



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08aw32cfger">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08aw32cfger</a>

Section Number	Title	Page
4.4	FLASH .....	50
4.4.1	Features .....	51
4.4.2	Program and Erase Times .....	51
4.4.3	Program and Erase Command Execution .....	52
4.4.4	Burst Program Execution .....	53
4.4.5	Access Errors .....	55
4.4.6	FLASH Block Protection .....	55
4.4.7	Vector Redirection .....	56
4.5	Security .....	56
4.6	FLASH Registers and Control Bits .....	58
4.6.1	FLASH Clock Divider Register (FCDIV) .....	58
4.6.2	FLASH Options Register (FOPT and NVOPT) .....	59
4.6.3	FLASH Configuration Register (FCNFG) .....	60
4.6.4	FLASH Protection Register (FPROT and NVPROT) .....	61
4.6.5	FLASH Status Register (FSTAT) .....	61
4.6.6	FLASH Command Register (FCMD) .....	63

## Chapter 5

### Resets, Interrupts, and System Configuration

5.1	Introduction .....	65
5.2	Features .....	65
5.3	MCU Reset .....	65
5.4	Computer Operating Properly (COP) Watchdog .....	66
5.5	Interrupts .....	66
5.5.1	Interrupt Stack Frame .....	67
5.5.2	External Interrupt Request (IRQ) Pin .....	68
5.5.3	Interrupt Vectors, Sources, and Local Masks .....	69
5.6	Low-Voltage Detect (LVD) System .....	71
5.6.1	Power-On Reset Operation .....	71
5.6.2	LVD Reset Operation .....	71
5.6.3	LVD Interrupt Operation .....	71
5.6.4	Low-Voltage Warning (LVW) .....	71
5.7	Real-Time Interrupt (RTI) .....	71
5.8	MCLK Output .....	72
5.9	Reset, Interrupt, and System Control Registers and Control Bits .....	72
5.9.1	Interrupt Pin Request Status and Control Register (IRQSC) .....	73
5.9.2	System Reset Status Register (SRS) .....	74
5.9.3	System Background Debug Force Reset Register (SBDFFR) .....	75
5.9.4	System Options Register (SOPT) .....	75
5.9.5	System MCLK Control Register (SMCLK) .....	76
5.9.6	System Device Identification Register (SDIDH, SDIDL) .....	77
5.9.7	System Real-Time Interrupt Status and Control Register (SRTISC) .....	78

Section Number	Title	Page
5.9.8	System Power Management Status and Control 1 Register (SPMSC1) .....	79
5.9.9	System Power Management Status and Control 2 Register (SPMSC2) .....	80

## Chapter 6 Parallel Input/Output

6.1	Introduction .....	81
6.2	Features .....	81
6.3	Pin Descriptions .....	82
6.3.1	Port A .....	82
6.3.2	Port B .....	82
6.3.3	Port C .....	83
6.3.4	Port D .....	83
6.3.5	Port E .....	84
6.3.6	Port F .....	85
6.3.7	Port G .....	85
6.4	Parallel I/O Control .....	86
6.5	Pin Control .....	87
6.5.1	Internal Pullup Enable .....	87
6.5.2	Output Slew Rate Control Enable .....	87
6.5.3	Output Drive Strength Select .....	87
6.6	Pin Behavior in Stop Modes .....	88
6.7	Parallel I/O and Pin Control Registers .....	88
6.7.1	Port A I/O Registers (PTAD and PTADD) .....	88
6.7.2	Port A Pin Control Registers (PTAPE, PTASE, PTADS) .....	89
6.7.3	Port B I/O Registers (PTBD and PTBDD) .....	91
6.7.4	Port B Pin Control Registers (PTBPE, PTBSE, PTBDS) .....	92
6.7.5	Port C I/O Registers (PTCD and PTCDD) .....	94
6.7.6	Port C Pin Control Registers (PTCPE, PTCSE, PTCDS) .....	95
6.7.7	Port D I/O Registers (PTDD and PTDDD) .....	97
6.7.8	Port D Pin Control Registers (PTDPE, PTDSE, PTDDS) .....	98
6.7.9	Port E I/O Registers (PTED and PTEDD) .....	100
6.7.10	Port E Pin Control Registers (PTEPE, PTESE, PTEDS) .....	101
6.7.11	Port F I/O Registers (PTFD and PTFDD) .....	103
6.7.12	Port F Pin Control Registers (PTFPE, PTFSE, PTFDS) .....	104
6.7.13	Port G I/O Registers (PTGD and PTGDD) .....	106
6.7.14	Port G Pin Control Registers (PTGPE, PTGSE, PTGDS) .....	107

## Chapter 7 Central Processor Unit (S08CPUV2)

7.1	Introduction .....	109
7.1.1	Features .....	109
7.2	Programmer's Model and CPU Registers .....	110

# Chapter 1

## Introduction

### 1.1 Overview

The MC9S08AW60, MC9S08AW48, MC9S08AW32, and MC9S08AW16 are members of the low-cost, high-performance HCS08 family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced HCS08 core and are available with a variety of modules, memory sizes, memory types, and package types. Refer to Table 1-1 for memory sizes and package types.

