

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	8MHz
Connectivity	SCI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	23
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08rd60dwe

Table 1-2. Block Versions

Module	Version
Analog Comparator (ACMP)	1
Carrier Modulator Transmitter (CMT)	1
Keyboard Interrupt (KBI)	1
Serial Communications Interface (SCI)	1
Serial Peripheral Interface (SPI)	3
Timer Pulse-Width Modulator (TPM)	1
Central Processing Unit (CPU)	2
Debug Module (DBG)	1
FLASH	1
System Control	2

1.4 System Clock Distribution

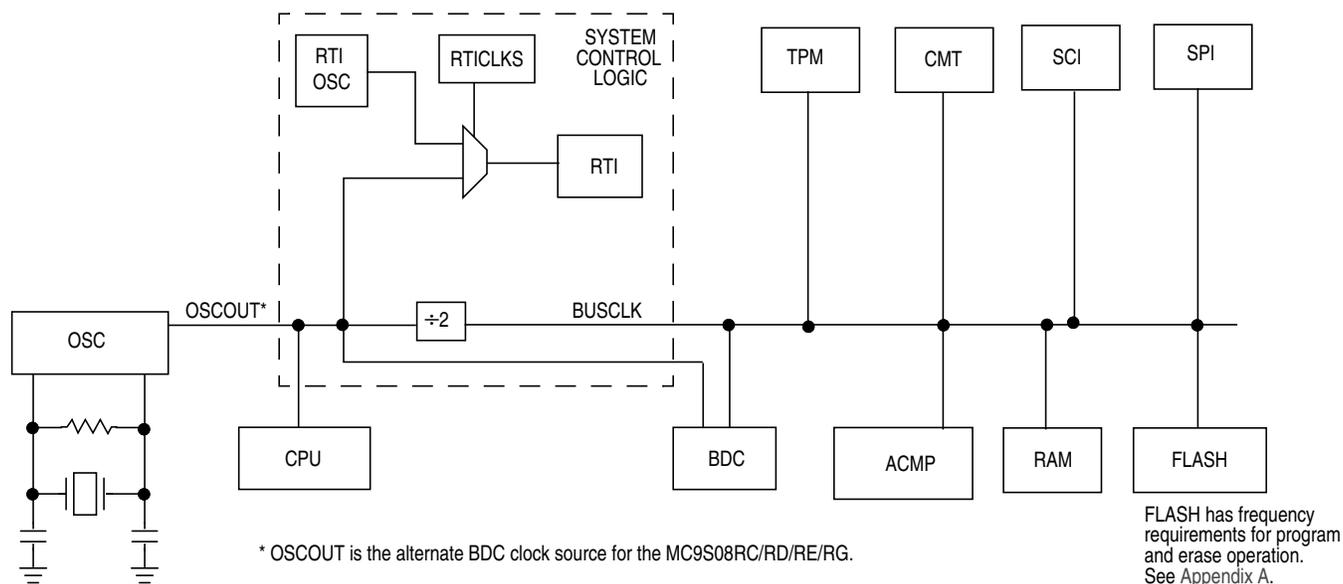


Figure 1-2. System Clock Distribution Diagram

Table 1-2 shows a simplified clock connection diagram for the MCU. The CPU operates at the input frequency of the oscillator. The bus clock frequency is half of the oscillator frequency and is used by all of the internal circuits with the exception of the CPU and RTI. The RTI can use either the oscillator input or the internal RTI oscillator as its clock source.

Chapter 3

Modes of Operation

3.1 Introduction

The operating modes of the MC9S08RC/RD/RE/RG are described in this section. Entry into each mode, exit from each mode, and functionality while in each of the modes are described.

3.2 Features

- Active background mode for code development
- Wait mode:
 - CPU shuts down to conserve power
 - System clocks running
 - Full voltage regulation maintained
- Stop modes:
 - System clocks stopped; voltage regulator in standby
 - Stop1 — Full power down of internal circuits for maximum power savings
 - Stop2 — Partial power down of internal circuits, RAM remains operational
 - Stop3 — All internal circuits powered for fast recovery

3.3 Run Mode

This is the normal operating mode for the MC9S08RC/RD/RE/RG. This mode is selected when the BKGD/MS pin is high at the rising edge of reset. In this mode, the CPU executes code from internal memory with execution beginning at the address fetched from memory at \$FFFE:\$FFFF after reset.

3.4 Active Background Mode

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

Active background mode is entered in any of five ways:

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

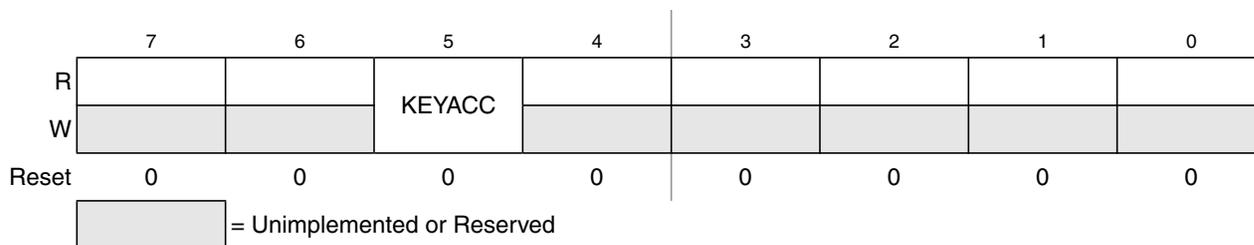


Figure 4-7. FLASH Configuration Register (FCNFG)

Table 4-8. FCNFG Field Descriptions

Field	Description
5 KEYACC	<p>Enable Writing of Access Key — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to Section 4.5, "Security."</p> <p>0 Writes to \$FFB0–\$FFB7 are interpreted as the start of a FLASH programming or erase command.</p> <p>1 Writes to NVBACKKEY (\$FFB0–\$FFB7) are interpreted as comparison key writes.</p> <p>Reads of the FLASH return invalid data.</p>

4.6.4 FLASH Protection Register (FPROT and NVPROT)

During reset, the contents of the nonvolatile location NVPROT is copied from FLASH into FPROT. Bits 0, 1, and 2 are not used and each always reads as 0. This register may be read at any time, but user program writes have no meaning or effect. Background debug commands can write to FPROT at \$1824.

