

Welcome to [E-XFL.COM](http://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	36
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gt8acfber">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gt8acfber</a>



Section Number	Title	Page
14.1.2.2	Power Down Mode .....	223
14.1.3	Block Diagram .....	223
14.2	External Signal Description .....	224
14.2.1	ADP7–ADP0 — Channel Input Pins .....	225
14.2.2	$V_{REFH}$ , $V_{REFL}$ — ATD Reference Pins .....	225
14.2.3	$V_{DDAD}$ , $V_{SSAD}$ — ATD Supply Pins .....	225
14.3	Register Definition .....	225
14.3.1	ATD Control (ATDC) .....	225
14.3.2	ATD Status and Control (ATDSC) .....	228
14.3.3	ATD Result Data (ATDRH, ATDRL) .....	229
14.3.4	ATD Pin Enable (ATDPE) .....	229
14.4	Functional Description .....	230
14.4.1	Mode Control .....	230
14.4.2	Sample and Hold .....	230
14.4.3	Analog Input Multiplexer .....	232
14.4.4	ATD Module Accuracy Definitions .....	232
14.5	Resets .....	235
14.6	Interrupts .....	235

## Chapter 15 Development Support

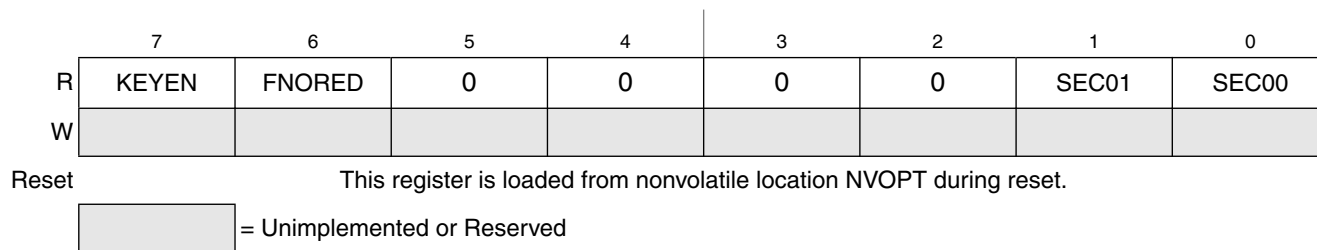
15.1	Introduction .....	237
15.1.1	Features .....	238
15.2	Background Debug Controller (BDC) .....	238
15.2.1	BKGD Pin Description .....	239
15.2.2	Communication Details .....	240
15.2.3	BDC Commands .....	244
15.2.4	BDC Hardware Breakpoint .....	246
15.3	On-Chip Debug System (DBG) .....	247
15.3.1	Comparators A and B .....	247
15.3.2	Bus Capture Information and FIFO Operation .....	247
15.3.3	Change-of-Flow Information .....	248
15.3.4	Tag vs. Force Breakpoints and Triggers .....	248
15.3.5	Trigger Modes .....	249
15.3.6	Hardware Breakpoints .....	251
15.4	Register Definition .....	251
15.4.1	BDC Registers and Control Bits .....	251
15.4.1.1	BDC Status and Control Register (BDCSCR) .....	252
15.4.1.2	BDC Breakpoint Match Register (BDCBKPT) .....	253
15.4.2	System Background Debug Force Reset Register (SBDFFR) .....	253
15.4.3	DBG Registers and Control Bits .....	254
15.4.3.1	Debug Comparator A High Register (DBGCAH) .....	254

**Table 4-7. FLASH Clock Divider Settings**

$f_{Bus}$	PRDIV8 (Binary)	DIV5:DIV0 (Decimal)	$f_{FCLK}$	Program/Erase Timing Pulse (5 $\mu$ s Min, 6.7 $\mu$ s Max)
20 MHz	1	12	192.3 kHz	5.2 $\mu$ s
10 MHz	0	49	200 kHz	5 $\mu$ s
8 MHz	0	39	200 kHz	5 $\mu$ s
4 MHz	0	19	200 kHz	5 $\mu$ s
2 MHz	0	9	200 kHz	5 $\mu$ s
1 MHz	0	4	200 kHz	5 $\mu$ s
200 kHz	0	0	200 kHz	5 $\mu$ s
150 kHz	0	0	150 kHz	6.7 $\mu$ s

## 4.6.2 FLASH Options Register (FOPT and NVOPT)

During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into FOPT. Bits 5 through 2 are not used and always read 0. This register may be read at any time, but writes have no meaning or effect. To change the value in this register, erase and reprogram the NVOPT location in FLASH memory as usual and then issue a new MCU reset.


**Figure 4-6. FLASH Options Register (FOPT)**
**Table 4-8. FOPT Field Descriptions**

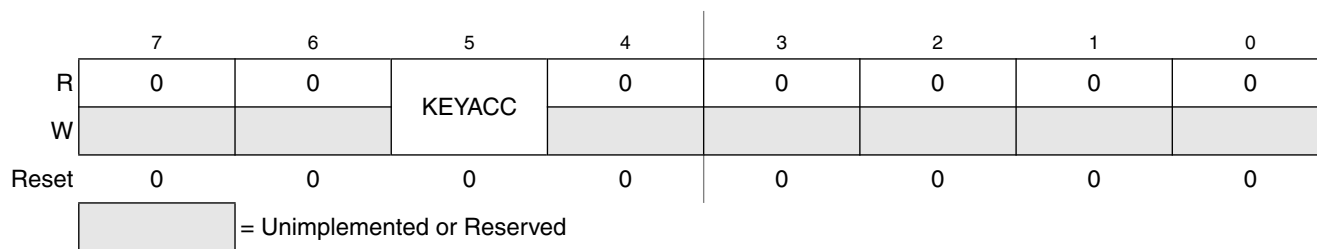
Field	Description
7 KEYEN	<p><b>Backdoor Key Mechanism Enable</b> — When this bit is 0, the backdoor key mechanism cannot be used to disengage security. The backdoor key mechanism is accessible only from user (secured) firmware. BDM commands cannot be used to write key comparison values that would unlock the backdoor key. For more detailed information about the backdoor key mechanism, refer to <a href="#">Section 4.5, “Security.”</a></p> <p>0 No backdoor key access allowed.</p> <p>1 If user firmware writes an 8-byte value that matches the nonvolatile backdoor key (NVBACKKEY through NVBACKKEY+7, in that order), security is temporarily disengaged until the next MCU reset.</p>

