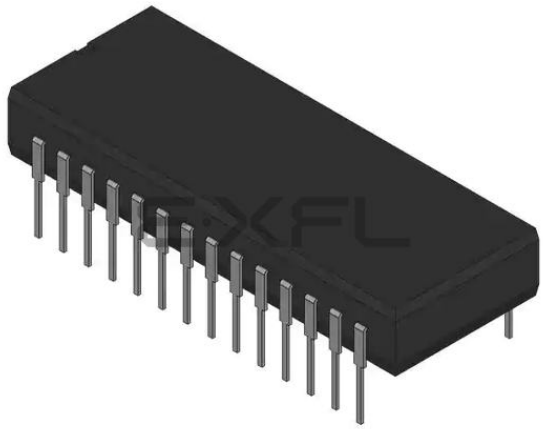


Welcome to E-XFL.COM

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	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08rd60cpe

Figure 4-2. Reset and Interrupt Vectors

Vector Number	Address (High/Low)	Vector	Vector Name
16 through 31	$FFC0:FFC1 ⇕ $FFDE:FFDF	Unused Vector Space (available for user program)	
15	$FFE0:FFE1	SPI[1]	Vspi1
14	$FFE2:FFE3	RTI	Vrti
13	$FFE4:FFE5	KBI2	Vkeyboard2
12	$FFE6:FFE7	KBI1	Vkeyboard1
11	$FFE8:FFE9	ACMP[2]	Vacmp1
10	$FFEA:FFEB	CMT	Vcmt
9	$FFEC:FFED	SCI Transmit[3]	Vsci1tx
8	$FFEE:FFEF	SCI Receive[3]	Vsci1rx
7	$FFF0:FFF1	SCI Error[3]	Vsci1err
6	$FFF2:FFF3	TPM Overflow	Vtpm1ovf
5	$FFF4:FFF5	TPM Channel 1	Vtpm1ch1
4	$FFF6:FFF7	TPM Channel 0	Vtpm1ch0
3	$FFF8:FFF9	IRQ	Virq
2	$FFFA:FFFB	Low Voltage Detect	Vlvd
1	$FFFC:FFFD	SWI	Vswi
0	$FFFE:FFFF	Reset	Vreset

1. The SPI module is not included on the MC9S08RC/RD/RE devices. This vector location is unused for those devices.
2. The analog comparator (ACMP) module is not included on the MC9S08RD devices. This vector location is unused for those devices.
3. The SCI module is not included on the MC9S08RC devices. This vector location is unused for those devices.

4.4.1 Features

Features of the FLASH memory include:

- FLASH Size
  - MC9S08RC/RD/RE/RG60 — 63374 bytes (124 pages of 512 bytes each)
  - MC9S08RC/RD/RE/RG32 — 32768 bytes (64 pages of 512 bytes each)
  - MC9S08RC/RD/RE16 — 16384 bytes (32 pages of 512 bytes each)
  - MC9S08RC/RD/RE8 — 8192 bytes (16 pages of 512 bytes each)
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature
- Flexible block protection
- Security feature for FLASH and RAM
- Auto power-down for low-frequency read accesses

4.4.2 Program and Erase Times

Before any program or erase command can be accepted, the FLASH clock divider register (FCDIV) must be written to set the internal clock for the FLASH module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz (see Section 4.6.1, “FLASH Clock Divider Register (FCDIV)”). This register can be written only once, so normally this write is done during reset initialization. FCDIV cannot be written if the access error flag, FACCERR in FSTAT, is set. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program and erase pulses. An integer number of these timing pulses are used by the command processor to complete a program or erase command.

Table 4-4 shows program and erase times. The bus clock frequency and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5$ µs. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 4-4. Program and Erase Times

Parameter	Cycles of FCLK	Time if FCLK = 200 kHz
Byte program	9	45 µs
Byte program (burst)	4	20 µs(1)
Page erase	4000	20 ms
Mass erase	20,000	100 ms

1. Excluding start/end overhead

4.4.3 Program and Erase Command Execution

The steps for executing any of the commands are listed below. The FCDIV register must be initialized and any error flags cleared before beginning command execution. The command execution steps are:

1. Write a data value to an address in the FLASH array. The address and data information from this write is latched into the FLASH interface. This write is a required first step in any command sequence. For erase and blank check commands, the value of the data is not important. For page erase commands, the address may be any address in the 512-byte page of FLASH to be erased. For mass erase and blank check commands, the address can be any address in the FLASH memory. Whole pages of 512 bytes are the smallest block of FLASH that may be erased. In the 60K version, there are two instances where the size of a block that is accessible to the user is less than 512 bytes: the first page following RAM, and the first page following the high page registers. These pages are overlapped by the RAM and high page registers respectively.