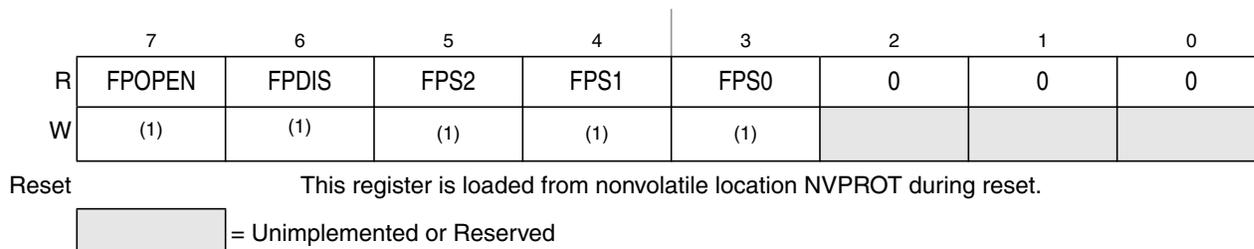


Figure 4-8. FLASH Protection Register (FPROT)

1. Background commands can be used to change the contents of these bits in FPROT.

Table 4-9. FPROT Field Descriptions

Field	Description
7 FPOPEN	<p>Open Unprotected FLASH for Program/Erase</p> <p>0 Entire FLASH memory is block protected (no program or erase allowed).</p> <p>1 Any FLASH location, not otherwise block protected or secured, may be erased or programmed.</p>
6 FPDIS	<p>FLASH Protection Disable</p> <p>0 FLASH block specified by FPS2:FPS0 is block protected (program and erase not allowed).</p> <p>1 No FLASH block is protected.</p>
5:3 FPS[2:0]	<p>FLASH Protect Size Selects — When FPDIS = 0, this 3-bit field determines the size of a protected block of FLASH locations at the high address end of the FLASH (see Table 4-10 and Table 4-11). Protected FLASH locations cannot be erased or programmed.</p>

5.8.4 System Options Register (SOPT)

This register may be read at any time. Bits 3 and 2 are unimplemented and always read 0. This is a write-once register so only the first write after reset is honored. Any subsequent attempt to write to SOPT (intentionally or unintentionally) is ignored to avoid accidental changes to these sensitive settings. SOPT must be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

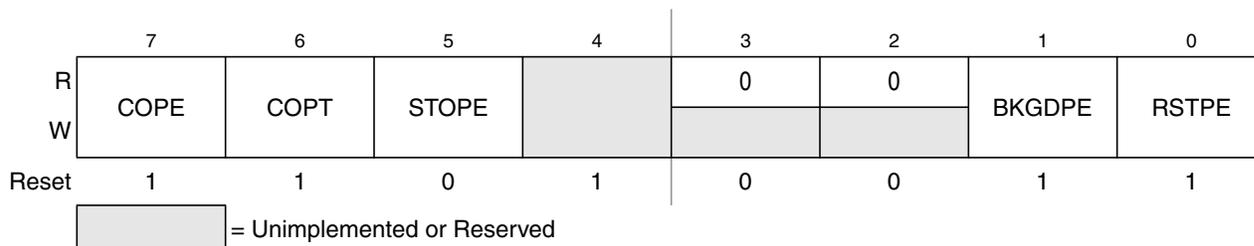


Figure 5-5. System Options Register (SOPT)

Table 5-5. SOPT Field Descriptions

Field	Description
7 COPE	COP Watchdog Enable — This write-once bit defaults to 1 after reset. 0 COP watchdog timer disabled. 1 COP watchdog timer enabled (force reset on timeout).
6 COPT	COP Watchdog Timeout — This write-once bit defaults to 1 after reset. 0 Short timeout period selected (2^{18} cycles of BUSCLK). 1 Long timeout period selected (2^{20} cycles of BUSCLK).
5 STOPE	Stop Mode Enable — This write-once bit defaults to 0 after reset, which disables stop mode. If stop mode is disabled and a user program attempts to execute a STOP instruction, an illegal opcode reset is forced. 0 Stop mode disabled. 1 Stop mode enabled.
1 BKGDPE	Background Debug Mode Pin Enable — The BKGDPE bit enables the PTD0/BKGD/MS pin to function as BKGD/MS. When the bit is clear, the pin will function as PTD0, which is an output only general purpose I/O. This pin always defaults to BKGD/MS function after any reset. 0 BKGD pin disabled. 1 BKGD pin enabled.
0 RSTPE	RESET Pin Enable — The RSTPE bit enables the PTD1/ $\overline{\text{RESET}}$ pin to function as $\overline{\text{RESET}}$. When the bit is clear, the pin will function as PTD1, which is an output only general purpose I/O. This pin always defaults to $\overline{\text{RESET}}$ function after any reset. 0 $\overline{\text{RESET}}$ pin disabled. 1 $\overline{\text{RESET}}$ pin enabled.

6.4.2 Internal Pullup Control

An internal pullup device can be enabled for each port pin that is configured as an input ($PTxDDn = 0$). The pullup device is available for a peripheral module to use, provided the peripheral is enabled and is an input function as long as the $PTxDDn = 0$.

NOTE

The voltage measured on the pulled up PTA0 pin will be less than V_{DD} . The internal gates connected to this pin are pulled all the way to V_{DD} . All other pins with enabled pullup resistors will have an unloaded measurement of V_{DD} .

6.5 Stop Modes

Depending on the stop mode, I/O functions differently as the result of executing a STOP instruction. An explanation of I/O behavior for the various stop modes follows:

- When the MCU enters stop1 mode, all internal registers, including general-purpose I/O control and data registers, are powered down. All of the general-purpose I/O pins assume their reset state: output buffers and pullups turned off. Upon exit from stop1, all I/O must be initialized as if the MCU had been reset.
- When the MCU enters stop2 mode, the internal registers are powered down as in stop1 but the I/O pin states are latched and held. For example, a port pin that is an output driving low continues to function as an output driving low even though its associated data direction and output data registers are powered down internally. Upon exit from stop2, the pins continue to hold their states until a 1 is written to the PPDACK bit. To avoid discontinuity in the pin state following exit from stop2, the user must restore the port control and data registers to the values they held before entering stop2. These values can be stored in RAM before entering stop2 because the RAM is maintained during stop2.
- In stop3 mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

6.6 Parallel I/O Registers and Control Bits

This section provides information about all registers and control bits associated with the parallel I/O ports.