**Table 4-8. FOPT Field Descriptions (continued)**

Field	Description
6 FNORED	<b>Vector Redirection Disable</b> — When this bit is 1, vector redirection is disabled. 0 Vector redirection enabled. 1 Vector redirection disabled.
1:0 SEC0[1:0]	<b>Security State Code</b> — This 2-bit field determines the security state of the MCU as shown below. When the MCU is secure, the contents of RAM and FLASH memory cannot be accessed by instructions from any unsecured source including the background debug interface. For more detailed information about security, refer to <a href="#">Section 4.5, “Security.”</a> 00 Secure 01 Secure 10 Unsecured 11 Secure SEC0[1:0] changes to 10 after successful backdoor key entry or a successful blank check of FLASH.

### 4.6.3 FLASH Configuration Register (FCNFG)

Bits 7 through 5 may be read or written at any time. Bits 4 through 0 always read 0 and cannot be written.


**Figure 4-7. FLASH Configuration Register (FCNFG)**
**Table 4-9. FCNFG Field Descriptions**

Field	Description
5 KEYACC	<b>Enable Writing of Access Key</b> — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to <a href="#">Section 4.5, “Security.”</a> 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a FLASH programming or erase command. 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes. Reads of the FLASH return invalid data.

### 4.6.4 FLASH Protection Register (FPROT and NVPROT)

During reset, the contents of the nonvolatile location NVPROT are copied from FLASH into FPROT. This register may be read at any time, but user program writes have no meaning or effect. Background debug commands can write to FPROT.

**Table 4-11. FSTAT Field Descriptions (continued)**

Field	Description
5 FPVIOL	<b>Protection Violation Flag</b> — FPVIOL is set automatically when FCBEF is cleared to register a command that attempts to erase or program a location in a protected block (the erroneous command is ignored). FPVIOL is cleared by writing a 1 to FPVIOL. 0 No protection violation. 1 An attempt was made to erase or program a protected location.
4 FACCERR	<b>Access Error Flag</b> — FACCERR is set automatically when the proper command sequence is not followed exactly (the erroneous command is ignored), if a program or erase operation is attempted before the FCDIV register has been initialized, or if the MCU enters stop while a command was in progress. For a more detailed discussion of the exact actions that are considered access errors, see <a href="#">Section 4.4.5, “Access Errors.”</a> FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error has occurred. 1 An access error has occurred.
2 FBLANK	<b>FLASH Verified as All Blank (Erased) Flag</b> — FBLANK is set automatically at the conclusion of a blank check command if the entire FLASH array was verified to be erased. FBLANK is cleared by clearing FCBEF to write a new valid command. Writing to FBLANK has no meaning or effect. 0 After a blank check command is completed and FCCF = 1, FBLANK = 0 indicates the FLASH array is not completely erased. 1 After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the FLASH array is completely erased (all 0xFF).

#### 4.6.6 FLASH Command Register (FCMD)

Only five command codes are recognized in normal user modes as shown in [Table 4-13](#). 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 4-10. FLASH Command Register (FCMD)**
**Table 4-12. FCMD Field Descriptions**

Field	Description
7:0 FCMD[7:0]	<b>FLASH Command Bits</b> -- See <a href="#">Table 4-13</a> for a description of FCMD[7:0].

## 8.5 HCS08 Instruction Set Summary

### Instruction Set Summary Nomenclature

The nomenclature listed here is used in the instruction descriptions in [Table 8-2](#).

#### Operators

- ( ) = Contents of register or memory location shown inside parentheses
- ← = Is loaded with (read: “gets”)
- & = Boolean AND
- | = Boolean OR
- ⊕ = Boolean exclusive-OR
- × = Multiply
- ÷ = Divide
- :
- + = Add
- = Negate (two’s complement)

#### CPU registers

- A = Accumulator
- CCR = Condition code register
- H = Index register, higher order (most significant) 8 bits
- X = Index register, lower order (least significant) 8 bits
- PC = Program counter
- PCH = Program counter, higher order (most significant) 8 bits
- PCL = Program counter, lower order (least significant) 8 bits
- SP = Stack pointer

#### Memory and addressing

- M = A memory location or absolute data, depending on addressing mode
- M:M + 0x0001 = A 16-bit value in two consecutive memory locations. The higher-order (most significant) 8 bits are located at the address of M, and the lower-order (least significant) 8 bits are located at the next higher sequential address.

#### Condition code register (CCR) bits

- V = Two’s complement overflow indicator, bit 7
- H = Half carry, bit 4
- I = Interrupt mask, bit 3
- N = Negative indicator, bit 2
- Z = Zero indicator, bit 1
- C = Carry/borrow, bit 0 (carry out of bit 7)