   NOTE

   Do not program any byte in the FLASH more than once after a successful erase operation. Reprogramming bits to a byte which is already programmed is not allowed without first erasing the page in which the byte resides or mass erasing the entire FLASH memory. Programming without first erasing may disturb data stored in the FLASH.

2. Write the command code for the desired command to FCMD. The five valid commands are blank check ($05), byte program ($20), burst program ($25), page erase ($40), and mass erase ($41). The command code is latched into the command buffer.
3. Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

A partial command sequence can be aborted manually by writing a 0 to FCBEF any time after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag, which must be cleared before starting a new command.

A strictly monitored procedure must be adhered to, or the command will not be accepted. This minimizes the possibility of any unintended changes to the FLASH memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. Figure 4-3 is a flowchart for executing all of the commands except for burst programming. The FCDIV register must be initialized before using any FLASH commands. This must be done only once following a reset.

5.6.1 Power-On Reset Operation

When power is initially applied to the MCU, or when the supply voltage drops below the V_{POR} level, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold the chip in reset until the supply has risen above the V_{LVD} level. Both the POR bit and the LVD bit in SRS are set following a POR.

5.6.2 LVD Reset Operation

The LVD can be configured to generate a reset upon detection of a low voltage condition. This is done by setting LVDRE to 1. LVDRE is a write-once bit that is set following a POR and is unaffected by other resets. When LVDRE = 1, setting the SAFE bit has no effect. After an LVD reset has occurred, the LVD system will hold the MCU in reset until the supply voltage is above the V_{LVD} level. The LVD bit in the SRS register is set following either an LVD reset or POR.

5.6.3 LVD Interrupt and Safe State Operation

When the voltage on the supply pin V_{DD} drops below V_{LVD} and the LVD circuit is configured for interrupt operation (LVDIE is set and LVDRE is clear), an LVD interrupt will occur. The LVD trip point is set above the minimum voltage at which the MCU can reliably operate, but the supply voltage may still be dropping. It is recommended that the user place the MCU in the safe state as soon as possible following a LVD interrupt. For systems where the supply voltage may drop so rapidly that the MCU may not have time to service the LVD interrupt and enter the safe state, it is recommended that the LVD be configured to generate a reset. The safe state is entered by executing a STOP instruction with the SAFE bit in the system power management status and control 1 (SPMSC1) register set while in a low voltage condition (LVDF = 1).

After the LVD interrupt has occurred, the user may configure the system to block all interrupts, resets, or wakeups by writing a 1 to the SAFE bit. While SAFE =1 and V_{DD} is below V_{REARM} all interrupts, resets, and wakeups are blocked. After V_{DD} is above V_{REARM}, the SAFE bit is ignored (the SAFE bit will still read a 1). After setting the SAFE bit, the MCU must be put into either the stop3 or stop2 mode before the supply voltage drops below the minimum operating voltage of the MCU. The supply voltage may now drop to a level just above the POR trip point and then restored to a level above V_{REARM} and the MCU state (in the case of stop3) and RAM contents will be preserved. When the supply voltage has been restored, interrupts, resets, and wakeups are then unblocked. When the MCU has recovered from stop mode, the SAFE bit should be cleared.

5.6.4 Low-Voltage Warning (LVW)

The LVD system has a low-voltage warning flag to indicate to the user that the supply voltage is approaching, but is still above, the low-voltage detect voltage. The LVW does not have an interrupt associated with it. However, the FLASH memory cannot be reliably programmed or erased below the V_{LVW} level, so the status of the LVWF bit in the system power management status and control 2 (SPMSC2) register must be checked before initiating any FLASH program or erase operation.

5.8.5 System Device Identification Register (SDIDH, SDIDL)

This read-only register is included so host development systems can identify the HCS08 derivative and revision number. This allows the development software to recognize where specific memory blocks, registers, and control bits are located in a target MCU.

	7	6	5	4	3	2	1	0
R	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
W								
Reset	0(1)	0(1)	0(1)	0(1)	0	0	0	0

= Unimplemented or Reserved

Figure 5-6. System Device Identification Register — High (SDIDH)

1. The revision number that is hard coded into these bits reflects the current silicon revision level.

Table 5-6. SDIDH Field Descriptions

Field	Description
7:4 REV[3:0]	**Revision Number** — The high-order 4 bits of address $1806 are hard coded to reflect the current mask set revision number (0–F).
3:0 ID[11:8]	**Part Identification Number** — Each derivative in the HCS08 Family has a unique identification number. The MC9S08RC/RD/RE/RG32/60 is hard coded to the value $004 and the MC9S08RC/RD/RE8/16 is hard coded to the value $003.