Refer to tables in the Memory chapter for the absolute address assignments for all parallel I/O registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

Table 7-2. HCS08 Instruction Set Summary (Sheet 4 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles ¹	
			V	H	I	N	Z	C					
CPX #opr8i CPX opr8a CPX opr16a CPX oprx16,X CPX oprx8,X CPX ,X CPX oprx16,SP CPX oprx8,SP	Compare X (Index Register Low) with Memory	(X) – (M) (CCR Updated But Operands Not Changed)		–	–					IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 B3 C3 D3 E3 F3 9ED3 9EE3	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
DAA	Decimal Adjust Accumulator After ADD or ADC of BCD Values	(A) ₁₀	U	–	–					INH	72		1
DBNZ opr8a,rel DBNZ rel DBNZX rel DBNZ oprx8,X,rel DBNZ ,X,rel DBNZ oprx8,SP,rel	Decrement and Branch if Not Zero	Decrement A, X, or M Branch if (result) ≠ 0 DBNZX Affects X Not H	–	–	–	–	–	–		DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	7 4 4 7 6 8
DEC opr8a DECA DECX DEC oprx8,X DEC ,X DEC oprx8,SP	Decrement	M ← (M) – 0x01 A ← (A) – 0x01 X ← (X) – 0x01 M ← (M) – 0x01 M ← (M) – 0x01 M ← (M) – 0x01		–	–				–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff ff ff	5 1 1 5 4 6
DIV	Divide	A ← (H:A) ÷ (X) H ← Remainder	–	–	–	–				INH	52		6
EOR #opr8i EOR opr8a EOR opr16a EOR oprx16,X EOR oprx8,X EOR ,X EOR oprx16,SP EOR oprx8,SP	Exclusive OR Memory with Accumulator	A ← (A ⊕ M)	0	–	–			–		IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 B8 C8 D8 E8 F8 9ED8 9EE8	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
INC opr8a INCA INCX INC oprx8,X INC ,X INC oprx8,SP	Increment	M ← (M) + 0x01 A ← (A) + 0x01 X ← (X) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01		–	–			–		DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	5 1 1 5 4 6
JMP opr8a JMP opr16a JMP oprx16,X JMP oprx8,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–		DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff ff	3 4 4 3 3
JSR opr8a JSR opr16a JSR oprx16,X JSR oprx8,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 0x0001 Push (PCH); SP ← (SP) – 0x0001 PC ← Unconditional Address	–	–	–	–	–	–		DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 6 5 5
LDA #opr8i LDA opr8a LDA opr16a LDA oprx16,X LDA oprx8,X LDA ,X LDA oprx16,SP LDA oprx8,SP	Load Accumulator from Memory	A ← (M)	0	–	–			–		IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 B6 C6 D6 E6 F6 9ED6 9EE6	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 3 5 4
LDHX #opr16i LDHX opr8a LDHX opr16a LDHX ,X LDHX oprx16,X LDHX oprx8,X LDHX oprx8,SP	Load Index Register (H:X) from Memory	H:X ← (M:M + 0x0001)	0	–	–			–		IMM DIR EXT IX IX2 IX1 SP1	45 55 32 9EAE 9EBE 9ECE 9EFE	jj kk dd ll hh ll ff ff ff	3 4 5 5 6 5 5

Table 7-2. HCS08 Instruction Set Summary (Sheet 6 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles ¹
			V	H	I	N	Z	C				
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	Rotate Right through Carry			-	-				DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	5 1 1 5 4 6
RSP	Reset Stack Pointer	SP ← 0xFF (High Byte Not Affected)	-	-	-	-	-	-	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 0x0001; Pull (CCR) SP ← (SP) + 0x0001; Pull (A) SP ← (SP) + 0x0001; Pull (X) SP ← (SP) + 0x0001; Pull (PCH) SP ← (SP) + 0x0001; Pull (PCL)							INH	80		9
RTS	Return from Subroutine	SP ← SP + 0x0001; Pull (PCH) SP ← SP + 0x0001; Pull (PCL)	-	-	-	-	-	-	INH	81		6
SBC # <i>opr8i</i> SBC <i>opr8a</i> SBC <i>opr16a</i> SBC <i>opr16,X</i> SBC <i>opr8,X</i> SBC <i>,X</i> SBC <i>opr16,SP</i> SBC <i>opr8,SP</i>	Subtract with Carry	A ← (A) - (M) - (C)		-	-				IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 B2 C2 D2 E2 F2 9ED2 9EE2	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 3 5 4
SEC	Set Carry Bit	C ← 1	-	-	-	-	-	1	INH	99		1
SEI	Set Interrupt Mask Bit	I ← 1	-	-	1	-	-	-	INH	9B		1
STA <i>opr8a</i> STA <i>opr16a</i> STA <i>opr16,X</i> STA <i>opr8,X</i> STA <i>,X</i> STA <i>opr16,SP</i> STA <i>opr8,SP</i>	Store Accumulator in Memory	M ← (A)	0	-	-			-	DIR EXT IX2 IX1 IX SP2 SP1	B7 C7 D7 E7 F7 9ED7 9EE7	dd hh ll ee ff ff ff ff ff	3 4 4 3 2 5 4
STHX <i>opr8a</i> STHX <i>opr16a</i> STHX <i>opr8,SP</i>	Store H:X (Index Reg.)	(M:M + 0x0001) ← (H:X)	0	-	-			-	DIR EXT SP1	35 96 9EFF	dd hh ll ff	4 5 5
STOP	Enable Interrupts: Stop Processing Refer to MCU Documentation	I bit ← 0; Stop Processing	-	-	0	-	-	-	INH	8E		2+
STX <i>opr8a</i> STX <i>opr16a</i> STX <i>opr16,X</i> STX <i>opr8,X</i> STX <i>,X</i> STX <i>opr16,SP</i> STX <i>opr8,SP</i>	Store X (Low 8 Bits of Index Register) in Memory	M ← (X)	0	-	-			-	DIR EXT IX2 IX1 IX SP2 SP1	BF CF DF EF FF 9EDF 9EEF	dd hh ll ee ff ff ff ff ff	3 4 4 3 2 5 4
SUB # <i>opr8i</i> SUB <i>opr8a</i> SUB <i>opr16a</i> SUB <i>opr16,X</i> SUB <i>opr8,X</i> SUB <i>,X</i> SUB <i>opr16,SP</i> SUB <i>opr8,SP</i>	Subtract	A ← (A) - (M)		-	-				IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 B0 C0 D0 E0 F0 9ED0 9EE0	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 3 5 4
SWI	Software Interrupt	PC ← (PC) + 0x0001 Push (PCL); SP ← (SP) - 0x0001 Push (PCH); SP ← (SP) - 0x0001 Push (X); SP ← (SP) - 0x0001 Push (A); SP ← (SP) - 0x0001 Push (CCR); SP ← (SP) - 0x0001 I ← 1; PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	-	-	1	-	-	-	INH	83		11

8.5 Functional Description

The CMT module consists of a carrier generator, a modulator, a transmitter output, and control registers. The block diagram is shown in Figure 8-2. When operating in time mode, the user independently defines the high and low times of the carrier signal to determine both period and duty cycle. The carrier generator resolution is 125 ns when operating with an 8 MHz internal bus frequency and the CMTDIV1 and CMTDIV0 bits in the CMTMSC register are both equal to 0. The carrier generator can generate signals with periods between 250 ns (4 MHz) and 127.5 μ s (7.84 kHz) in steps of 125 ns. See Table 8-1.