#### CCR activity notation

- = Bit not affected

Table 8-2. HCS08 Instruction Set Summary (Sheet 5 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles <sup>1</sup>
			V	H	I	N	Z	C				
LDX #opr8i LDX opr8a LDX opr16a LDX oprx16,X LDX oprx8,X LDX ,X LDX oprx16,SP LDX oprx8,SP	Load X (Index Register Low) from Memory	$X \leftarrow (M)$	0	-	-	↑	↑	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEF	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
LSL opr8a LSLA LSLX LSL oprx8,X LSL ,X LSL oprx8,SP	Logical Shift Left (Same as ASL)		↑	-	-	↑	↑	↑	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
LSR opr8a LSRA LSRX LSR oprx8,X LSR ,X LSR oprx8,SP	Logical Shift Right		↑	-	-	0	↑	↑	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	5 1 1 5 4 6
MOV opr8a,opr8a MOV opr8a,X+ MOV #opr8i,opr8a MOV ,X+,opr8a	Move	$(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$ $H:X \leftarrow (H:X) + 0x0001$ in IX+/DIR and DIR/IX+ Modes	0	-	-	↑	↑	-	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E 5E 6E 7E	dd dd dd ii ii dd dd	5 5 4 5
MUL	Unsigned multiply	$X:A \leftarrow (X) \times (A)$	-	0	-	-	-	0	INH	42		5
NEG opr8a NEGA NEGX NEG oprx8,X NEG ,X NEG oprx8,SP	Negate (Two's Complement)	$M \leftarrow -(M) = 0x00 - (M)$ $A \leftarrow -(A) = 0x00 - (A)$ $X \leftarrow -(X) = 0x00 - (X)$ $M \leftarrow -(M) = 0x00 - (M)$ $M \leftarrow -(M) = 0x00 - (M)$ $M \leftarrow -(M) = 0x00 - (M)$				↑	↑	↑	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	5 1 1 5 4 6
NOP	No Operation	Uses 1 Bus Cycle	-	-	-	-	-	-	INH	9D		1
NSA	Nibble Swap Accumulator	$A \leftarrow (A[3:0]:A[7:4])$	-	-	-	-	-	-	INH	62		1
ORA #opr8i ORA opr8a ORA opr16a ORA oprx16,X ORA oprx8,X ORA ,X ORA oprx16,SP ORA oprx8,SP	Inclusive OR Accumulator and Memory	$A \leftarrow (A)   (M)$	0	-	-	↑	↑	-	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA BA CA DA EA FA 9EDA 9EEA	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
PSHA	Push Accumulator onto Stack	Push (A); $SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	87		2
PSHH	Push H (Index Register High) onto Stack	Push (H); $SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	8B		2
PSHX	Push X (Index Register Low) onto Stack	Push (X); $SP \leftarrow (SP) - 0x0001$	-	-	-	-	-	-	INH	89		2
PULA	Pull Accumulator from Stack	$SP \leftarrow (SP + 0x0001)$ ; Pull (A)	-	-	-	-	-	-	INH	86		3
PULH	Pull H (Index Register High) from Stack	$SP \leftarrow (SP + 0x0001)$ ; Pull (H)	-	-	-	-	-	-	INH	8A		3
PULX	Pull X (Index Register Low) from Stack	$SP \leftarrow (SP + 0x0001)$ ; Pull (X)	-	-	-	-	-	-	INH	88		3
ROL opr8a ROLA ROLX ROL oprx8,X ROL ,X ROL oprx8,SP	Rotate Left through Carry		↑	-	-	↑	↑	↑	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	5 1 1 5 4 6



### 9.4.7.1 FLL Engaged External Unlocked

FEE unlocked is entered when FEE is entered and the count error ( $\Delta n$ ) output from the subtractor is greater than the maximum  $n_{\text{unlock}}$  or less than the minimum  $n_{\text{unlock}}$ , as required by the lock detector to detect the unlock condition.

The ICG will remain in this state while the count error ( $\Delta n$ ) is greater than the maximum  $n_{\text{lock}}$  or less than the minimum  $n_{\text{lock}}$ , as required by the lock detector to detect the lock condition.

In this state, the pulse counter, subtractor, digital loop filter, and DCO form a closed loop and attempt to lock it according to their operational descriptions later in this section. Upon entering this state and until the FLL becomes locked, the output clock signal ICGOUT frequency is given by  $f_{\text{ICGDCLK}} / (2 \times R)$ . This extra divide by two prevents frequency overshoots during the initial locking process from exceeding chip-level maximum frequency specifications. After the FLL has locked, if an unexpected loss of lock causes it to re-enter the unlocked state while the ICG remains in FEE mode, the output clock signal ICGOUT frequency is given by  $f_{\text{ICGDCLK}} / R$ .

### 9.4.7.2 FLL Engaged External Locked

FEE locked is entered from FEE unlocked when the count error ( $\Delta n$ ) is less than  $n_{\text{lock}}$  (max) and greater than  $n_{\text{lock}}$  (min) for a given number of samples, as required by the lock detector to detect the lock condition. The output clock signal ICGOUT frequency is given by  $f_{\text{ICGDCLK}}/R$ . In FLL engaged external locked, the filter value is updated only once every four comparison cycles. The update made is an average of the error measurements taken in the four previous comparisons.

## 9.4.8 FLL Lock and Loss-of-Lock Detection

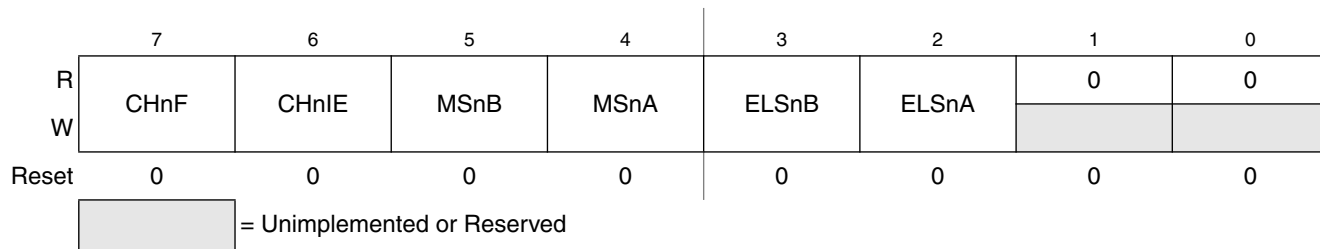
To determine the FLL locked and loss-of-lock conditions, the pulse counter counts the pulses of the DCO for one comparison cycle (see [Table 9-9](#) for explanation of a comparison cycle) and passes this number to the subtractor. The subtractor compares this value to the value in MFD and produces a count error,  $\Delta n$ . To achieve locked status,  $\Delta n$  must be between  $n_{\text{lock}}$  (min) and  $n_{\text{lock}}$  (max). After the FLL has locked,  $\Delta n$  must stay between  $n_{\text{unlock}}$  (min) and  $n_{\text{unlock}}$  (max) to remain locked. If  $\Delta n$  goes outside this range unexpectedly, the LOLS status bit is set and remains set until cleared by software or until the MCU is reset. LOLS is cleared by reading ICGS1 then writing 1 to ICGIF (LOLRE = 0), or by a loss-of-lock induced reset (LOLRE = 1), or by any MCU reset.