	7	6	5	4	3	2	1	0
R	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
W								
Reset, 8/16K:	0	0	0	0	0	1	1	1
Reset, 32/60K:	0	0	0	0	0	1	0	0

= Unimplemented or Reserved

Figure 5-7. System Device Identification Register — Low (SDIDL)

Table 5-7. SDIDL Field Descriptions

Field	Description
7:0 ID[7:0]	**Part Identification Number** — Each derivative in the HCS08 Family has a unique identification number. The MC9S08RC/RD/RE/RG32/60 is hard coded to the value $004 and the MC9S08RC/RD/RE8/16 is hard coded to the value $003.

6.6.3 Port C Registers (PTCD, PTCPE, and PTCDD)

Port C pins used as general-purpose I/O pins are controlled by the port C data (PTCD), data direction (PTCDD), and pullup enable (PTCPE) registers.

	7	6	5	4	3	2	1	0
R W	PTCD7	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
Reset	0	0	0	0	0	0	0	0

Figure 6-12. Port C Data Register (PTCD)

Table 6-7. PTCD Field Descriptions

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

	7	6	5	4	3	2	1	0
R W	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
Reset	0	0	0	0	0	0	0	0

Figure 6-13. Pullup Enable for Port C (PTCPE)

Table 6-8. PTCPE Field Descriptions

Field	Description
7:0 PTCPE[7:0]	**Pullup Enable for Port C Bits** — For port C pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled. For port C pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled. 0 Internal pullup device disabled. 1 Internal pullup device enabled.

0	=	Bit forced to 0
1	=	Bit forced to 1
	=	Bit set or cleared according to results of operation
U	=	Undefined after the operation

Machine coding notation

dd	=	Low-order 8 bits of a direct address 0x0000–0x00FF (high byte assumed to be 0x00)
ee	=	Upper 8 bits of 16-bit offset
ff	=	Lower 8 bits of 16-bit offset or 8-bit offset
ii	=	One byte of immediate data
jj	=	High-order byte of a 16-bit immediate data value
kk	=	Low-order byte of a 16-bit immediate data value
hh	=	High-order byte of 16-bit extended address
ll	=	Low-order byte of 16-bit extended address
rr	=	Relative offset

Source form

Everything in the source forms columns, *except expressions in italic characters,* is literal information that must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic is always a literal expression. All commas, pound signs (#), parentheses, and plus signs (+) are literal characters.

n	—	Any label or expression that evaluates to a single integer in the range 0–7
opr8i	—	Any label or expression that evaluates to an 8-bit immediate value
opr16i	—	Any label or expression that evaluates to a 16-bit immediate value
opr8a	—	Any label or expression that evaluates to an 8-bit value. The instruction treats this 8-bit value as the low order 8 bits of an address in the direct page of the 64-Kbyte address space (0x00xx).
opr16a	—	Any label or expression that evaluates to a 16-bit value. The instruction treats this value as an address in the 64-Kbyte address space.
oprx8	—	Any label or expression that evaluates to an unsigned 8-bit value, used for indexed addressing
oprx16	—	Any label or expression that evaluates to a 16-bit value. Because the HCS08 has a 16-bit address bus, this can be either a signed or an unsigned value.
rel	—	Any label or expression that refers to an address that is within –128 to +127 locations from the next address after the last byte of object code for the current instruction. The assembler will calculate the 8-bit signed offset and include it in the object code for this instruction.

Address modes

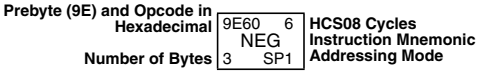
INH	=	Inherent (no operands)
IMM	=	8-bit or 16-bit immediate
DIR	=	8-bit direct
EXT	=	16-bit extended

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

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

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

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



8.5.8 Background Mode Operation

When the microcontroller is in active background mode, the CMT temporarily suspends all counting until the microcontroller returns to normal user mode.

8.6 CMT Registers and Control Bits

The following registers control and monitor CMT operation:

- CMT carrier generator data registers (CMTCGH1, CMTCGL1, CMTCGH2, CMTCGL2)
- CMT output control register (CMTOC)
- CMT modulator status and control register (CMTMSC)
- CMT modulator period data registers (CMTCMD1, CMTCMD2, CMTCMD3, CMTCMD4)

8.6.1 Carrier Generator Data Registers (CMTCGH1, CMTCGL1, CMTCGH2, and CMTCGL2)

The carrier generator data registers contain the primary and secondary high and low values for generating the carrier output.