Table 8-1. Clock Divide

Bus Clock (MHz)	CMTDIV1:CMTDIV0	Carrier Generator Resolution (μ s)	Min Carrier Generator Period (μ s)	Min Modulator Period (μ s)
8	0:0	0.125	0.25	1.0
8	0:1	0.25	0.5	2.0
8	1:0	0.5	1.0	4.0
8	1:1	1.0	2.0	8.0

The possible duty cycle options will depend upon the number of counts required to complete the carrier period. For example, a 1.6 MHz signal has a period of 625 ns and will therefore require 5×125 ns counts to generate. These counts may be split between high and low times, so the duty cycles available will be 20 percent (one high, four low), 40 percent (two high, three low), 60 percent (three high, two low) and 80 percent (four high, one low).

For lower frequency signals with larger periods, higher resolution (as a percentage of the total period) duty cycles are possible.

When the BASE bit in the CMT modulator status and control register (CMTMSC) is set, the carrier output (f_{CG}) to the modulator is held high continuously to allow for the generation of baseband protocols.

A third mode allows the carrier generator to alternate between two sets of high and low times. When operating in FSK mode, the generator will toggle between the two sets when instructed by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention.

The modulator provides a simple method to control protocol timing. The modulator has a minimum resolution of 1.0 μ s with an 8 MHz internal bus clock. It can count bus clocks (to provide real-time control) or it can count carrier clocks (for self-clocked protocols). See Section 8.5.2, "Modulator," for more details.

The transmitter output block controls the state of the infrared out pin (IRO). The modulator output is gated on to the IRO pin when the modulator/carrier generator is enabled.

A summary of the possible modes is shown in Table 8-2.

9.4.1 KBI x Status and Control Register (KBIXSC)

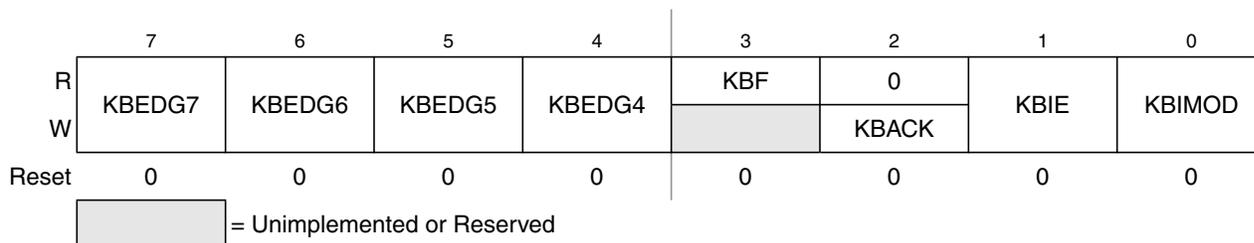


Figure 9-3. KBI x Status and Control Register (KBIXSC)

Table 9-1. KBIXSC Field Descriptions

Field	Description
7:4 KBEDG[7:4]	Keyboard Edge Select for KBI Port Bits — Each of these read/write bits selects the polarity of the edges and/or levels that are recognized as trigger events on the corresponding KBI port pin when it is configured as a keyboard interrupt input (KBIPEn = 1). Also see the KBIMOD control bit, which determines whether the pin is sensitive to edges-only or edges and levels. 0 Falling edges/low levels. 1 Rising edges/high levels.
3 KBF	Keyboard Interrupt Flag — This read-only status flag is set whenever the selected edge event has been detected on any of the enabled KBI port pins. This flag is cleared by writing a logic 1 to the KBACK control bit. The flag will remain set if KBIMOD = 1 to select edge-and-level operation and any enabled KBI port pin remains at the asserted level. 0 No KBI interrupt pending. 1 KBI interrupt pending. KBF can be used as a software pollable flag (KBIE = 0) or it can generate a hardware interrupt request to the CPU (KBIE = 1). KBF must be cleared before entering stop mode.
2 KBACK	Keyboard Interrupt Acknowledge — This write-only bit (reads always return 0) is used to clear the KBF status flag by writing a logic 1 to KBACK. When KBIMOD = 1 to select edge-and-level operation and any enabled KBI port pin remains at the asserted level, KBF is being continuously set so writing 1 to KBACK does not clear the KBF flag.
1 KBIE	Keyboard Interrupt Enable — This read/write control bit determines whether hardware interrupts are generated when the KBF status flag equals 1. When KBIE = 0, no hardware interrupts are generated, but KBF can still be used for software polling. 0 KBF does not generate hardware interrupts (use polling). 1 KBI hardware interrupt requested when KBF = 1.
0 KBIMOD	Keyboard Detection Mode — This read/write control bit selects either edge-only detection or edge-and-level detection. KBI port bits 3 through 0 can detect falling edges-only or falling edges and low levels. KBI port bits 7 through 4 can be configured to detect either: <ul style="list-style-type: none"> • Rising edges-only or rising edges and high levels (KBEDGn = 1) • Falling edges-only or falling edges and low levels (KBEDGn = 0) 0 Edge-only detection. 1 Edge-and-level detection.

10.3 TPM Block Diagram

The TPM uses one input/output (I/O) pin per channel, TPM1CH_n where *n* is the channel number (for example, 0–4). The TPM shares its I/O pins with general-purpose I/O port pins (refer to the Pins and Connections chapter for more information). Figure 10-2 shows the structure of a TPM. Some MCUs include more than one TPM, with various numbers of channels.