Table 1-2 summarizes the peripheral availability per package type for the devices available in the MC9S08AW60 Series.

**Table 1-1. Devices in the MC9S08AW60 Series**

Device	FLASH	RAM	Package
MC9S08AW60	63,280	2048	64 QFP
MC9S08AW48	49,152		64 LQFP
MC9S08AW32	32,768		48 QFN 44 LQFP
MC9S08AW16	16,384	1024	48 QFN 44 LQFP

**Table 1-2. Peripherals Available per Package Type**

Feature	Package Options		
	64-pin	48-pin	44-pin
ADC	16-channel	8-channel	8-channel
IIC	yes	yes	yes
IRQ	yes	yes	yes
KB11	8	7	6
SCI1	yes	yes	yes
SCI2	yes	yes	yes
SPI1	yes	yes	yes
TPM1	6-channel	4-channel	4-channel
TPM1CLK	yes	no	no
TPM2	2-channel	2-channel	2-channel
TPM2CLK	yes	no	no
I/O pins	54	38	34

### 1.2 MCU Block Diagrams

The block diagram shows the structure of the MC9S08AW60 Series.

- The output of the digitally-controlled oscillator (DCO) in the frequency-locked loop sub-module
- Control bits inside the ICG determine which source is connected.
- FFE is a control signal generated inside the ICG. If the frequency of ICGOUT  $> 4 \times$  the frequency of ICGERCLK, this signal is a logic 1 and the fixed-frequency clock will be ICGERCLK/2. Otherwise the fixed-frequency clock will be BUSCLK.
- ICGLCLK — Development tools can select this internal self-clocked source ( $\sim 8$  MHz) to speed up BDC communications in systems where the bus clock is slow.
- ICGERCLK — External reference clock can be selected as the real-time interrupt clock source. Can also be used as the ALTCLK input to the ADC module.

After entering active background mode, the CPU is held in a suspended state waiting for serial background commands rather than executing instructions from the user's application program.

Background commands are of two types:

- Non-intrusive commands, defined as commands that can be issued while the user program is running. Non-intrusive commands can be issued through the BKGD pin while the MCU is in run mode; non-intrusive commands can also be executed when the MCU is in the active background mode. Non-intrusive commands include:
  - Memory access commands
  - Memory-access-with-status commands
  - BDC register access commands
  - The BACKGROUND command
- Active background commands, which can only be executed while the MCU is in active background mode. Active background commands include commands to:
  - Read or write CPU registers
  - Trace one user program instruction at a time
  - Leave active background mode to return to the user's application program (GO)

The active background mode is used to program a bootloader or user application program into the FLASH program memory before the MCU is operated in run mode for the first time. When the MC9S08AW60 Series is shipped from the Freescale Semiconductor factory, the FLASH program memory is erased by default unless specifically noted so there is no program that could be executed in run mode until the FLASH memory is initially programmed. The active background mode can also be used to erase and reprogram the FLASH memory after it has been previously programmed.

For additional information about the active background mode, refer to Chapter 15, "Development Support."