If the ICG enters the off state due to stop mode when ENBDM = OSCSTEN = 0, the FLL loses locked status (LOCK is cleared), but LOLS remains unchanged because this is not an unexpected loss-of-lock condition. Though it would be unusual, if ENBDM is cleared to 0 while the MCU is in stop, the ICG enters the off state. Because this is an unexpected stopping of clocks, LOLS will be set when the MCU wakes up from stop.

Expected loss of lock occurs when the MFD or CLKS bits are changed or in FEI mode only, when the TRIM bits are changed. In these cases, the LOCK bit will be cleared until the FLL regains lock, but the LOLS will not be set.

### 10.3.4 Timer x Channel n Status and Control Register (TPMxCnSC)

TPMxCnSC contains the channel interrupt status flag and control bits that are used to configure the interrupt enable, channel configuration, and pin function.



**Figure 10-8. Timer x Channel n Status and Control Register (TPMxCnSC)**

**Table 10-4. TPMxCnSC Register Field Descriptions**

Field	Description
7 CHnF	<p><b>Channel n Flag</b> — When channel n is configured for input capture, this flag bit is set when an active edge occurs on the channel n pin. When channel n is an output compare or edge-aligned PWM channel, CHnF is set when the value in the TPM counter registers matches the value in the TPM channel n value registers. This flag is seldom used with center-aligned PWMs because it is set every time the counter matches the channel value register, which correspond to both edges of the active duty cycle period.</p> <p>A corresponding interrupt is requested when CHnF is set and interrupts are enabled (CHnIE = 1). Clear CHnF by reading TPMxCnSC while CHnF is set and then writing a 0 to CHnF. If another interrupt request occurs before the clearing sequence is complete, the sequence is reset so CHnF would remain set after the clear sequence was completed for the earlier CHnF. This is done so a CHnF interrupt request cannot be lost by clearing a previous CHnF. Reset clears CHnF. Writing a 1 to CHnF has no effect.</p> <p>0 No input capture or output compare event occurred on channel n 1 Input capture or output compare event occurred on channel n</p>
6 CHnIE	<p><b>Channel n Interrupt Enable</b> — This read/write bit enables interrupts from channel n. Reset clears CHnIE.</p> <p>0 Channel n interrupt requests disabled (use software polling) 1 Channel n interrupt requests enabled</p>
5 MSnB	<p><b>Mode Select B for TPM Channel n</b> — When CPWMS = 0, MSnB = 1 configures TPM channel n for edge-aligned PWM mode. For a summary of channel mode and setup controls, refer to <a href="#">Table 10-5</a>.</p>
4 MSnA	<p><b>Mode Select A for TPM Channel n</b> — When CPWMS = 0 and MSnB = 0, MSnA configures TPM channel n for input capture mode or output compare mode. Refer to <a href="#">Table 10-5</a> for a summary of channel mode and setup controls.</p>
3:2 ELSn[B:A]	<p><b>Edge/Level Select Bits</b> — Depending on the operating mode for the timer channel as set by CPWMS:MSnB:MSnA and shown in <a href="#">Table 10-5</a>, these bits select the polarity of the input edge that triggers an input capture event, select the level that will be driven in response to an output compare match, or select the polarity of the PWM output.</p> <p>Setting ELSnB:ELSnA to 0:0 configures the related timer pin as a general-purpose I/O pin unrelated to any timer channel functions. This function is typically used to temporarily disable an input capture channel or to make the timer pin available as a general-purpose I/O pin when the associated timer channel is set up as a software timer that does not require the use of a pin.</p>

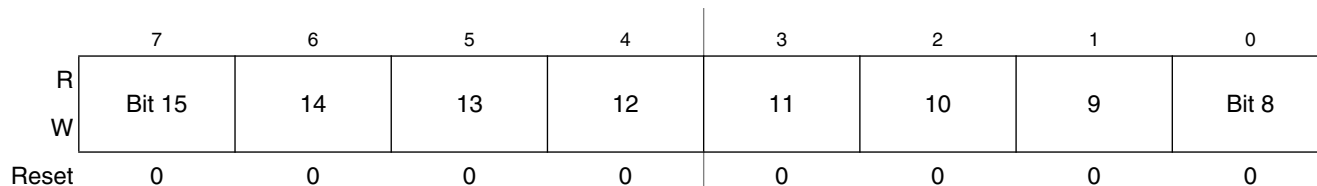
**Table 10-5. Mode, Edge, and Level Selection**

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)

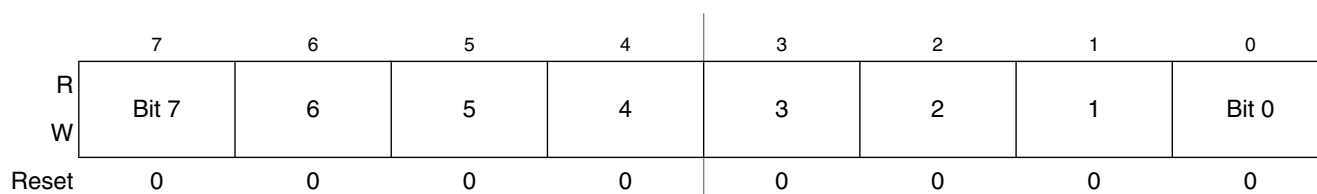
If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts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

### 10.3.5 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.

### 10.4.2.2 Output Compare Mode

With the output compare function, the TPM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter reaches the value in the channel value registers of an output compare channel, the TPM can set, clear, or toggle the channel pin.