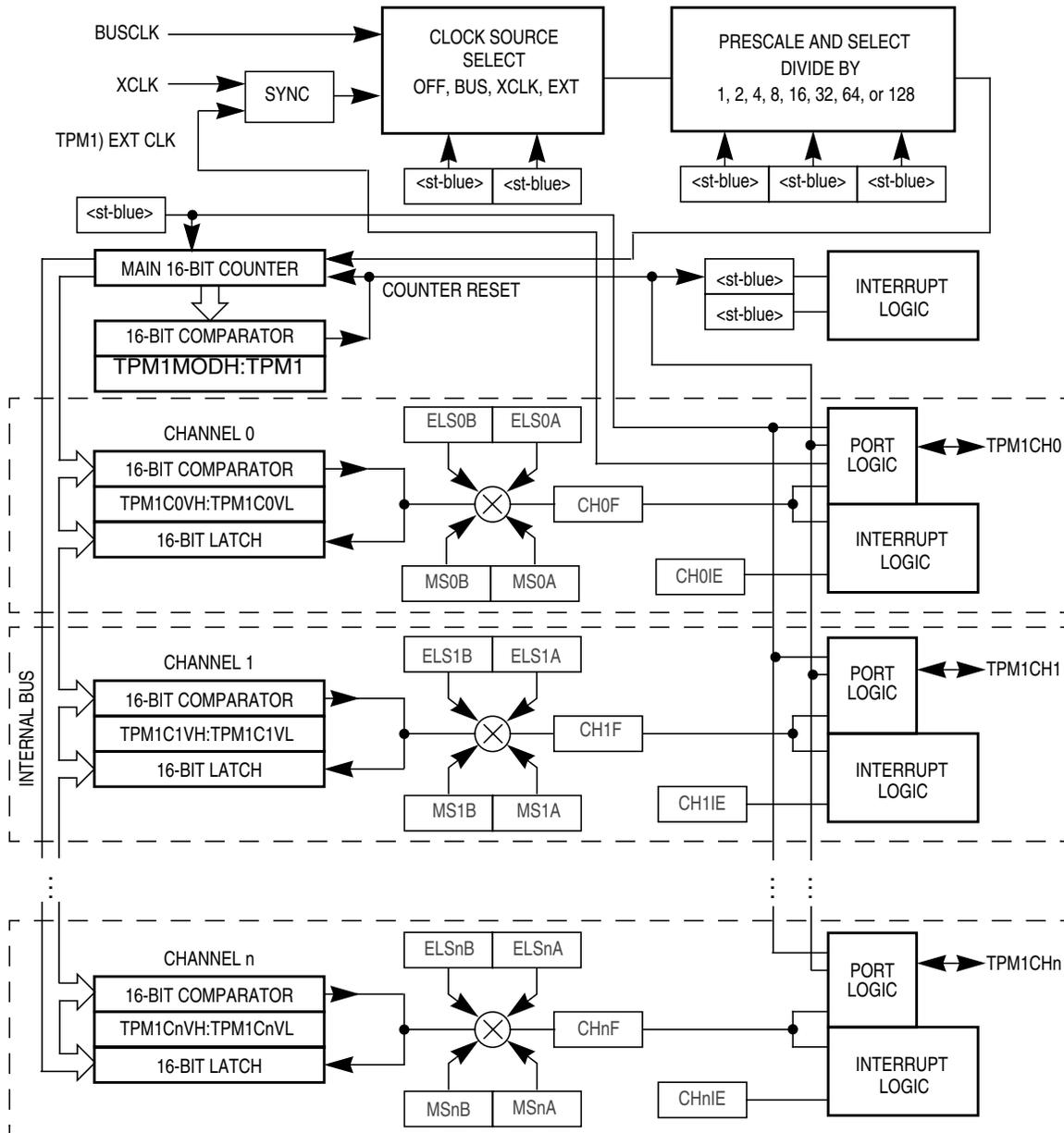


Figure 10-2. TPM Block Diagram

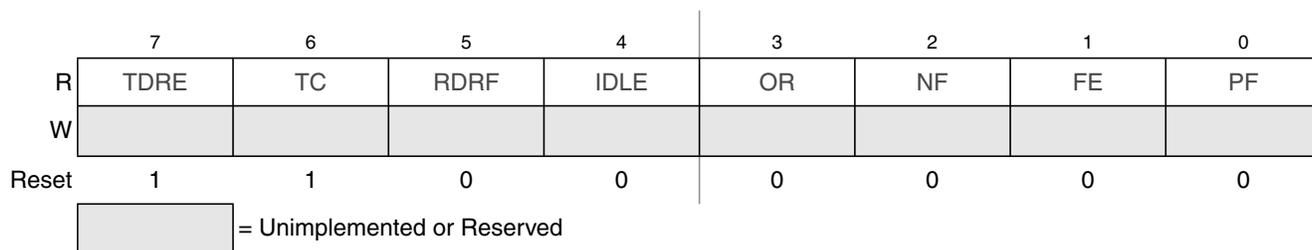
The central component of the TPM is the 16-bit counter that can operate as a free-running counter, a modulo counter, or an up-/down-counter when the TPM is configured for center-aligned PWM. The TPM counter (when operating in normal up-counting mode) provides the timing reference for the input capture,

Table 12-4. SCI1C2 Register Field Descriptions (continued)

Field	Description
1 RWU	Receiver Wakeup Control — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wakeup condition. The wakeup condition is either an idle line between messages (WAKE = 0, idle-line wakeup), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wakeup). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Section 12.3.3.2, “Receiver Wakeup Operation,” for more details. 0 Normal SCI receiver operation. 1 SCI receiver in standby waiting for wakeup condition.
0 SBK	Send Break — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Section 12.3.2.1, “Send Break and Queued Idle,” for more details. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

12.2.4 SCI Status Register 1 (SCI1S1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (which do not involve writing to this register) are used to clear these status flags.


Figure 12-7. SCI Status Register 1 (SCI1S1)
Table 12-5. SCI1S1 Register Field Descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag — TDRE is set immediately after reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCI1S1 with TDRE = 1 and then write to the SCI data register (SCI1D). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	Transmission Complete Flag — TC is set immediately after reset and when TDRE = 1 and no data, preamble, or break character is being transmitted. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete). TC is cleared automatically by reading SCI1S1 with TC = 1 and then doing one of the following three things: <ul style="list-style-type: none"> • Write to the SCI data register (SCI1D) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to SBK in SCI1C2

13.1 Features

Features of the SPI module include:

- Master or slave mode operation
- Full-duplex or single-wire bidirectional option
- Programmable transmit bit rate
- Double-buffered transmit and receive
- Serial clock phase and polarity options
- Slave select output
- Selectable MSB-first or LSB-first shifting

13.2 Block Diagrams

This section includes block diagrams showing SPI system connections, the internal organization of the SPI module, and the SPI clock dividers that control the master mode bit rate.

13.2.1 SPI System Block Diagram

Figure 13-2 shows the SPI modules of two MCUs connected in a master-slave arrangement. The master device initiates all SPI data transfers. During a transfer, the master shifts data out (on the MOSI1 pin) to the slave while simultaneously shifting data in (on the MISO1 pin) from the slave. The transfer effectively exchanges the data that was in the SPI shift registers of the two SPI systems. The SPSCCK1 signal is a clock output from the master and an input to the slave. The slave device must be selected by a low level on the slave select input ($\overline{SS1}$ pin). In this system, the master device has configured its $\overline{SS1}$ pin as an optional slave select output.

