

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	25
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	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08rc8fje

Table 2-2. Signal Properties (continued)

Pin Name	Dir ⁽¹⁾	High Current Pin	Pullup ⁽²⁾	Comments ⁽³⁾
PTD2/IRQ	I/O	N	SWC ⁽⁵⁾	Available only in 32-, 44-, and 48-pin packages
PTD3	I/O	N	SWC	Available only in 44- and 48-pin packages
PTD4/ACMP1-	I/O	N	SWC	Available only in 32-, 44-, and 48-pin packages
PTD5/ACMP1+	I/O	N	SWC	Available only in 32-, 44-, and 48-pin packages
PTD6/TPM1CH0	I/O	N	SWC	
PTE0	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE1	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE2	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE3	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE4	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE5	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE6	I/O	N	SWC	Available only in 44- and 48-pin packages
PTE7	I/O	N	SWC	Available only in 44- and 48-pin packages

1. Unless otherwise indicated, all digital inputs have input hysteresis.
2. SWC is software-controlled pullup resistor, the register is associated with the respective port.
3. Not all general-purpose I/O pins are available on all packages. To avoid extra current drain from floating input pins, the user's reset initialization routine in the application program should either enable on-chip pullup devices or change the direction of unconnected pins to outputs so the pins do not float.
4. When these pins are configured as $\overline{\text{RESET}}$ or BKGD/MS pullup device is enabled.
5. When configured for the IRQ function, this pin will have a pullup device enabled when the IRQ is set for falling edge detection and a pulldown device enabled when the IRQ is set for rising edge detection.

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

Background commands are of two types:

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

The active background mode is used to program a bootloader or user application program into the FLASH program memory before the MCU is operated in run mode for the first time. When the MC9S08RC/RD/RE/RG is shipped from the Freescale Semiconductor factory, the FLASH program memory is usually erased so there is no program that could be executed in run mode until the FLASH memory is initially programmed. The active background mode can also be used to erase and reprogram the FLASH memory after it has been previously programmed.

For additional information about the active background mode, refer to the Development Support chapter.

3.5 Wait Mode

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

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

3.6 Stop Modes

One of three stop modes is entered upon execution of a STOP instruction when the STOPE bit in the system option register is set. In all stop modes, all internal clocks are halted. If the STOPE bit is not set when the CPU executes a STOP instruction, the MCU will not enter any of the stop modes and an illegal opcode reset is forced. The stop modes are selected by setting the appropriate bits in SPMSC2.

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

Table 3-1. Stop Mode Behavior

Mode	PDC	PPDC	CPU, Digital Peripherals, FLASH	RAM	OSC	ACMP	Regulator	I/O Pins	RTI
Stop1	1	0	Off	Off	Off	Standby	Standby	Reset	Off
Stop2	1	1	Off	Standby	Off	Standby	Standby	States held	Optionally on
Stop3	0	Don't care	Standby	Standby	Off	Standby	Standby	States held	Optionally on

3.6.1 Stop1 Mode

Stop1 mode provides the lowest possible standby power consumption by causing the internal circuitry of the MCU to be powered down. To enter stop1, the user must execute a STOP instruction with the PDC bit in SPMSC2 set and the PPDC bit clear. Stop1 can be entered only if the LVD reset is disabled (LVDRE = 0).

When the MCU is in stop1 mode, all internal circuits that are powered from the voltage regulator are turned off. The voltage regulator is in a low-power standby state, as are the OSC and ACMP.

Exit from stop1 is done by asserting any of the wakeup pins on the MCU: $\overline{\text{RESET}}$, IRQ, or KBI1, which have been enabled. IRQ and KBI pins are always active-low when used as wakeup pins in stop1 regardless of how they were configured before entering stop1.

Upon wakeup from stop1 mode, the MCU will start up as from a power-on reset (POR). The CPU will take the reset vector.

3.6.2 Stop2 Mode

Stop2 mode provides very low standby power consumption and maintains the contents of RAM and the current state of all of the I/O pins. To select entry into stop2 upon execution of a STOP instruction, the user must execute a STOP instruction with the PPDC and PDC bits in SPMSC2 set. Stop2 can be entered only if LVDRE = 0.