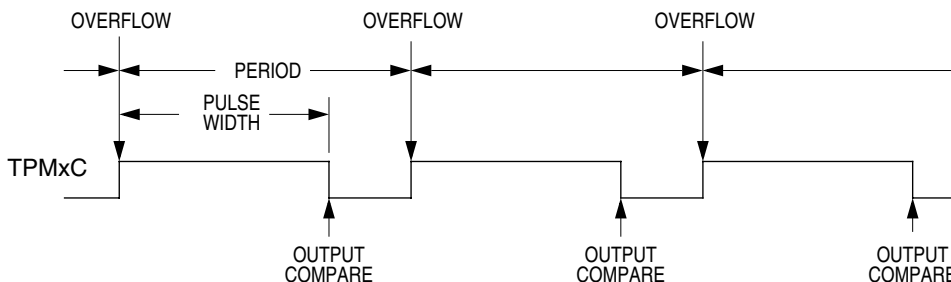
In output compare mode, values are transferred to the corresponding timer channel value registers only after both 8-bit bytes of a 16-bit register have been written. This coherency sequence can be manually reset by writing to the channel status/control register (TPMxCnSC).

An output compare event sets a flag bit (CHnF) that can optionally generate a CPU interrupt request.

### 10.4.2.3 Edge-Aligned PWM Mode

This type of PWM output uses the normal up-counting mode of the timer counter (CPWMS = 0) and can be used when other channels in the same TPM are configured for input capture or output compare functions. The period of this PWM signal is determined by the setting in the modulus register (TPMxMODH:TPMxMODL). The duty cycle is determined by the setting in the timer channel value register (TPMxCnVH:TPMxCnVL). The polarity of this PWM signal is determined by the setting in the ELSnA control bit. Duty cycle cases of 0 percent and 100 percent are possible.

As [Figure 10-11](#) shows, the output compare value in the TPM channel registers determines the pulse width (duty cycle) of the PWM signal. The time between the modulus overflow and the output compare is the pulse width. If ELSnA = 0, the counter overflow forces the PWM signal high and the output compare forces the PWM signal low. If ELSnA = 1, the counter overflow forces the PWM signal low and the output compare forces the PWM signal high.



**Figure 10-11. PWM Period and Pulse Width (ELSnA = 0)**

When the channel value register is set to 0x0000, the duty cycle is 0 percent. By setting the timer channel value register (TPMxCnVH:TPMxCnVL) to a value greater than the modulus setting, 100% duty cycle can be achieved. This implies that the modulus setting must be less than 0xFFFF to get 100% duty cycle.

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 either register, TPMxCnVH or TPMxCnVL, write to buffer registers. In edge-PWM mode, 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 value in the TPMxCnTH:TPMxCnTL counter is 0x0000. (The new duty cycle does not take effect until the next full period.)

### 11.3.3.2.1 Idle-Line Wakeup

When WAKE = 0, the receiver is configured for idle-line wakeup. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode that determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When the RWU bit is set, the idle character that wakes a receiver does not set the receiver idle bit, IDLE, or the receive data register full flag, RDRF. It therefore will not generate an interrupt when this idle character occurs. The receiver will wake up and wait for the next data transmission which will set RDRF and generate an interrupt if enabled.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

### 11.3.3.2.2 Address-Mark Wakeup

When WAKE = 1, the receiver is configured for address-mark wakeup. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode).

Address-mark wakeup allows messages to contain idle characters but requires that the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the receivers RWU bit before the stop bit is received and sets the RDRF flag.

## 11.3.4 Interrupts and Status Flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF and IDLE events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these eight interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCIxD. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD1 high. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever TC = 1. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCIxD. The RDRF flag is cleared by reading SCIxS1 while RDRF = 1 and then reading SCIxD.

### 13.3.5 IIC Data I/O Register (IICD)

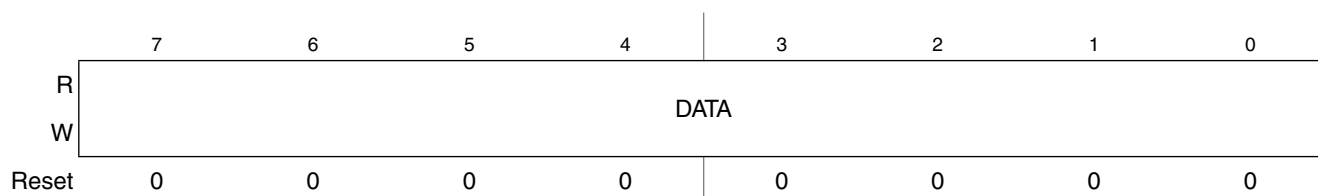


Figure 13-7. IIC Data I/O Register (IICD)

Table 13-6. IICD Register Field Descriptions

Field	Description
7:0 DATA	<b>Data</b> — In master transmit mode, when data is written to the IICD, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data.

**NOTE**

When transmitting out of master receive mode, the IIC mode should be switched before reading the IICD register to prevent an inadvertent initiation of a master receive data transfer.

In slave mode, the same functions are available after an address match has occurred.

Note that the TX bit in IICC must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, then reading the IICD will not initiate the receive.

Reading the IICD will return the last byte received while the IIC is configured in either master receive or slave receive modes. The IICD does not reflect every byte that is transmitted on the IIC bus, nor can software verify that a byte has been written to the IICD correctly by reading it back.

In master transmit mode, the first byte of data written to IICD following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7–bit 1) concatenated with the required R/W bit (in position bit 0).

**Table 14-3. Available Result Data Formats**

RES8	DJM	SGN	Data Formats of Result	Analog Input $V_{REFH} = V_{DDA}, V_{REFL} = V_{SSA}$ ATDRH:ATDRL	
				$V_{DDA}$	$V_{SSA}$
1	0	0	8-bit : left justified : unsigned	\$FF:\$00	\$00:\$00
1	0	1	8-bit : left justified : signed	\$7F:\$00	\$80:\$00
1	1	X <sup>1</sup>	8-bit : left justified <sup>2</sup> : unsigned	\$FF:\$00	\$00:\$00
0	0	0	10-bit : left justified : unsigned	\$FF:\$C0	\$00:\$00
0	0	1	10-bit : left justified : signed	\$7F:\$C0	\$80:\$00
0	1	X <sup>1</sup>	10-bit : right justified : unsigned	\$03:\$FF	\$00:\$00

<sup>1</sup> The SGN bit is only effective when DJM = 0. When DJM = 1, SGN is ignored.