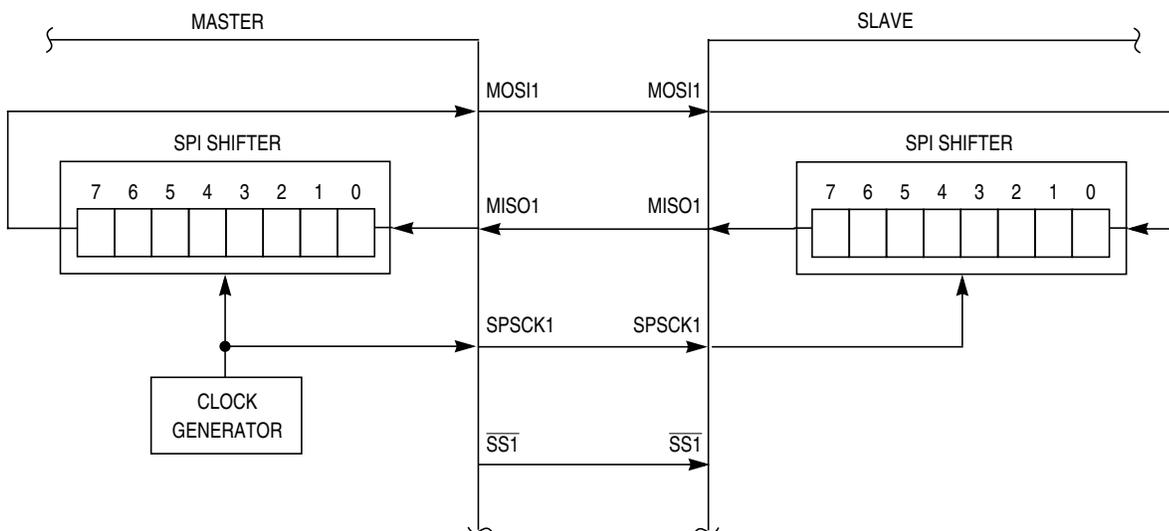


Figure 13-2. SPI System Connections

13.3.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

Figure 13-5 shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCCK edge and bit 8 ending one-half SPSCCK cycle after the sixteenth SPSCCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output pin from a master and the MISO waveform applies to the MISO output from a slave. The \overline{SS} OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master \overline{SS} output goes to active low one-half SPSCCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The \overline{SS} IN waveform applies to the slave select input of a slave.

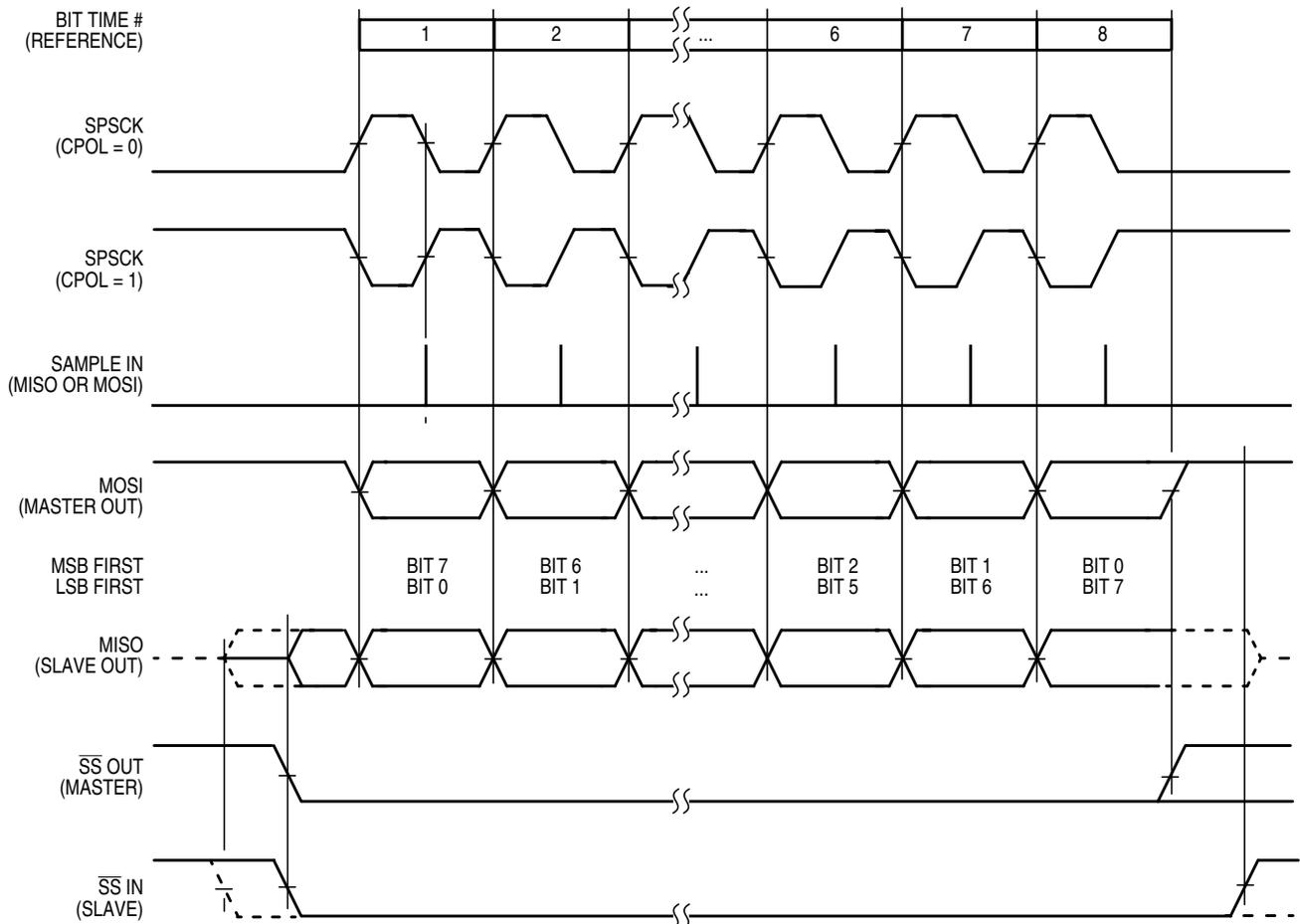


Figure 13-5. SPI Clock Formats (CPHA = 1)

When $CPHA = 0$, the slave begins to drive its MISO output with the first data bit value (MSB or LSB depending on $LSBFE$) when $\overline{SS1}$ goes to active low. The first SPSCCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the second SPSCCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When $CPHA = 0$, the slave's \overline{SS} input must go to its inactive high level between transfers.

13.3.2 SPI Pin Controls

The SPI optionally shares four port pins. The function of these pins depends on the settings of SPI control bits. When the SPI is disabled ($SPE = 0$), these four pins revert to being general-purpose port I/O pins that are not controlled by the SPI.

13.3.2.1 SPSCCK1 — SPI Serial Clock

When the SPI is enabled as a slave, this pin is the serial clock input. When the SPI is enabled as a master, this pin is the serial clock output.