### 3.5 Wait Mode

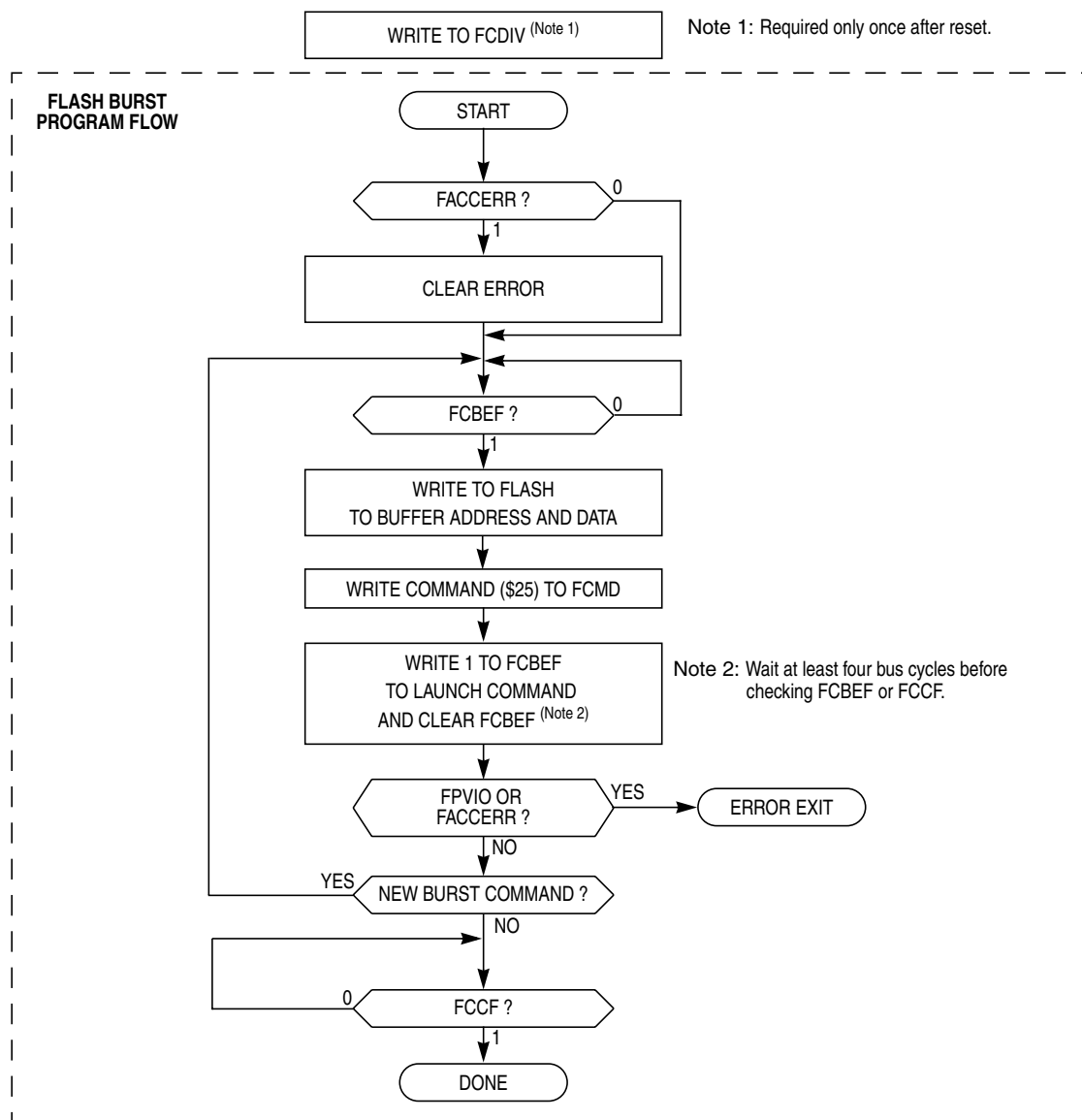
Wait mode is entered by executing a WAIT instruction. Upon execution of the WAIT instruction, the CPU enters a low-power state in which it is not clocked. The I bit in CCR is cleared when the CPU enters the wait mode, enabling interrupts. When an interrupt request occurs, the CPU exits the wait mode and resumes processing, beginning with the stacking operations leading to the interrupt service routine.

While the MCU is in wait mode, there are some restrictions on which background debug commands can be used. Only the BACKGROUND command and memory-access-with-status commands are available when the MCU is in wait mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from wait mode and enter active background mode.

### 3.6 Stop Modes

One of two stop modes is entered upon execution of a STOP instruction when the STOPE bit in the system option register is set. In both stop modes, all internal clocks are halted. If the STOPE bit is not set when

The first byte of a series of sequential bytes being programmed in burst mode will take the same amount of time to program as a byte programmed in standard mode. Subsequent bytes will program in the burst program time provided that the conditions above are met. In the case the next sequential address is the beginning of a new row, the program time for that byte will be the standard time instead of the burst time. This is because the high voltage to the array must be disabled and then enabled again. If a new burst command has not been queued before the current command completes, then the charge pump will be disabled and high voltage removed from the array.



**Figure 4-4. FLASH Burst Program Flowchart**

### 4.6.6 FLASH Command Register (FCMD)

Only five command codes are recognized in normal user modes as shown in Table 4-14. Refer to Section 4.4.3, “Program and Erase Command Execution” for a detailed discussion of FLASH programming and erase operations.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
Reset	0	0	0	0	0	0	0	0

Figure 4-11. FLASH Command Register (FCMD)

Table 4-13. FCMD Register Field Descriptions

Field	Description
FCMD[7:0]	<b>FLASH Command Bits</b> — See Table 4-14

Table 4-14. FLASH Commands

Command	FCMD	Equate File Label
Blank check	\$05	mBlank
Byte program	\$20	mByteProg
Byte program — burst mode	\$25	mBurstProg
Page erase (512 bytes/page)	\$40	mPageErase
Mass erase (all FLASH)	\$41	mMassErase

All other command codes are illegal and generate an access error.

It is not necessary to perform a blank check command after a mass erase operation. Only blank check is required as part of the security unlocking mechanism.