<sup>2</sup> 8-bit results are always in ATDRH.

**Table 14-4. Clock Prescaler Values**

PRS	Factor = (PRS + 1) × 2	Max Bus Clock MHz (2 MHz max ATD Clock) <sup>1</sup>	Max Bus Clock MHz (1 MHz max ATD Clock) <sup>2</sup>	Min Bus Clock <sup>3</sup> MHz (500 kHz min ATD Clock)
0000	2	4	2	1
0001	4	8	4	2
0010	6	12	6	3
0011	8	16	8	4
0100	10	20	10	5
0101	12	20	12	6
0110	14	20	14	7
0111	16	20	16	8
1000	18	20	18	9
1001	20	20	20	10
1010	22	20	20	11
1011	24	20	20	12
1100	26	20	20	13
1101	28	20	20	14
1110	30	20	20	15
1111	32	20	20	16

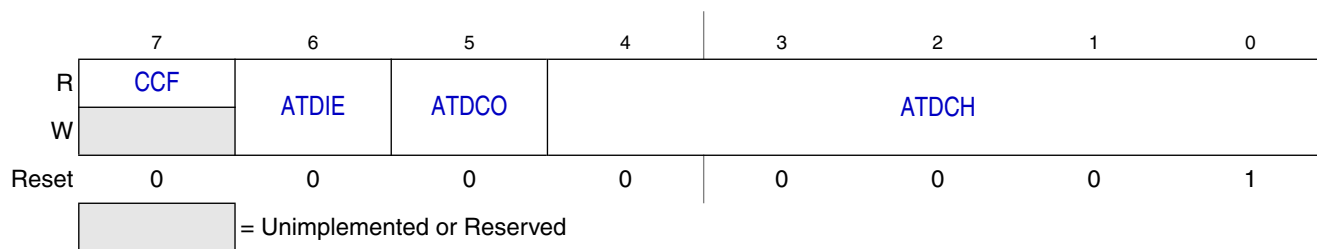
<sup>1</sup> Maximum ATD conversion clock frequency is 2 MHz. The max bus clock frequency is computed from the max ATD conversion clock frequency times the indicated prescaler setting; i.e., for a PRS of 0, max bus clock = 2 (max ATD conversion clock frequency) × 2 (Factor) = 4 MHz.

<sup>2</sup> Use these settings if the maximum desired ATD conversion clock frequency is 1 MHz. The max bus clock frequency is computed from the max ATD conversion clock frequency times the indicated prescaler setting; i.e., for a PRS of 0, max bus clock = 1 (max ATD conversion clock frequency) × 2 (Factor) = 2 MHz.

<sup>3</sup> Minimum ATD conversion clock frequency is 500 kHz. The min bus clock frequency is computed from the min ATD conversion clock frequency times the indicated prescaler setting; i.e., for a PRS of 1, min bus clock = 0.5 (min ATD conversion clock frequency) × 2 (Factor) = 1 MHz.

### 14.3.2 ATD Status and Control (ATDSC)

Writes to the ATD status and control register clears the CCF flag, cancels any pending interrupts, and initiates a new conversion.



**Figure 14-6. ATD Status and Control Register (ATDSC)**

**Table 14-5. ATDSC Register Field Descriptions**

Field	Description
7 CCF	<b>Conversion Complete Flag</b> — The CCF is a read-only bit which is set each time a conversion is complete. The CCF bit is cleared whenever the ATDSC register is written. It is also cleared whenever the result registers, ATDRH or ATDRL, are read. 0 Current conversion is not complete. 1 Current conversion is complete.
6 ATDIE	<b>ATD Interrupt Enabled</b> — When this bit is set, an interrupt is generated upon completion of an ATD conversion. At this time, the result registers contain the result data generated by the conversion. The interrupt will remain pending as long as the conversion complete flag CCF is set. If the ATDIE bit is cleared, then the CCF bit must be polled to determine when the conversion is complete. Note that system reset clears pending interrupts. 0 ATD interrupt disabled. 1 ATD interrupt enabled.
5 ATDCO	<b>ATD Continuous Conversion</b> — When this bit is set, the ATD will convert samples continuously and update the result registers at the end of each conversion. When this bit is cleared, only one conversion is completed between writes to the ATDSC register. 0 Single conversion mode. 1 Continuous conversion mode.
4:0 ATDCH	<b>Analog Input Channel Select</b> — This field of bits selects the analog input channel whose signal is sampled and converted to digital codes. <a href="#">Table 14-6</a> lists the coding used to select the various analog input channels.

**Table 14-6. Analog Input Channel Select Coding**

ATDCH	Analog Input Channel
00	AD0
01	AD1
02	AD2
03	AD3
04	AD4
05	AD5
06	AD6
07	AD7
08–1D	Reserved (default to $V_{REFL}$ )
1E	$V_{REFH}$
1F	$V_{REFL}$



## 14.4 Functional Description

The ATD uses a successive approximation register (SAR) architecture. The ATD contains all the necessary elements to perform a single analog-to-digital conversion.

A write to the ATDSC register initiates a new conversion. A write to the ATDC register will interrupt the current conversion but it will not initiate a new conversion. A write to the ATDPE register will also abort the current conversion but will not initiate a new conversion. If a conversion is already running when a write to the ATDSC register is made, it will be aborted and a new one will be started.

### 14.4.1 Mode Control

The ATD has a mode control unit to communicate with the sample and hold (S/H) machine and the SAR machine when necessary to collect samples and perform conversions. The mode control unit signals the S/H machine to begin collecting a sample and for the SAR machine to begin receiving a sample. At the end of the sample period, the S/H machine signals the SAR machine to begin the analog-to-digital conversion process. The conversion process is terminated when the SAR machine signals the end of conversion to the mode control unit. For  $V_{REFL}$  and  $V_{REFH}$ , the SAR machine uses the reference potentials to set the sampled signal level within itself without relying on the S/H machine to deliver them.

