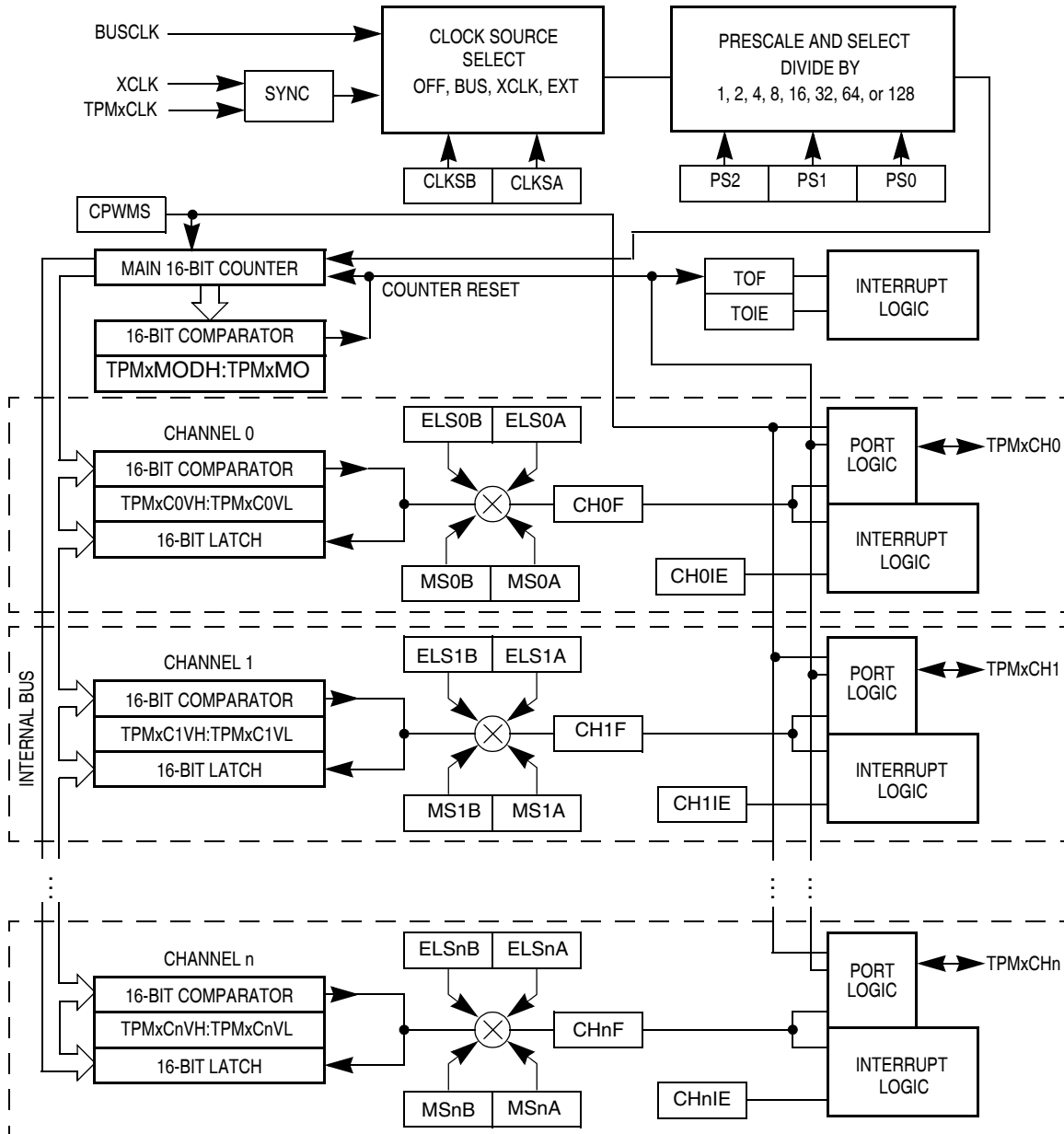


Figure 11-2. TPM Block Diagram

The central component of the TPM is the 16-bit counter that can operate as a free-running counter, a modulo counter, or an up-/down-counter when the TPM is configured for center-aligned PWM. The TPM counter (when operating in normal up-counting mode) provides the timing reference for the input capture, output compare, and edge-aligned PWM functions. The timer counter modulo registers, TPMxMODH:TPMxMODL, control the modulo value of the counter. (The values 0x0000 or 0xFFFF effectively make the counter free running.) Software can read the counter value at any time without affecting the counting sequence. Any write to either byte of the TPMxCNT counter resets the counter regardless of the data value written.