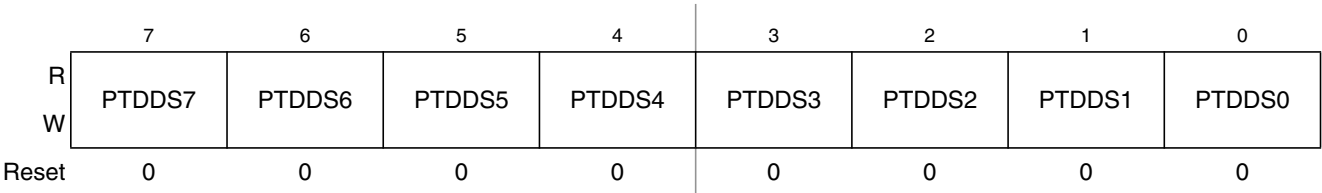


Figure 6-28. Output Drive Strength Selection for Port D (PTDDS)

Table 6-21. PTDDS Register Field Descriptions

Field	Description
7:0 PTDDS[7:0]	<b>Output Drive Strength Selection for Port D Bits</b> — Each of these control bits selects between low and high output drive for the associated PTD pin. 0 Low output drive enabled for port D bit n. 1 High output drive enabled for port D bit n.

Table 7-3. Opcode Map (Sheet 2 of 2)

Bit-Manipulation	Branch	Read-Modify-Write				Control				Register/Memory							
						9E60 6 3 SP1 NEG						9ED0 5 4 SP2 SUB	9EE0 4 3 SP1 SUB				
						9E61 6 4 SP1 CBEQ						9ED1 5 4 SP2 CMP	9EE1 4 3 SP1 CMP				
												9ED2 5 4 SP2 SBC	9EE2 4 3 SP1 SBC				
						9E63 6 3 SP1 COM						9ED3 5 4 SP2 CPX	9EE3 4 3 SP1 CPX	9EF3 6 3 SP1 CPHX			
						9E64 6 3 SP1 LSR						9ED4 5 4 SP2 AND	9EE4 4 3 SP1 AND				
												9ED5 5 4 SP2 BIT	9EE5 4 3 SP1 BIT				
						9E66 6 3 SP1 ROR						9ED6 5 4 SP2 LDA	9EE6 4 3 SP1 LDA				
						9E67 6 3 SP1 ASR						9ED7 5 4 SP2 STA	9EE7 4 3 SP1 STA				
						9E68 6 3 SP1 LSL						9ED8 5 4 SP2 EOR	9EE8 4 3 SP1 EOR				
						9E69 6 3 SP1 ROL						9ED9 5 4 SP2 ADC	9EE9 4 3 SP1 ADC				
						9E6A 6 3 SP1 DEC						9EDA 5 4 SP2 ORA	9EEA 4 3 SP1 ORA				
						9E6B 8 4 SP1 DBNZ						9EDB 5 4 SP2 ADD	9EEB 4 3 SP1 ADD				
						9E6C 6 3 SP1 INC											
						9E6D 5 3 SP1 TST											
										9EAE 5 2 IX LDHX	9EBE 6 4 IX2 LDHX	9ECE 5 3 IX1 LDHX	9EDE 5 4 SP2 LDX	9EEE 4 3 SP1 LDX	9EFE 5 3 SP1 LDHX		
						9E6F 6 3 SP1 CLR						9EDF 5 4 SP2 STX	9EEF 4 3 SP1 STX	9EFF 5 3 SP1 STHX			

INH Inherent  
 IMM Immediate  
 DIR Direct  
 EXT Extended  
 DD DIR to DIR  
 IX+D IX+ to DIR  
 REL Relative  
 IX Indexed, No Offset  
 IX1 Indexed, 8-Bit Offset  
 IX2 Indexed, 16-Bit Offset  
 IMD IMM to DIR  
 DIX+ DIR to IX+  
 SP1 Stack Pointer, 8-Bit Offset  
 SP2 Stack Pointer, 16-Bit Offset  
 IX+ Indexed, No Offset with Post Increment  
 IX1+ Indexed, 1-Byte Offset with Post Increment

Note: All Sheet 2 Opcodes are Preceded by the Page 2 Prebyte (9E)

Prebyte (9E) and Opcode in Hexadecimal  
 Number of Bytes

9E60 6 3 SP1 NEG	HCS08 Cycles Instruction Mnemonic Addressing Mode
------------------------	---

Table 8-12. MFD and RFD Decode Table

101	14	101	+32
110	16	110	+64
111	18	111	+128

## 8.5.2 Example #1: External Crystal = 32 kHz, Bus Frequency = 4.19 MHz

In this example, the FLL will be used (in FEE mode) to multiply the external 32 kHz oscillator up to 8.38 MHz to achieve 4.19 MHz bus frequency.

After the MCU is released from reset, the ICG is in self-clocked mode (SCM) and supplies approximately 8 MHz on ICGOUT, which corresponds to a 4 MHz bus frequency ( $f_{Bus}$ ).