13.3.2.2 MOSI1 — Master Data Out, Slave Data In

When the SPI is enabled as a master and SPI pin control zero ($SPC0$) is 0 (not bidirectional mode), this pin is the serial data output. When the SPI is enabled as a slave and $SPC0 = 0$, this pin is the serial data input. If $SPC0 = 1$ to select single-wire bidirectional mode, and master mode is selected, this pin becomes the bidirectional data I/O pin (MOMI). Also, the bidirectional mode output enable bit determines whether the pin acts as an input ($BIDIROE = 0$) or an output ($BIDIROE = 1$). If $SPC0 = 1$ and slave mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

13.3.2.3 MISO1 — Master Data In, Slave Data Out

When the SPI is enabled as a master and SPI pin control zero ($SPC0$) is 0 (not bidirectional mode), this pin is the serial data input. When the SPI is enabled as a slave and $SPC0 = 0$, this pin is the serial data output. If $SPC0 = 1$ to select single-wire bidirectional mode, and slave mode is selected, this pin becomes the bidirectional data I/O pin (SISO) and the bidirectional mode output enable bit determines whether the pin acts as an input ($BIDIROE = 0$) or an output ($BIDIROE = 1$). If $SPC0 = 1$ and master mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

13.3.2.4 $\overline{SS1}$ — Slave Select

When the SPI is enabled as a slave, this pin is the low-true slave select input. When the SPI is enabled as a master and mode fault enable is off ($MODFEN = 0$), this pin is not used by the SPI and reverts to being a general-purpose port I/O pin. When the SPI is enabled as a master and $MODFEN = 1$, the slave select output enable bit determines whether this pin acts as the mode fault input ($SSOE = 0$) or as the slave select output ($SSOE = 1$).

Figure 15-4 shows the host receiving a logic 0 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 start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

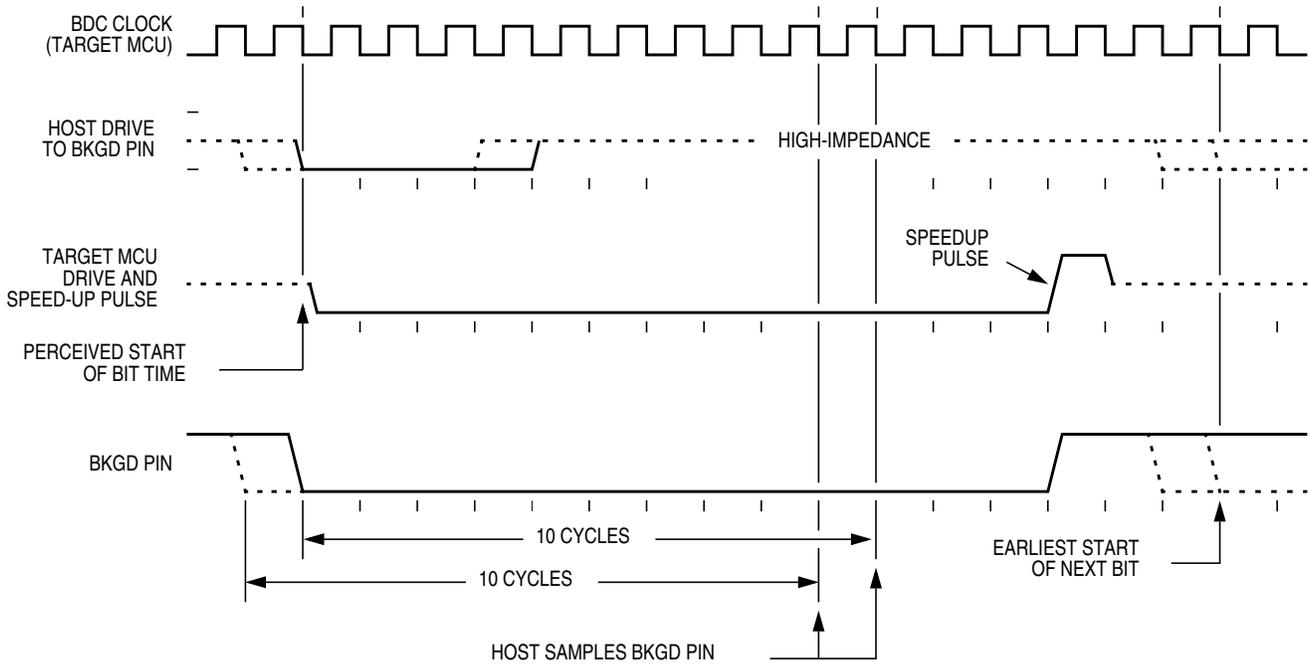


Figure 15-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

15.4.3.5 Debug FIFO High Register (DBGFH)

This register provides read-only access to the high-order eight bits of the FIFO. Writes to this register have no meaning or effect. In the event-only trigger modes, the FIFO only stores data into the low-order byte of each FIFO word, so this register is not used and will read 0x00.

Reading DBGFH does not cause the FIFO to shift to the next word. When reading 16-bit words out of the FIFO, read DBGFH before reading DBGFL because reading DBGFL causes the FIFO to advance to the next word of information.

15.4.3.6 Debug FIFO Low Register (DBGFL)

This register provides read-only access to the low-order eight bits of the FIFO. Writes to this register have no meaning or effect.

Reading DBGFL causes the FIFO to shift to the next available word of information. When the debug module is operating in event-only modes, only 8-bit data is stored into the FIFO (high-order half of each FIFO word is unused). When reading 8-bit words out of the FIFO, simply read DBGFL repeatedly to get successive bytes of data from the FIFO. It isn't necessary to read DBGFH in this case.

Do not attempt to read data from the FIFO while it is still armed (after arming but before the FIFO is filled or ARMF is cleared) because the FIFO is prevented from advancing during reads of DBGFL. This can interfere with normal sequencing of reads from the FIFO.

Reading DBGFL while the debugger is not armed causes the address of the most-recently fetched opcode to be stored to the last location in the FIFO. By reading DBGFH then DBGFL periodically, external host software can develop a profile of program execution. After eight reads from the FIFO, the ninth read will return the information that was stored as a result of the first read. To use the profiling feature, read the FIFO eight times without using the data to prime the sequence and then begin using the data to get a delayed picture of what addresses were being executed. The information stored into the FIFO on reads of DBGFL (while the FIFO is not armed) is the address of the most-recently fetched opcode.

15.4.3.9 Debug Status Register (DBGS)

This is a read-only status register.