	7	6	5	4	3	2	1	0
R W	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
Reset	u	u	u	u	u	u	u	u

u = Unaffected

Figure 8-8. Carrier Generator Data Register High 1(CMTCGH1)

Table 8-3. CMTCGH1 Field Descriptions

Field	Description
7:0 PH[7:0]	**Primary Carrier High Time Data Values** — When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see Section 8.5.2.1, “Time Mode”), this register pair is always selected. When operating in FSK mode (see Section 8.5.2.3, “FSK Mode”), this register pair and the secondary register pair are alternatively selected under control of the modulator. The primary carrier high and low time values are unaffected out of reset. These bits must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

	7	6	5	4	3	2	1	0
R W	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Reset	u	u	u	u	u	u	u	u

u = Unaffected

Figure 8-9. Carrier Generator Data Register Low 1 (CMTCGL1)

Table 8-4. CMTCGL1 Field Descriptions

Field	Description
7:0 PL[7:0]	**Primary Carrier Low Time Data Values** — When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see Section 8.5.2.1, “Time Mode”), this register pair is always selected. When operating in FSK mode (see Section 8.5.2.3, “FSK Mode”), this register pair and the secondary register pair are alternatively selected under control of the modulator. The primary carrier high and low time values are unaffected out of reset. These bits must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

	7	6	5	4	3	2	1	0
R W	SH7	SH6	SH5	SH4	SH3	SH2	SH1	SH0
Reset	u	u	u	u	u	u	u	u

u = Unaffected

Figure 8-10. Carrier Generator Data Register High 2 (CMTCGH2)

Table 8-5. CMTCGH2 Field Descriptions

Field	Description
7:0 SH[7:0]	**Secondary Carrier High Time Data Values** — When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see Section 8.5.2.1, “Time Mode"), this register pair is never selected. When operating in FSK mode (see Section 8.5.2.3, “FSK Mode"), this register pair and the primary register pair are alternatively selected under control of the modulator. The secondary carrier high and low time values are unaffected out of reset. These bits must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

10.6.4 PWM End-of-Duty-Cycle Events

For channels that are configured for PWM operation, there are two possibilities:

- When the channel is configured for edge-aligned PWM, the channel flag is set when the timer counter matches the channel value register that marks the end of the active duty cycle period.
- When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle, which are the times when the timer counter matches the channel value register.

The flag is cleared by the 2-step sequence described in Section 10.6.1, “Clearing Timer Interrupt Flags.”

10.7 TPM Registers and Control Bits

The TPM includes:

- An 8-bit status and control register (TPM1SC)
- A 16-bit counter (TPM1CNTH:TPM1CNTL)
- A 16-bit modulo register (TPM1MODH:TPM1MODL)

Each timer channel has:

- An 8-bit status and control register (TPM1CnSC)
- A 16-bit channel value register (TPM1CnVH:TPM1CnVL)

Refer to the direct-page register summary in the Memory chapter of this data sheet for the absolute address assignments for all TPM registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

Chapter 11
Serial Communications Interface (S08SCIV1)

11.1 Introduction

The MC9S08RDxx, MC9S08RExx, and MC9S08RGxx devices include a serial communications interface (SCI) module, which is sometimes called a universal asynchronous receiver/transmitters (UART). The SCI module shares pins with PTB0 and PTB1 port pins. When the SCI is enabled, the pins are controlled by the SCI module.

Figure 11-1 is a device-level block diagram with the SCI highlighted.

13.2.3 SPI Baud Rate Generation

As shown in Figure 13-4, the clock source for the SPI baud rate generator is the bus clock. The three prescale bits (SPPR2:SPPR1:SPPR0) choose a prescale divisor of 1, 2, 3, 4, 5, 6, 7, or 8. The three rate select bits (SPR2:SPR1:SPR0) divide the output of the prescaler stage by 2, 4, 8, 16, 32, 64, 128, or 256 to get the internal SPI master mode bit-rate clock.