### 11.4.3 Center-Aligned PWM Mode

This type of PWM output uses the up-/down-counting mode of the timer counter (CPWMS = 1). The output compare value in TPMxCnVH:TPMxCnVL determines the pulse width (duty cycle) of the PWM signal and the period is determined by the value in TPMxMODH:TPMxMODL.

TPMxMODH:TPMxMODL must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results. ELSnA will determine the polarity of the CPWM output.

$$\text{pulse width} = 2 \times (\text{TPMxCnVH:TPMxCnVL}) \quad \text{Eqn. 11-1}$$

$$\begin{aligned} \text{period} &= 2 \times (\text{TPMxMODH:TPMxMODL}); \\ &\text{for TPMxMODH:TPMxMODL} = 0x0001\text{--}0x7FFF \end{aligned} \quad \text{Eqn. 11-2}$$

If the channel value register TPMxCnVH:TPMxCnVL is zero or negative (bit 15 set), the duty cycle will be 0%. If TPMxCnVH:TPMxCnVL is a positive value (bit 15 clear) and is greater than the (nonzero) modulus setting, the duty cycle will be 100% because the duty cycle compare will never occur. This implies the usable range of periods set by the modulus register is 0x0001 through 0x7FFE (0x7FFF if generation of 100% duty cycle is not necessary). This is not a significant limitation because the resulting period is much longer than required for normal applications.

TPMxMODH:TPMxMODL = 0x0000 is a special case that must not be used with center-aligned PWM mode. When CPWMS = 0, this case corresponds to the counter running free from 0x0000 through 0xFFFF, but when CPWMS = 1 the counter needs a valid match to the modulus register somewhere other than at 0x0000 in order to change directions from up-counting to down-counting.

Figure 11-12 shows the output compare value in the TPM channel registers (multiplied by 2), which determines the pulse width (duty cycle) of the CPWM signal. If ELSnA = 0, the compare match while counting up forces the CPWM output signal low and a compare match while counting down forces the output high. The counter counts up until it reaches the modulo setting in TPMxMODH:TPMxMODL, then counts down until it reaches zero. This sets the period equal to two times TPMxMODH:TPMxMODL.

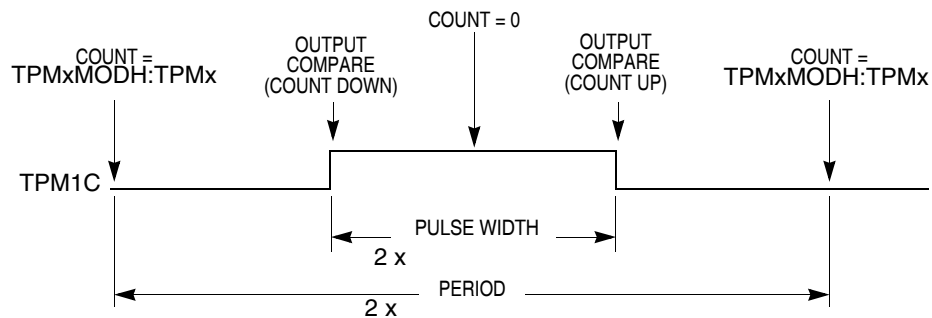


Figure 11-12. CPWM Period and Pulse Width (ELSnA = 0)

Center-aligned PWM outputs typically produce less noise than edge-aligned PWMs because fewer I/O pin transitions are lined up at the same system clock edge. This type of PWM is also required for some types of motor drives.