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

When the MCU is in stop2 mode, all internal circuits that are powered from the voltage regulator are turned off, except for the RAM. The voltage regulator is in a low-power standby state, as is the ACMP. Upon entry

Memory

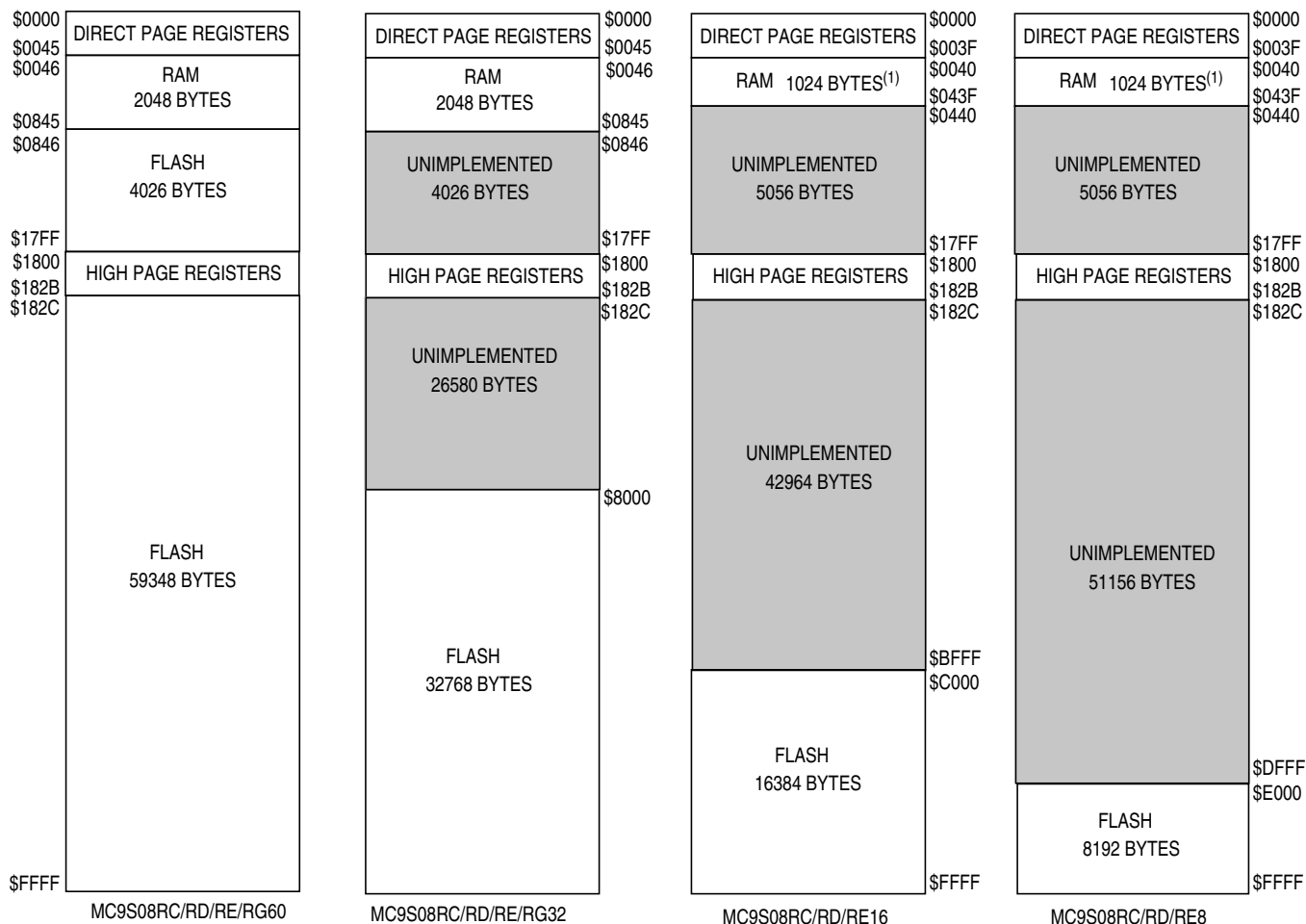


Figure 4-1. MC9S08RC/RD/RE/RG Memory Map

4.1.1 Reset and Interrupt Vector Assignments

Figure 4-2 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the Freescale-provided equate file for the MC9S08RC/RD/RE/RG. For more details about resets, interrupts, interrupt priority, and local interrupt mask controls, refer to the Chapter 5, “Resets, Interrupts, and System Configuration.”