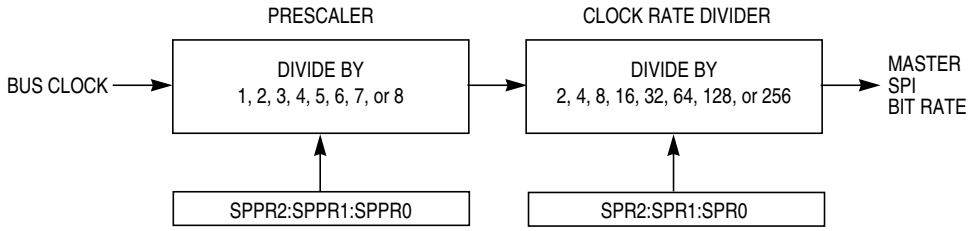


Figure 13-4. SPI Baud Rate Generation

13.3 Functional Description

An SPI transfer is initiated by checking for the SPI transmit buffer empty flag (SPTEF = 1) and then writing a byte of data to the SPI data register (SPI1D) in the master SPI device. When the SPI shift register is available, this byte of data is moved from the transmit data buffer to the shifter, SPTEF is set to indicate there is room in the buffer to queue another transmit character if desired, and the SPI serial transfer starts.

During the SPI transfer, data is sampled (read) on the MISO1 pin at one SPSCK edge and shifted, changing the bit value on the MOSI1 pin, one-half SPSCK cycle later. After eight SPSCK cycles, the data that was in the shift register of the master has been shifted out the MOSI1 pin to the slave while eight bits of data were shifted in the MISO1 pin into the master's shift register. At the end of this transfer, the received data byte is moved from the shifter into the receive data buffer and SPRF is set to indicate the data can be read by reading SPI1D. If another byte of data is waiting in the transmit buffer at the end of a transfer, it is moved into the shifter, SPTEF is set, and a new transfer is started.

Normally, SPI data is transferred most significant bit (MSB) first. If the least significant bit first enable (LSBFE) bit is set, SPI data is shifted LSB first.

When the SPI is configured as a slave, its $\overline{SS1}$ pin must be driven low before a transfer starts and $\overline{SS1}$ must stay low throughout the transfer. If a clock format where CPHA = 0 is selected, $\overline{SS1}$ must be driven to a logic 1 between successive transfers. If CPHA = 1, $\overline{SS1}$ may remain low between successive transfers. See Section 13.3.1, "SPI Clock Formats," for more details.

Because the transmitter and receiver are double buffered, a second byte, in addition to the byte currently being shifted out, can be queued into the transmit data buffer, and a previously received character can be in the receive data buffer while a new character is being shifted in. The SPTEF flag indicates when the transmit buffer has room for a new character. The SPRF flag indicates when a received character is available in the receive data buffer. The received character must be read out of the receive buffer (read SPI1D) before the next transfer is finished or a receive overrun error results.

In the case of a receive overrun, the new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for such an overrun condition so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

Table 15-1. BDC Command Summary

Command Mnemonic	Active BDM/ Non-intrusive	Coding Structure	Description
SYNC	Non-intrusive	n/a[1]	Request a timed reference pulse to determine target BDC communication speed
ACK_ENABLE	Non-intrusive	D5/d	Enable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
ACK_DISABLE	Non-intrusive	D6/d	Disable acknowledge protocol. Refer to Freescale document order no. HCS08RMv1/D.
BACKGROUND	Non-intrusive	90/d	Enter active background mode if enabled (ignore if ENBDM bit equals 0)
READ_STATUS	Non-intrusive	E4/SS	Read BDC status from BDCSCR
WRITE_CONTROL	Non-intrusive	C4/CC	Write BDC controls in BDCSCR
READ_BYTE	Non-intrusive	E0/AAAA/d/RD	Read a byte from target memory
READ_BYTE_WS	Non-intrusive	E1/AAAA/d/SS/RD	Read a byte and report status
READ_LAST	Non-intrusive	E8/SS/RD	Re-read byte from address just read and report status
WRITE_BYTE	Non-intrusive	C0/AAAA/WD/d	Write a byte to target memory
WRITE_BYTE_WS	Non-intrusive	C1/AAAA/WD/d/SS	Write a byte and report status
READ_BKPT	Non-intrusive	E2/RBKP	Read BDCBKPT breakpoint register
WRITE_BKPT	Non-intrusive	C2/WBKP	Write BDCBKPT breakpoint register
GO	Active BDM	08/d	Go to execute the user application program starting at the address currently in the PC
TRACE1	Active BDM	10/d	Trace 1 user instruction at the address in the PC, then return to active background mode
TAGGO	Active BDM	18/d	Same as GO but enable external tagging (HCS08 devices have no external tagging pin)
READ_A	Active BDM	68/d/RD	Read accumulator (A)
READ_CCR	Active BDM	69/d/RD	Read condition code register (CCR)
READ_PC	Active BDM	6B/d/RD16	Read program counter (PC)
READ_HX	Active BDM	6C/d/RD16	Read H and X register pair (H:X)
READ_SP	Active BDM	6F/d/RD16	Read stack pointer (SP)
READ_NEXT	Active BDM	70/d/RD	Increment H:X by one then read memory byte located at H:X
READ_NEXT_WS	Active BDM	71/d/SS/RD	Increment H:X by one then read memory byte located at H:X. Report status and data.
WRITE_A	Active BDM	48/WD/d	Write accumulator (A)
WRITE_CCR	Active BDM	49/WD/d	Write condition code register (CCR)
WRITE_PC	Active BDM	4B/WD16/d	Write program counter (PC)
WRITE_HX	Active BDM	4C/WD16/d	Write H and X register pair (H:X)
WRITE_SP	Active BDM	4F/WD16/d	Write stack pointer (SP)
WRITE_NEXT	Active BDM	50/WD/d	Increment H:X by one, then write memory byte located at H:X
WRITE_NEXT_WS	Active BDM	51/WD/d/SS	Increment H:X by one, then write memory byte located at H:X. Also report status.