Because the HCS08 is a family of 8-bit MCUs, the settings in the timer channel registers are buffered to ensure coherent 16-bit updates and to avoid unexpected PWM pulse widths. Writes to any of the registers, TPMxMODH, TPMxMODL, TPMxCnVH, and TPMxCnVL, actually write to buffer registers. Values are

transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the timer counter overflows (reverses direction from up-counting to down-counting at the end of the terminal count in the modulus register). This TPMxCNT overflow requirement only applies to PWM channels, not output compares.

Optionally, when  $\text{TPMxCNTH}:\text{TPMxCNTL} = \text{TPMxMODH}:\text{TPMxMODL}$ , the TPM can generate a TOF interrupt at the end of this count. The user can choose to reload any number of the PWM buffers, and they will all update simultaneously at the start of a new period.

Writing to TPMxSC cancels any values written to TPMxMODH and/or TPMxMODL and resets the coherency mechanism for the modulo registers. Writing to TPMxCnSC cancels any values written to the channel value registers and resets the coherency mechanism for TPMxCnVH:TPMxCnVL.

## 11.5 TPM Interrupts

The TPM generates an optional interrupt for the main counter overflow and an interrupt for each channel. The meaning of channel interrupts depends on the mode of operation for each channel. If the channel is configured for input capture, the interrupt flag is set each time the selected input capture edge is recognized. If the channel is configured for output compare or PWM modes, the interrupt flag is set each time the main timer counter matches the value in the 16-bit channel value register. See the [Resets, Interrupts, and System Configuration](#) chapter for absolute interrupt vector addresses, priority, and local interrupt mask control bits.

For each interrupt source in the TPM, a flag bit is set on recognition of the interrupt condition such as timer overflow, channel input capture, or output compare events. This flag may be read (polled) by software to verify that the action has occurred, or an associated enable bit (TOIE or CHnIE) can be set to enable hardware interrupt generation. While the interrupt enable bit is set, a static interrupt will be generated whenever the associated interrupt flag equals 1. It is the responsibility of user software to perform a sequence of steps to clear the interrupt flag before returning from the interrupt service routine.

### 11.5.1 Clearing Timer Interrupt Flags

TPM interrupt flags are cleared by a 2-step process that includes a read of the flag bit while it is set (1) followed by a write of 0 to the bit. If a new event is detected between these two steps, the sequence is reset and the interrupt flag remains set after the second step to avoid the possibility of missing the new event.

### 11.5.2 Timer Overflow Interrupt Description

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the 16-bit timer counter counts from 0x0000 through 0xFFFF and overflows to 0x0000 on the next counting clock. TOF becomes set at the transition from 0xFFFF to 0x0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to 0x0000. When the counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

## Chapter 12

# Development Support

### 12.1 Introduction

Development support systems in the HCS08 include the background debug controller (BDC). The BDC provides a single-wire debug interface to the target MCU that provides a convenient interface for programming the on-chip flash and other nonvolatile memories. The BDC is also the primary debug interface for development and allows non-intrusive access to memory data and traditional debug features such as CPU register modify, breakpoints, and single instruction trace commands.