Provided the key enable (KEYEN) bit is 1, the 8-byte comparison key can be used to temporarily disengage memory security. This key mechanism can be accessed only through user code running in secure memory. (A security key cannot be entered directly through background debug commands.) This security key can be disabled completely by programming the KEYEN bit to 0. If the security key is disabled, the only way to disengage security is by mass erasing the FLASH if needed (normally through the background debug interface) and verifying that FLASH is blank. To avoid returning to secure mode after the next reset, program the security bits (SEC01:SEC00) to the unsecured state (1:0).

4.3 RAM

The MC9S08RC/RD/RE/RG includes static RAM. The locations in RAM below \$0100 can be accessed using the more efficient direct addressing mode, and any single bit in this area can be accessed with the bit-manipulation instructions (BCLR, BSET, BRCLR, and BRSET). Locating the most frequently accessed program variables in this area of RAM is preferred.

The RAM retains data when the MCU is in low-power wait, stop2, or stop3 mode. At power-on or after wakeup from stop1, the contents of RAM are uninitialized. RAM data is unaffected by any reset provided that the supply voltage does not drop below the minimum value for RAM retention.

For compatibility with older M68HC05 MCUs, the HCS08 resets the stack pointer to \$00FF. In the MC9S08RC/RD/RE/RG, it is usually best to reinitialize the stack pointer to the top of the RAM so the direct page RAM can be used for frequently accessed RAM variables and bit-addressable program variables. Include the following 2-instruction sequence in your reset initialization routine (where RamLast is equated to the highest address of the RAM in the Freescale-provided equate file).

```
LDHX    #RamLast+1    ;point one past RAM
TXS                    ;SP<-(H:X-1)
```

When security is enabled, the RAM is considered a secure memory resource and is not accessible through BDM or through code executing from non-secure memory. See Section 4.5, “Security,” for a detailed description of the security feature.

4.4 FLASH

The FLASH memory is intended primarily for program storage. In-circuit programming allows the operating program to be loaded into the FLASH memory after final assembly of the application product. It is possible to program the entire array through the single-wire background debug interface. Because no special voltages are needed for FLASH erase and programming operations, in-application programming is also possible through other software-controlled communication paths. For a more detailed discussion of in-circuit and in-application programming, refer to the *HCS08 Family Reference Manual, Volume I*, Freescale Semiconductor document order number HCS08RMv1/D.

4.6.5 FLASH Status Register (FSTAT)

Bits 3, 1, and 0 always read 0 and writes have no meaning or effect. The remaining five bits are status bits that can be read at any time. Writes to these bits have special meanings that are discussed in the bit descriptions.

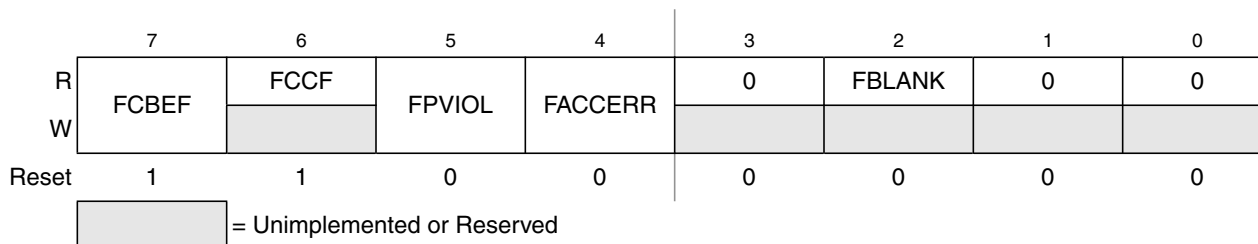


Figure 4-9. FLASH Status Register (FSTAT)