The mode control unit organizes the conversion, specifies the input sample channel, and moves the digital output data from the SAR register to the result register. The result register consists of a dual-port register. The SAR register writes data into the register through one port while the module data bus reads data out of the register through the other port.

### 14.4.2 Sample and Hold

The S/H machine accepts analog signals and stores them as capacitor charge on a storage node located in the SAR machine. Only one sample can be held at a time so the S/H machine and the SAR machine can not run concurrently even though they are independent machines. [Figure 14-10](#) shows the placement of the various resistors and capacitors.

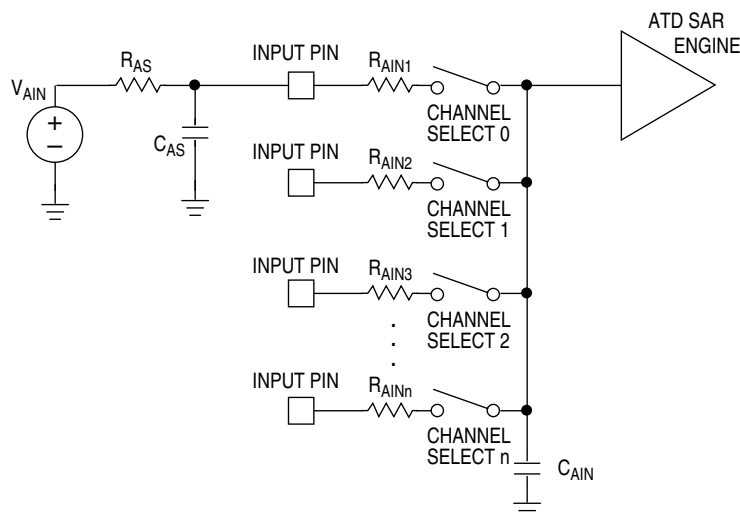


Figure 14-10. Resistor and Capacitor Placement

## 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:

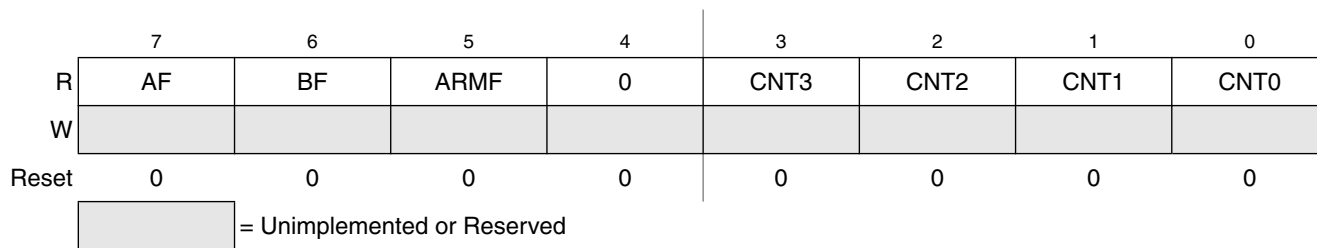
- Generation of a breakpoint to the CPU
- Storage of data bus values into the FIFO
- Starting to store change-of-flow addresses into the FIFO (begin type trace)
- Stopping the storage of change-of-flow addresses into the FIFO (end type trace)

### 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

### 15.4.3.9 Debug Status Register (DBGS)

This is a read-only status register.



**Figure 15-9. Debug Status Register (DBGS)**

**Table 15-6. DBGS Register Field Descriptions**

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

- 1 Self-clocked mode frequency is the frequency that the DCO generates when the FLL is open-loop.
- 2 Loss of reference frequency is the reference frequency detected internally, which transitions the ICG into self-clocked mode if it is not in the desired range.
- 3 Loss of DCO frequency is the DCO frequency detected internally, which transitions the ICG into FLL bypassed external mode (if an external reference exists) if it is not in the desired range.
- 4 This parameter is characterized before qualification rather than 100% tested.
- 5 Proper PC board layout procedures must be followed to achieve specifications.
- 6 This specification applies to the period of time required for the FLL to lock after entering FLL engaged internal or external modes. If a crystal/resonator is being used as the reference, this specification assumes it is already running.
- 7 Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum  $f_{ICGOUT}$ . Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the FLL circuitry via  $V_{DDA}$  and  $V_{SSA}$  and variation in crystal oscillator frequency increase the  $C_{Jitter}$  percentage for a given interval.
- 8 See [Figure A-10](#)