[1] The SYNC command is a special operation that does not have a command code.

A force-type breakpoint waits for the current instruction to finish and then acts upon the breakpoint request. The usual action in response to a breakpoint is to go to active background mode rather than continuing to the next instruction in the user application program.

The tag vs. force terminology is used in two contexts within the debug module. The first context refers to breakpoint requests from the debug module to the CPU. The second refers to match signals from the comparators to the debugger control logic. When a tag-type break request is sent to the CPU, a signal is entered into the instruction queue along with the opcode so that if/when this opcode ever executes, the CPU will effectively replace the tagged opcode with a BGND opcode so the CPU goes to active background mode rather than executing the tagged instruction. When the TRGSEL control bit in the DBGT register is set to select tag-type operation, the output from comparator A or B is qualified by a block of logic in the debug module that tracks opcodes and only produces a trigger to the debugger if the opcode at the compare address is actually executed. There is separate opcode tracking logic for each comparator so more than one compare event can be tracked through the instruction queue at a time.

15.3.5 Trigger Modes

The trigger mode controls the overall behavior of a debug run. The 4-bit TRG field in the DBGT register selects one of nine trigger modes. When TRGSEL = 1 in the DBGT register, the output of the comparator must propagate through an opcode tracking circuit before triggering FIFO actions. The BEGIN bit in DBGT chooses whether the FIFO begins storing data when the qualified trigger is detected (begin trace), or the FIFO stores data in a circular fashion from the time it is armed until the qualified trigger is detected (end trigger).

A debug run is started by writing a 1 to the ARM bit in the DBGC register, which sets the ARMF flag and clears the AF and BF flags and the CNT bits in DBGS. A begin-trace debug run ends when the FIFO gets full. An end-trace run ends when the selected trigger event occurs. Any debug run can be stopped manually by writing a 0 to ARM or DBGEN in DBGC.

In all trigger modes except event-only modes, the FIFO stores change-of-flow addresses. In event-only trigger modes, the FIFO stores data in the low-order eight bits of the FIFO.

The BEGIN control bit is ignored in event-only trigger modes and all such debug runs are begin type traces. When TRGSEL = 1 to select opcode fetch triggers, it is not necessary to use R/W in comparisons because opcode tags would only apply to opcode fetches that are always read cycles. It would also be unusual to specify TRGSEL = 1 while using a full mode trigger because the opcode value is normally known at a particular address.

The following trigger mode descriptions only state the primary comparator conditions that lead to a trigger. Either comparator can usually be further qualified with R/W by setting RWAEN (RWBEN) and the corresponding RWA (RWB) value to be matched against R/W. The signal from the comparator with optional R/W qualification is used to request a CPU breakpoint if BRKEN = 1 and TAG determines whether the CPU request will be a tag request or a force request.