Table 4-12. FSTAT Field Descriptions

Field	Description
7 FCBEF	FLASH Command Buffer Empty Flag — The FCBEF bit is used to launch commands. It also indicates that the command buffer is empty so that a new command sequence can be executed when performing burst programming. The FCBEF bit is cleared by writing a 1 to it or when a burst program command is transferred to the array for programming. Only burst program commands can be buffered. 0 Command buffer is full (not ready for additional commands). 1 A new burst program command may be written to the command buffer.
6 FCCF	FLASH Command Complete Flag — FCCF is set automatically when the command buffer is empty and no command is being processed. FCCF is cleared automatically when a new command is started (by writing 1 to FCBEF to register a command). Writing to FCCF has no meaning or effect. 0 Command in progress 1 All commands complete
5 FPVIOL	Protection Violation Flag — 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	Access Error Flag — 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 Section 4.4.5, "Access Errors." FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error. 1 An access error has occurred.
2 FBLANK	FLASH Verified as All Blank (erased) Flag — 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 \$FF).

Chapter 5

Resets, Interrupts, and System Configuration

5.1 Introduction

This section discusses basic reset and interrupt mechanisms and the various sources of reset and interrupts in the MC9S08RC/RD/RE/RG. Some interrupt sources from peripheral modules are discussed in greater detail within other sections of this data sheet. This section gathers basic information about all reset and interrupt sources in one place for easy reference. A few reset and interrupt sources, including the computer operating properly (COP) watchdog and real-time interrupt (RTI), are not part of on-chip peripheral systems having their own sections but are part of the system control logic.

5.2 Features

Reset and interrupt features include:

- Multiple sources of reset for flexible system configuration and reliable operation:
 - Power-on detection (POR)
 - Low voltage detection (LVD) with enable
 - External reset pin with enable ($\overline{\text{RESET}}$)
 - COP watchdog with enable and two timeout choices
 - Illegal opcode
 - Illegal address (on 16K and 8K devices)
 - Serial command from a background debug host
- Reset status register (SRS) to indicate source of most recent reset; flag to indicate stop2 (partial power down) mode recovery (PPDF)
- Separate interrupt vectors for each module (reduces polling overhead) (see Table 5-1)
- Safe state for protecting the MCU in low-voltage condition

5.3 MCU Reset

Resetting the MCU provides a way to start processing from a known set of initial conditions. During reset, most control and status registers are forced to initial values and the program counter is loaded from the reset vector (\$FFFE:\$FFFF). On-chip peripheral modules are disabled and I/O pins are initially configured as general-purpose high-impedance inputs with pullup devices disabled. The I bit in the condition code register (CCR) is set to block maskable interrupts until the user program has a chance to initialize the stack pointer (SP) and system control settings. SP is forced to \$00FF at reset.

The TPM can be configured to use PTD6 as either an input capture, output compare, PWM, or external clock input pin. Refer to the Chapter 6, “Parallel Input/Output,” for more information about using PTD6 as a timer pin.

6.3.5 Port E

Port E	Bit 7	6	5	4	3	2	1	Bit 0
MCU Pin:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0

Figure 6-5. Port E Pin Names

Port E is an 8-bit general-purpose I/O port.

Port E pins are available as general-purpose I/O pins controlled by the port E data (PTED), data direction (PTEDD), and pullup enable (PTEPE) registers. Refer to Section 6.4, “Parallel I/O Controls,” for more information about general-purpose I/O control.

6.4 Parallel I/O Controls

Provided no on-chip peripheral is controlling a port pin, the pins operate as general-purpose I/O pins that are accessed and controlled by a data register (PTxD), a data direction register (PTxDD), and a pullup enable register (PTxPE) where x is A, B, C, D, or E.

Reads of the data register return the pin value (if PTxDDn = 0) or the contents of the port data register (if PTxDDn = 1). Writes to the port data register are latched into the port register whether the pin is controlled by an on-chip peripheral or the pin is configured as an input. If the corresponding pin is not controlled by a peripheral and is configured as an output, this level will be driven out the port pin.