The clock scheme will be FLL engaged, external (FEE). So

$$f_{ICGOUT} = f_{ext} * P * N / R ; P = 64, f_{ext} = 32 \text{ kHz} \quad \text{Eqn. 8-1}$$

Solving for N / R gives:

$$N / R = 8.38 \text{ MHz} / (32 \text{ kHz} * 64) = 4 ; \text{ we can choose } N = 4 \text{ and } R = 1 \quad \text{Eqn. 8-2}$$

The values needed in each register to set up the desired operation are:

**ICGC1 = \$38 (%00111000)**

Bit 7	HGO	0	Configures oscillator for low power
Bit 6	RANGE	0	Configures oscillator for low-frequency range; FLL prescale factor is 64
Bit 5	REFS	1	Oscillator using crystal or resonator is requested
Bits 4:3	CLKS	11	FLL engaged, external reference clock mode
Bit 2	OSCSTEN	0	Oscillator disabled
Bit 1	LOCD	0	Loss-of-clock detection enabled
Bit 0		0	Unimplemented or reserved, always reads zero

**ICGC2 = \$00 (%00000000)**

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bits 6:4	MFD	000	Sets the MFD multiplication factor to 4
Bit 3	LOCRES	0	Generates an interrupt request on loss of clock
Bits 2:0	RFD	000	Sets the RFD division factor to ÷1

**ICGS1 = \$xx**

This is read only except for clearing interrupt flag

**ICGS2 = \$xx**

This is read only; should read DCOS = 1 before performing any time critical tasks

**ICGFLTLU/L = \$xx**

Only needed in self-clocked mode; FLT will be adjusted by loop to give 8.38 MHz DCO clock  
 Bits 15:12 unused 0000

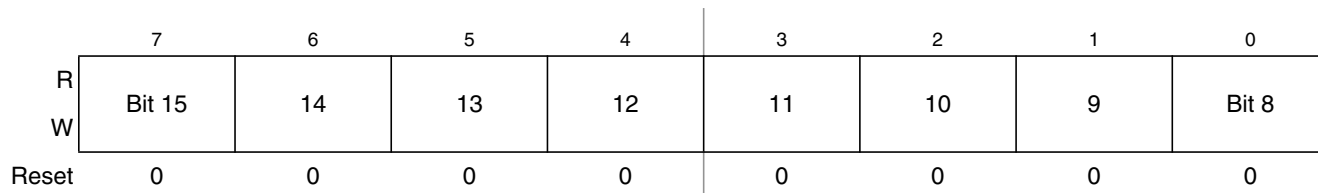
**Table 10-5. Mode, Edge, and Level Selection**

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00	Pin not used for TPM channel; use as an external clock for the TPM or revert to general-purpose I/O	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

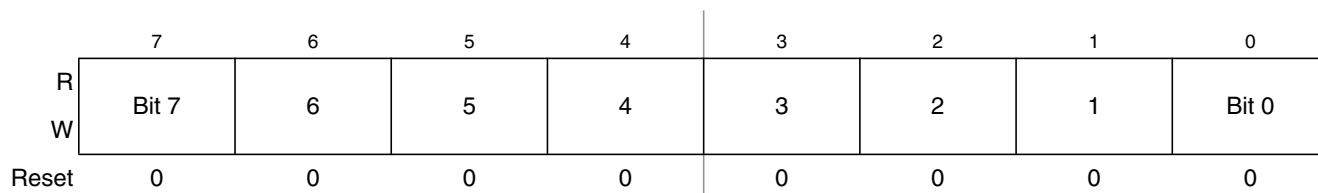
If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts or using the status flags to avoid any unexpected behavior.

### 10.4.5 Timer x Channel Value Registers (TPMxCnVH:TPMxCnVL)

These read/write registers contain the captured TPM counter value of the input capture function or the output compare value for the output compare or PWM functions. The channel value registers are cleared by reset.