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

Field	Description
2 WS	**Wait or Stop Status** — When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host should issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands. 0 Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active) 1 Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode
1 WSF	**Wait or Stop Failure Status** — This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.) 0 Memory access did not conflict with a wait or stop instruction 1 Memory access command failed because the CPU entered wait or stop mode
0 DVF	**Data Valid Failure Status** — This status bit is not used in the MC9S08RC/RD/RE/RG because it does not have any slow access memory. 0 Memory access did not conflict with a slow memory access 1 Memory access command failed because CPU was not finished with a slow memory access

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 (SBDFR)

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

A.8 Oscillator Characteristics

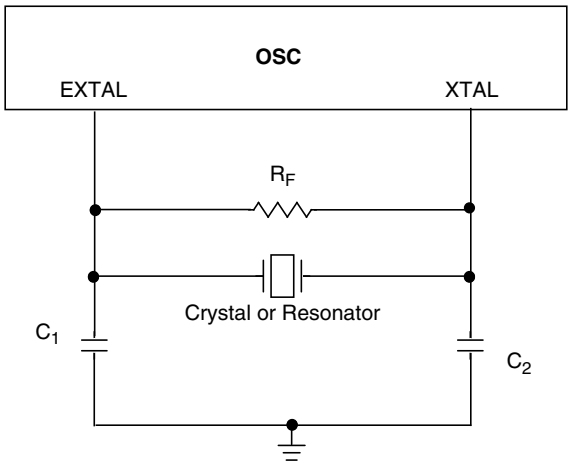


Table A-8. OSC Electrical Specifications (Temperature Range = -40 to 85°C Ambient)

Characteristic	Symbol	Min	Typ(1)	Max	Unit
Frequency	f_{OSC}	1	—	16	MHz
Load Capacitors	C_1 C_2	Note(2)			
Feedback resistor	R_F		1		MΩ

1. Data in typical column was characterized at 3.0 V, 25°C or is typical recommended value.
2. See crystal or resonator manufacturer's recommendation.

A.9 AC Characteristics

This section describes ac timing characteristics for each peripheral system.

A.9.1 Control Timing

Table A-9. Control Timing

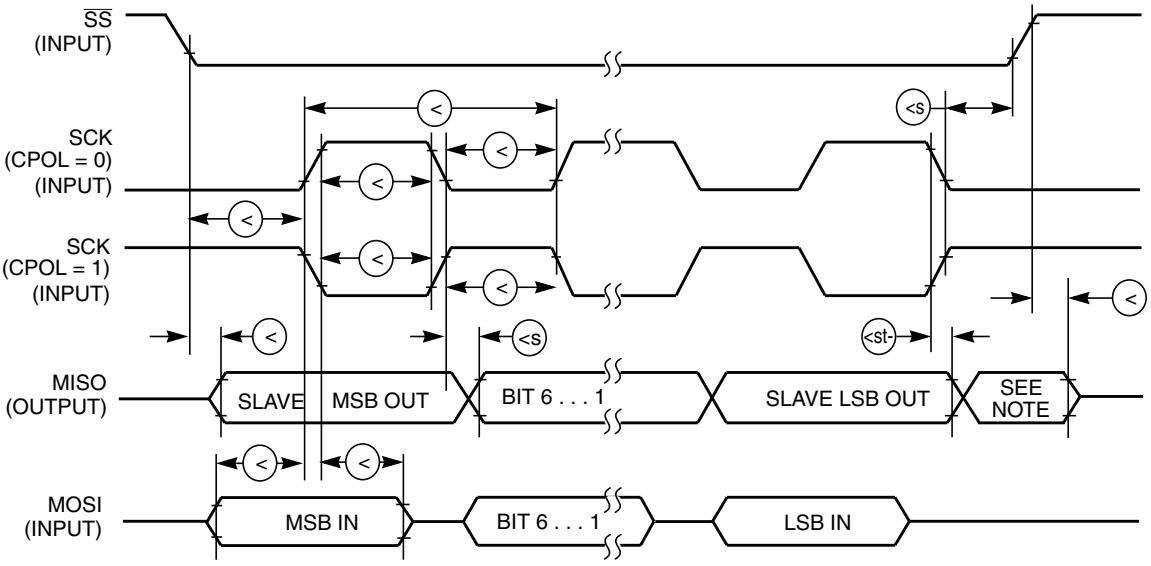
Parameter	Symbol	Min	Typical	Max	Unit
Bus frequency (t_{cyc} = 1/f_{Bus})	f_{Bus}	dc	—	8	MHz
Real time interrupt internal oscillator period	t_{RTI}	400		1600	µs
External reset pulse width(1)	t_{extrst}	1.5 t_{cyc}		—	ns
Reset low drive(2)	t_{rstdrv}	34 t_{cyc}		—	ns
Active background debug mode latch setup time	t_{MSSU}	25		—	ns
Active background debug mode latch hold time	t_{MSH}	25		—	ns
IRQ pulse width(3)	t_{ILIH}	1.5 t_{cyc}		—	ns
Port rise and fall time (load = 50 pF)(4)	t_{Rise}, t_{Fall}	—	3		ns

1. This is the shortest pulse that is guaranteed to be recognized as a reset pin request. Shorter pulses are not guaranteed to override reset requests from internal sources.