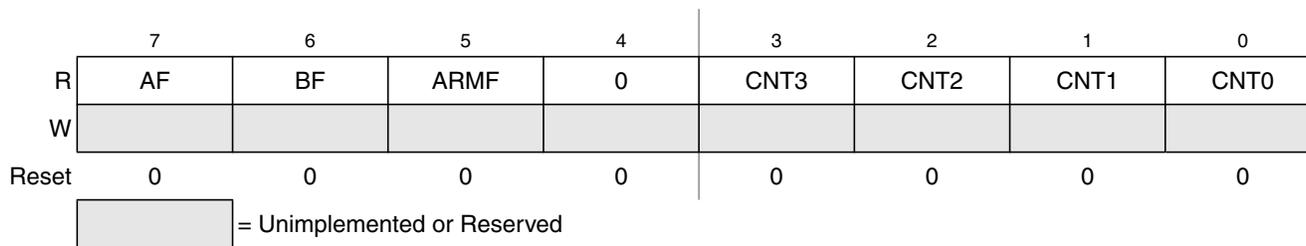


Figure 15-9. Debug Status Register (DBGS)

Table 15-6. DBGS Register Field Descriptions

Field	Description
7 AF	Trigger Match A Flag — 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	Trigger Match B Flag — 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	Arm Flag — 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]	FIFO Valid Count — 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

A.10 FLASH Specifications

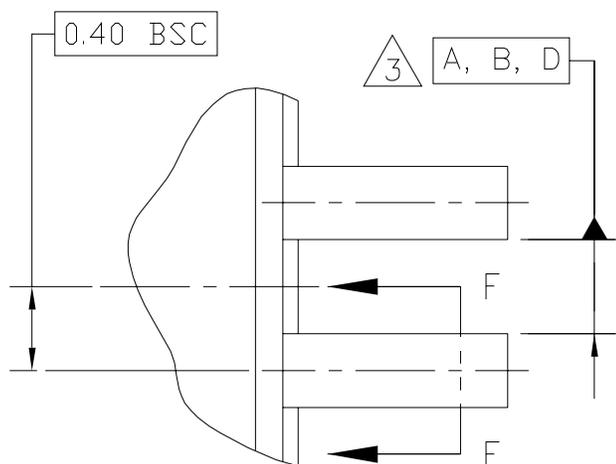
This section provides details about program/erase times and program-erase endurance for the FLASH memory.

Program and erase operations do not require any special power sources other than the normal V_{DD} supply. For more detailed information about program/erase operations, see the *Memory* chapter.

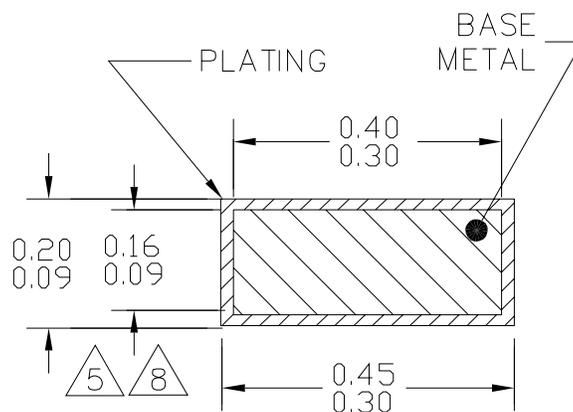
Table A-12. FLASH Characteristics

Characteristic	Symbol	Min	Typical	Max	Unit
Supply voltage for program/erase	$V_{\text{prog/erase}}$	2.05		3.6	V
Supply voltage for read operation $0 < f_{\text{BUS}} < 8 \text{ MHz}$	V_{Read}	1.8		3.6	V
Internal FCLK frequency ⁽¹⁾	f_{FCLK}	150		200	kHz
Internal FCLK period (1/FCLK)	t_{FcyC}	5		6.67	μs
Byte program time (random location) ⁽²⁾	t_{prog}		9		t_{FcyC}
Byte program time (burst mode) ⁽²⁾	t_{Burst}		4		t_{FcyC}
Page erase time ⁽²⁾	t_{Page}		4000		t_{FcyC}
Mass erase time ⁽²⁾	t_{Mass}		20,000		t_{FcyC}
Program/erase endurance ⁽³⁾ T_L to $T_H = -40^\circ\text{C}$ to $+85^\circ\text{C}$ $T = 25^\circ\text{C}$		10,000	100,000	—	cycles
Data retention ⁽⁴⁾	$t_{\text{D_ret}}$	15	100	—	years

1. The frequency of this clock is controlled by a software setting.
2. These values are hardware state machine controlled. User code does not need to count cycles. This information supplied for calculating approximate time to program and erase.
3. **Typical endurance for FLASH** was evaluated for this product family on the 9S12Dx64. For additional information on how Freescale Semiconductor defines typical endurance, please refer to Engineering Bulletin EB619/D, *Typical Endurance for Nonvolatile Memory*.
4. **Typical data retention** values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale Semiconductor defines typical data retention, please refer to Engineering Bulletin EB618/D, *Typical Data Retention for Nonvolatile Memory*.

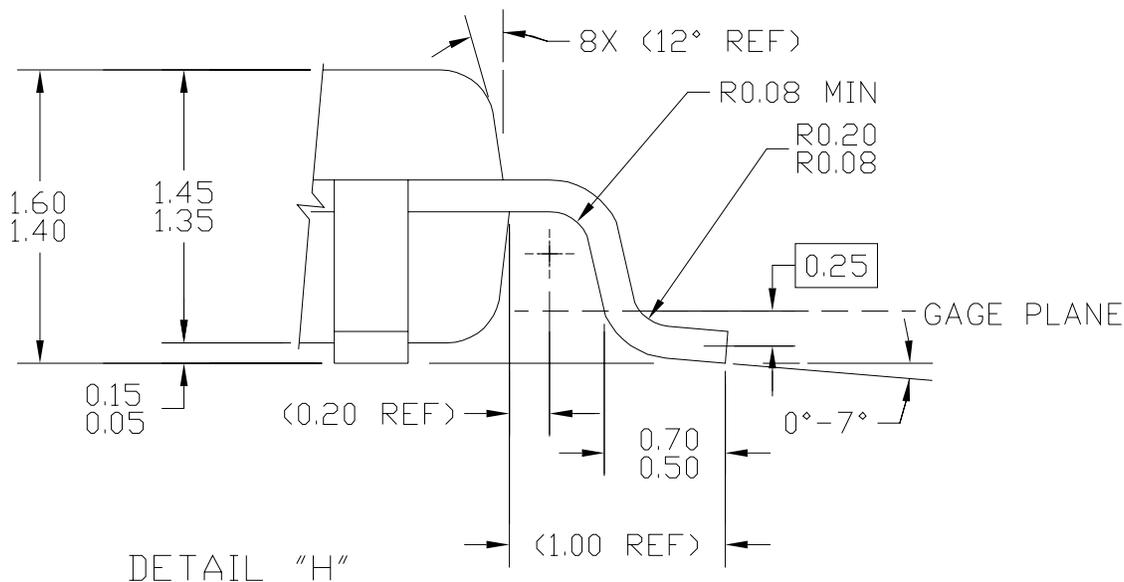


DETAIL G



M 0.2 M C A-B D

SECTION F-F
 ROTATED 90°CW
 32 PLACES



DETAIL "H"

TITLE:
 LOW PROFILE QUAD FLAT PACK (LQFP)
 32 LEAD, 0.8 PITCH (7 X 7 X 1.4)

CASE NUMBER: 873A-04
 STANDARD: JEDEC MS-026 BBA
 PACKAGE CODE: 6300 SHEET: 2 OF 3