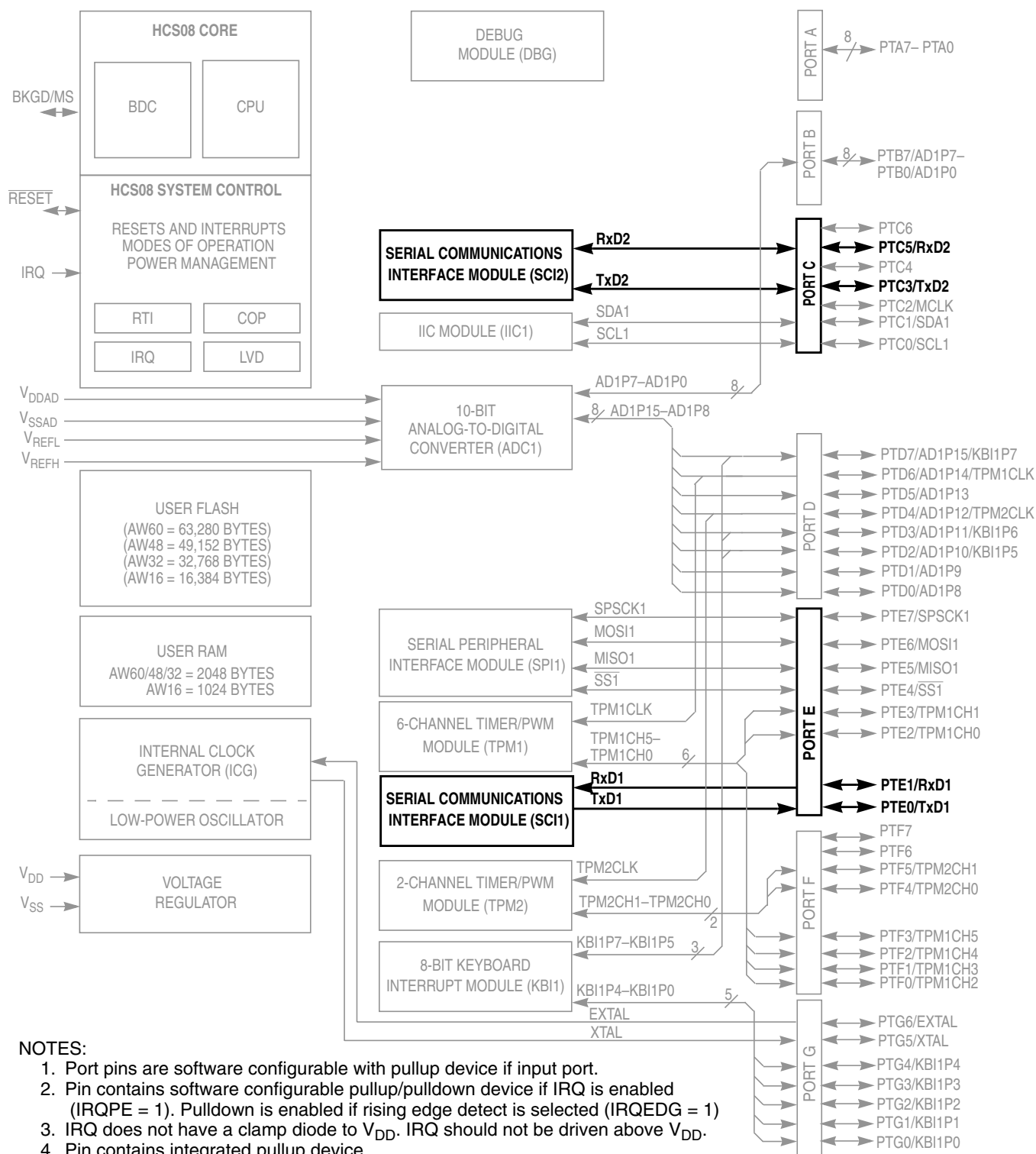


**Figure 10-9. Timer x Channel Value Register High (TPMxCnVH)**

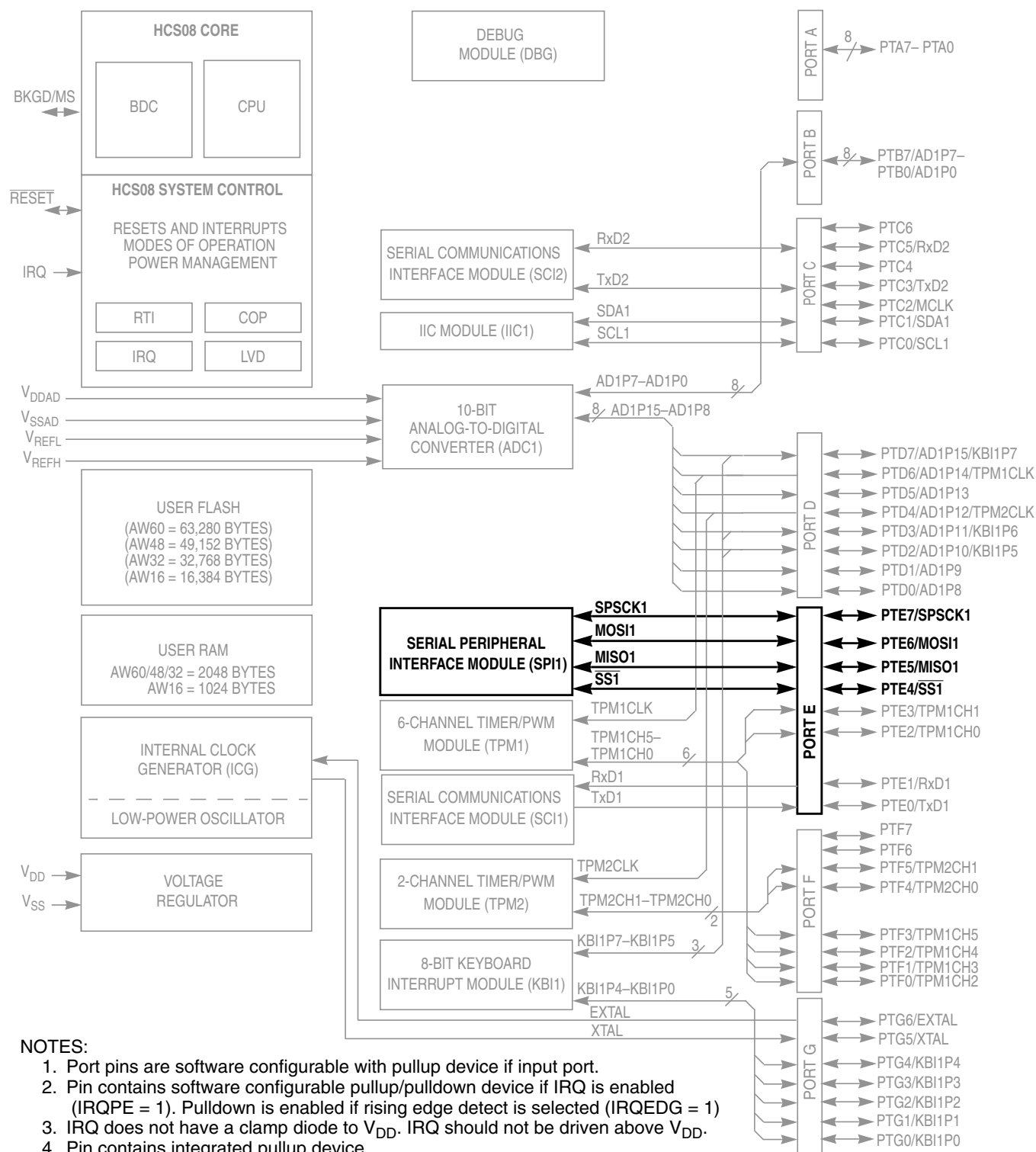


**Figure 10-10. Timer Channel Value Register Low (TPMxCnVL)**

In input capture mode, reading either byte (TPMxCnVH or TPMxCnVL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This latching mechanism also resets (becomes unlatched) when the TPMxCnSC register is written.



**Figure 11-1. Block Diagram Highlighting the SCI Modules**



**Figure 12-1. Block Diagram Highlighting the SPI Module**

**Table 12-1. SPI1C1 Field Descriptions (continued)**

Field	Description
4 MSTR	<b>Master/Slave Mode Select</b> 0 SPI module configured as a slave SPI device 1 SPI module configured as a master SPI device
3 CPOL	<b>Clock Polarity</b> — This bit effectively places an inverter in series with the clock signal from a master SPI or to a slave SPI device. Refer to Section 12.4.1, “SPI Clock Formats” for more details. 0 Active-high SPI clock (idles low) 1 Active-low SPI clock (idles high)
2 CPHA	<b>Clock Phase</b> — This bit selects one of two clock formats for different kinds of synchronous serial peripheral devices. Refer to Section 12.4.1, “SPI Clock Formats” for more details. 0 First edge on SPSCCK occurs at the middle of the first cycle of an 8-cycle data transfer 1 First edge on SPSCCK occurs at the start of the first cycle of an 8-cycle data transfer