Table A-11. SPI Electrical Characteristic

Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency[3] Master Slave	 f_{op} f_{op}	 $f_{Bus}/2048$ dc	 $f_{Bus}/2$ $f_{Bus}/4$	Hz
1	Cycle time Master Slave	 t_{SCK} t_{SCK}	 2 4	 2048 —	 t_{cyc} t_{cyc}
2	Enable lead time Master Slave	 t_{Lead} t_{Lead}	 — 1/2	 1/2 —	 t_{SCK} t_{SCK}
3	Enable lag time Master Slave	 t_{Lag} t_{Lag}	 — 1/2	 1/2 —	 t_{SCK} t_{SCK}
4	Clock (SPSCK) high time Master and Slave	t_{SCKH}	1/2 t_{SCK} – 25	—	ns
5	Clock (SPSCK) low time Master and Slave	t_{SCKL}	1/2 t_{SCK} – 25	—	ns
6	Data setup time (inputs) Master Slave	 $t_{SI(M)}$ $t_{SI(S)}$	 30 30	 — —	 ns ns
7	Data hold time (inputs) Master Slave	 $t_{HI(M)}$ $t_{HI(S)}$	 30 30	 — —	 ns ns
8	Access time, slave[4]	t_A	0	40	ns
9	Disable time, slave[5]	t_{dis}	—	40	ns
10	Data setup time (outputs) Master Slave	 t_{SO} t_{SO}	 25 25	 — —	 ns ns
11	Data hold time (outputs) Master Slave	 t_{HO} t_{HO}	 –10 –10	 — —	 ns ns

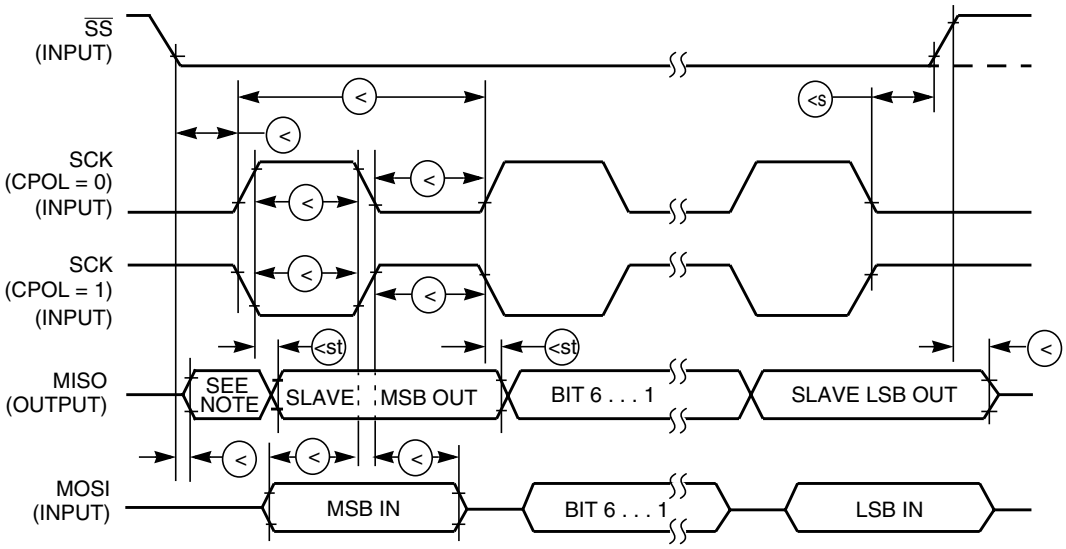
1. Refer to Figure A-11 through Figure A-14.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless noted; 100 pF load on all SPI pins. All timing assumes slew rate control disabled and high drive strength enabled for SPI output pins.
3. Maximum baud rate must be limited to 5 MHz due to pad input characteristics.
4. Time to data active from high-impedance state.
5. Hold time to high-impedance state.



NOTE:

1. Not defined but normally MSB of character just received

Figure A-13. SPI Slave Timing (CPHA = 0)



NOTE:

1. Not defined but normally LSB of character just received

Figure A-14. SPI Slave Timing (CPHA = 1)

NOTES

1. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982.

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 -L- , -M- AND -N- TO BE DETERMINED AT DATUM PLANE -H- .

5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE -T- .

6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H- .

7. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE DIMENSION TO EXCEED 0.53. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.07.