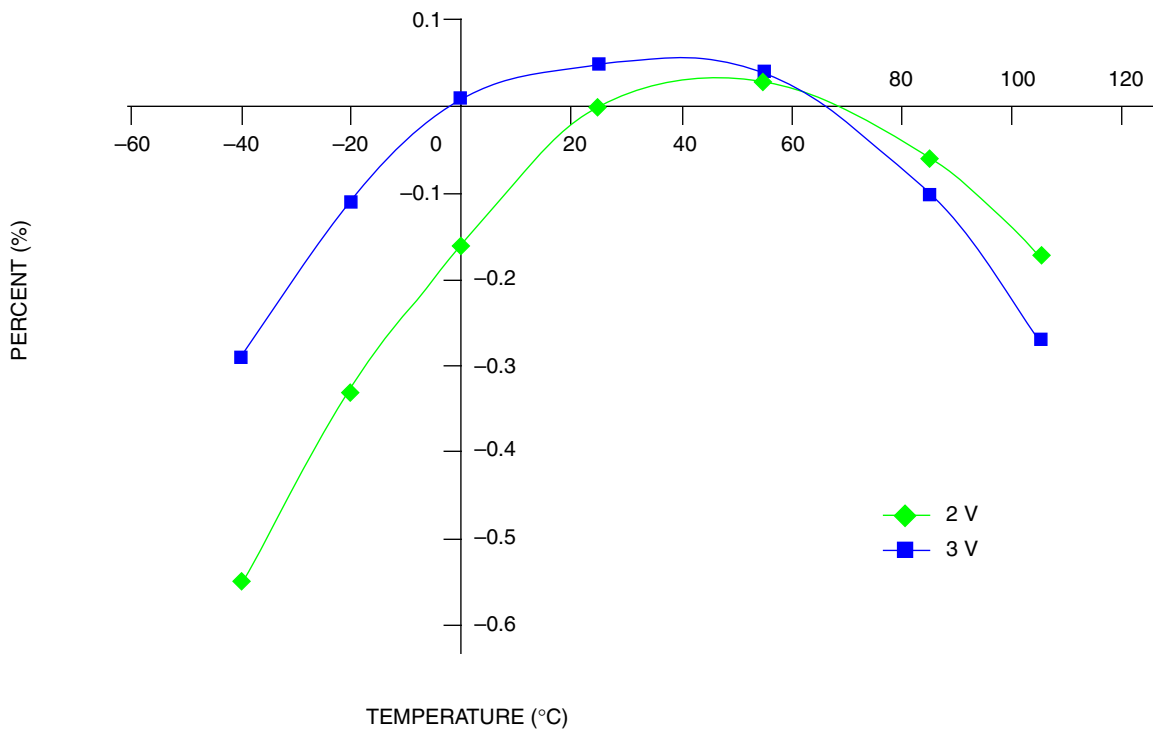
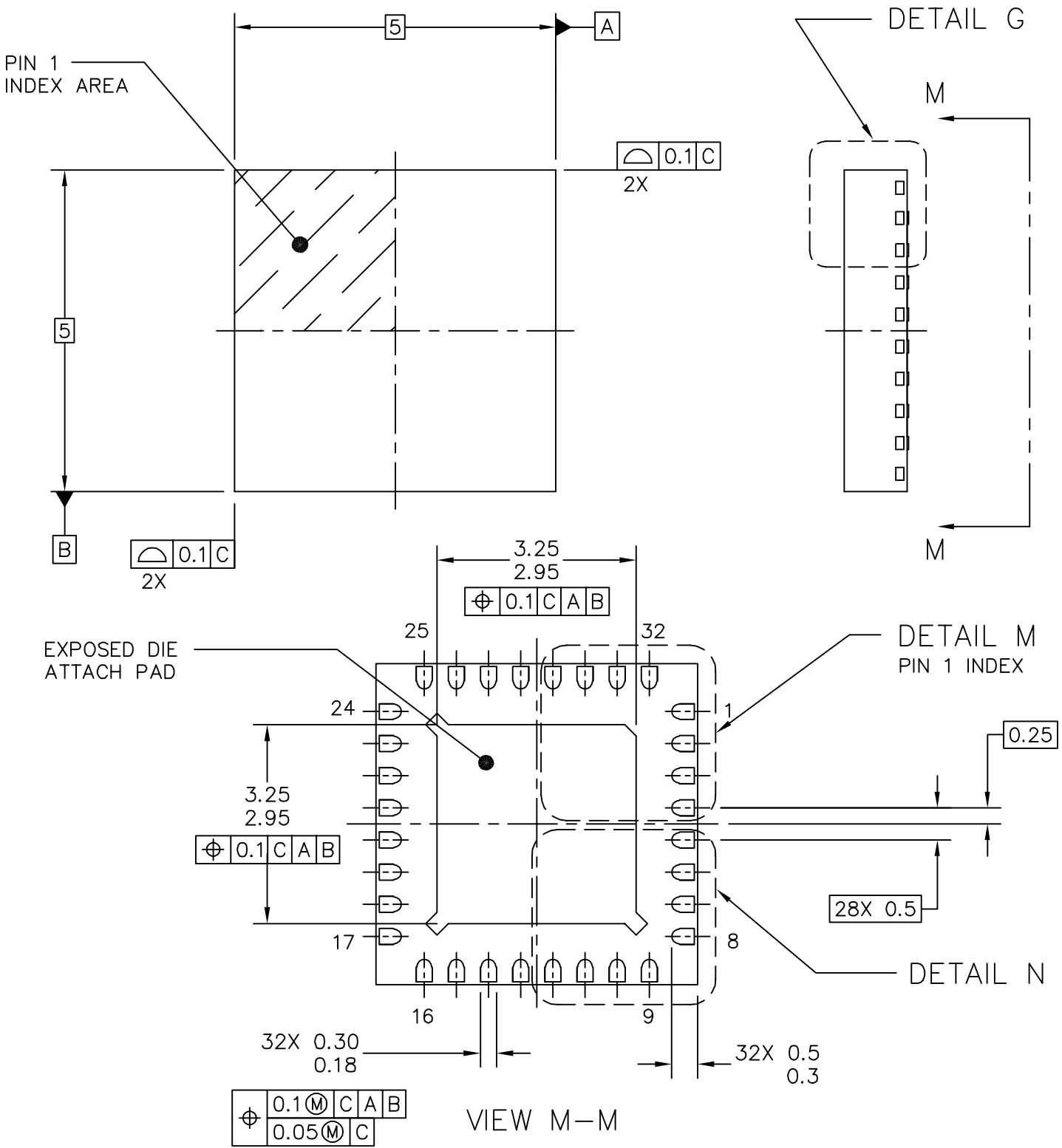


Figure A-10. Internal Oscillator Deviation from Trimmed Frequency

## A.10 AC Characteristics

This section describes ac timing characteristics for each peripheral system. For detailed information about how clocks for the bus are generated, see [Chapter 9, “Internal Clock Generator \(S08ICGV4\).”](#)



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	<b>MECHANICAL OUTLINE</b>	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED QUAD FLAT NON-LEADED PACKAGE (QFN) 32 TERMINAL, 0.5 PITCH (5 X 5 X 1) CASE OUTLINE	DOCUMENT NO: 98ARH99035A	REV: J	
	CASE NUMBER: 1311-06	17 MAR 2006	
	STANDARD: JEDEC-MO-220 VHHD-2		