1 SSOE	<b>Slave Select Output Enable</b> — This bit is used in combination with the mode fault enable (MODFEN) bit in SPCR2 and the master/slave (MSTR) control bit to determine the function of the $\overline{SS}$ pin as shown in Table 12-2.
0 LSBFE	<b>LSB First (Shifter Direction)</b> 0 SPI serial data transfers start with most significant bit 1 SPI serial data transfers start with least significant bit

**Table 12-2.  $\overline{SS}$  Pin Function**

MODFEN	SSOE	Master Mode	Slave Mode
0	0	General-purpose I/O (not SPI)	Slave select input
0	1	General-purpose I/O (not SPI)	Slave select input
1	0	$\overline{SS}$ input for mode fault	Slave select input
1	1	Automatic $\overline{SS}$ output	Slave select input

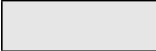
**NOTE**

Ensure that the SPI should not be disabled (SPE=0) at the same time as a bit change to the CPHA bit. These changes should be performed as separate operations or unexpected behavior may occur.

### 12.3.2 SPI Control Register 2 (SPI1C2)

This read/write register is used to control optional features of the SPI system. Bits 7, 6, 5, and 2 are not implemented and always read 0.

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

**Figure 12-6. SPI Control Register 2 (SPI1C2)**

## Chapter 13

# Inter-Integrated Circuit (S08IICV1)

### 13.1 Introduction

The MC9S08AW60 Series of microcontrollers has an inter-integrated circuit (IIC) module for communication with other integrated circuits. The two pins associated with this module, SCL and SDA, are open-drain outputs and are shared with port C pins 0 and 1, respectively.