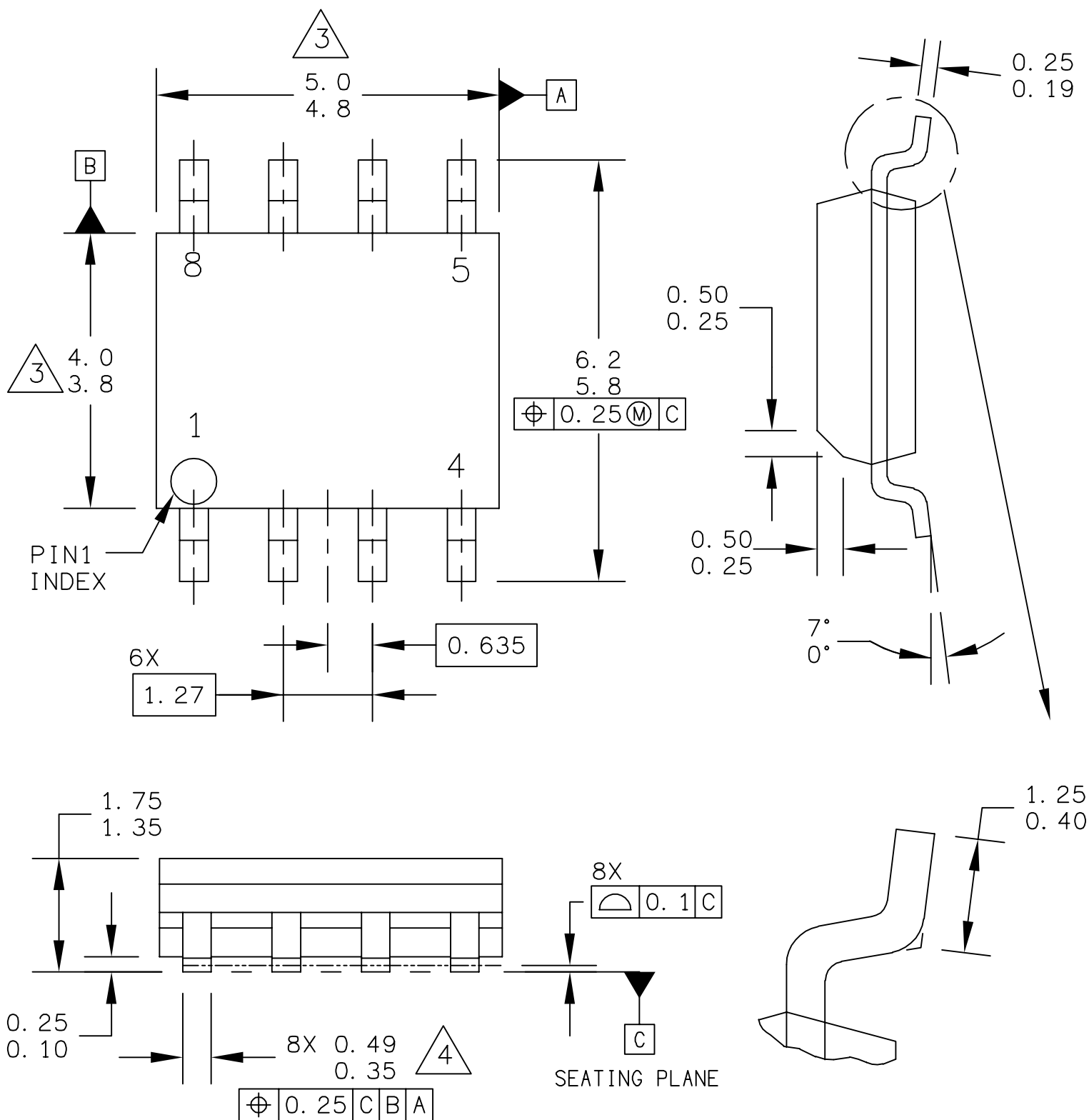
In the HCS08 Family, address and data bus signals are not available on external pins (not even in test modes). Debug is done through commands fed into the target MCU via the single-wire background debug interface. The debug module provides a means to selectively trigger and capture bus information so an external development system can reconstruct what happened inside the MCU on a cycle-by-cycle basis without having external access to the address and data signals.

#### 12.1.1 Forcing Active Background

The method for forcing active background mode depends on the specific HCS08 derivative. For the MC9S08QD4 series, you can force active background mode by holding the BKGD pin low as the MCU exits the reset condition independent of what caused the reset. If no debug pod is connected to the BKGD pin, the MCU will always reset into normal operating mode.

#### 12.1.2 Module Configuration

The alternative BDC clock source for MC9S08QD4 series is the ICGCLK. See [Chapter 9, “Internal Clock Source \(S08ICSV1\)”](#), for more information about ICGCLK and how to select clock sources.



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	<b>MECHANICAL OUTLINE</b>	PRINT VERSION NOT TO SCALE	
TITLE:  8LD SOIC NARROW BODY	DOCUMENT NO: 98ASB42564B		REV: U
	CASE NUMBER: 751-07		07 APR 2005
	STANDARD: JEDEC MS-012AA		