### 13.1.1 Features

The IIC includes these distinctive features:

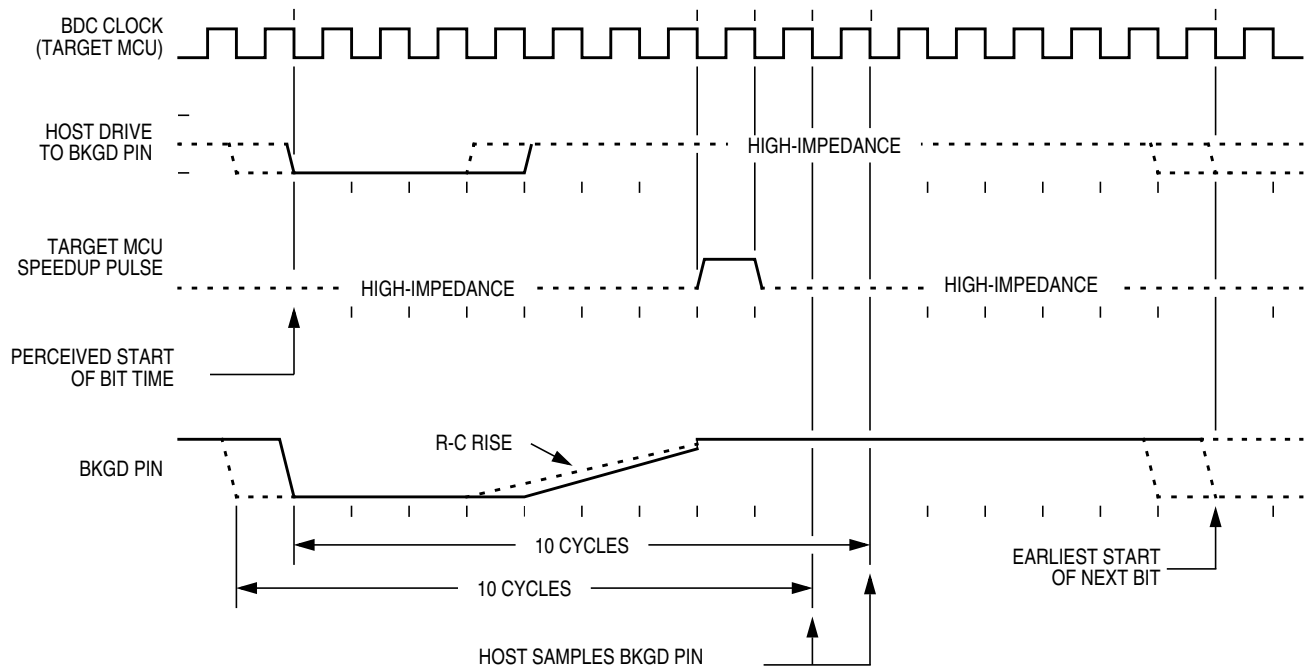
- Compatible with IIC bus standard
- Multi-master operation
- Software programmable for one of 64 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- START and STOP signal generation/detection
- Repeated START signal generation
- Acknowledge bit generation/detection
- Bus busy detection

### 13.1.2 Modes of Operation

The IIC functions the same in normal and monitor modes. A brief description of the IIC in the various MCU modes is given here.

- Run mode — This is the basic mode of operation. To conserve power in this mode, disable the module.
- Wait mode — The module will continue to operate while the MCU is in wait mode and can provide a wake-up interrupt.
- Stop mode — The IIC is inactive in stop3 mode for reduced power consumption. The STOP instruction does not affect IIC register states. Stop2 will reset the register contents.

Figure 15-3 shows the host receiving a logic 1 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target BDC cycles). The host must release the low drive before the target MCU drives a brief active-high speedup pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about 10 cycles after it started the bit time.



**Figure 15-3. BDC Target-to-Host Serial Bit Timing (Logic 1)**

## 15.3 On-Chip Debug System (DBG)

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

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

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in Section 15.3.6, "Hardware Breakpoints."

### 15.3.1 Comparators A and B

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

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

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

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

### 15.3.2 Bus Capture Information and FIFO Operation

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

**Table 15-2. BDCSCR Register Field Descriptions (continued)**

Field	Description
2 WS	<p><b>Wait or Stop Status</b> — 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.</p> <p>0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active)</p> <p>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</p>
1 WSF	<p><b>Wait or Stop Failure Status</b> — 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.)</p> <p>0 Memory access did not conflict with a wait or stop instruction</p> <p>1 Memory access command failed because the CPU entered wait or stop mode</p>
0 DVF	<p><b>Data Valid Failure Status</b> — This status bit is not used in the MC9S08AW60 Series because it does not have any slow access memory.</p> <p>0 Memory access did not conflict with a slow memory access</p> <p>1 Memory access command failed because CPU was not finished with a slow memory access</p>

### 15.4.1.2 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 15.2.4, “BDC Hardware Breakpoint.”

### 15.4.2 System Background Debug Force Reset Register (SBDFFR)

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



NOTES:

- 1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
- 2. CONTROLLING DIMENSION: MILLIMETER.
- 3. DATUM PLANE –H– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
- 4. DATUMS A–B AND –D– TO BE DETERMINED AT DATUM PLANE –H–.
- 5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE –C–.
- 6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –H–.
- 7. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF THE DIMENSION AT MAXIMUM MATERIAL CONDICTION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE:  64LD QFP (14 X 14)		DOCUMENT NO: 98ASB42844B		REV: B
		CASE NUMBER: 840B-01		20 MAY 2005
		STANDARD: NON-JEDEC		