6.4.1 Data Direction Control

The data direction control bits determine whether the pin output driver is enabled, and they control what is read for port data register reads. Each port pin has a data direction control bit. When PTxDDn = 0, the corresponding pin is an input and reads of PTxD return the pin value. When PTxDDn = 1, the corresponding pin is an output and reads of PTxD return the last value written to the port data register. When a peripheral module or system function is in control of a port pin, the data direction control still controls what is returned for reads of the port data register, even though the peripheral system has overriding control of the actual pin direction.

For the MC9S08RC/RD/RE/RG MCU, reads of PTD0/BKGD/MS and PTD1/ $\overline{\text{RESET}}$ will return the value on the output pin.

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

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.

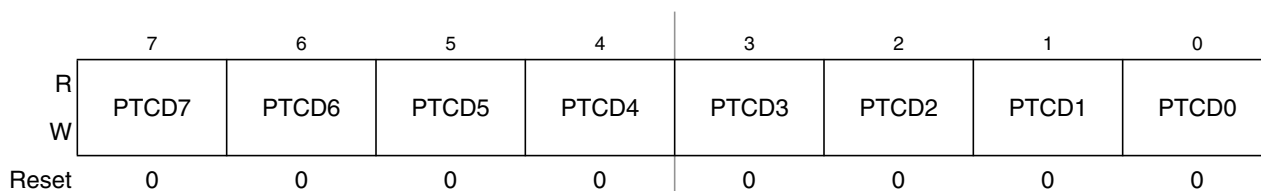


Figure 6-12. Port C Data Register (PTCD)

Table 6-7. PTCD Field Descriptions

Field	Description
7:0 PTCD[7:0]	<p>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.</p> <p>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.</p>

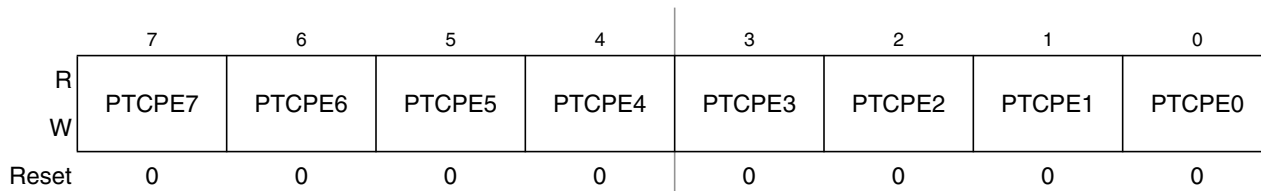


Figure 6-13. Pullup Enable for Port C (PTCPE)

Table 6-8. PTCPE Field Descriptions

Field	Description
7:0 PTCPE[7:0]	<p>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.</p> <p>0 Internal pullup device disabled. 1 Internal pullup device enabled.</p>

Chapter 7

Central Processor Unit (S08CPUV2)

7.1 Introduction

This section provides summary information about the registers, addressing modes, and instruction set of the CPU of the HCS08 Family. For a more detailed discussion, refer to the *HCS08 Family Reference Manual, volume 1*, Freescale Semiconductor document order number HCS08RMV1/D.

The HCS08 CPU is fully source- and object-code-compatible with the M68HC08 CPU. Several instructions and enhanced addressing modes were added to improve C compiler efficiency and to support a new background debug system which replaces the monitor mode of earlier M68HC08 microcontrollers (MCU).

7.1.1 Features

Features of the HCS08 CPU include:

- Object code fully upward-compatible with M68HC05 and M68HC08 Families
- All registers and memory are mapped to a single 64-Kbyte address space
- 16-bit stack pointer (any size stack anywhere in 64-Kbyte address space)
- 16-bit index register (H:X) with powerful indexed addressing modes
- 8-bit accumulator (A)
- Many instructions treat X as a second general-purpose 8-bit register
- Seven addressing modes:
 - Inherent — Operands in internal registers
 - Relative — 8-bit signed offset to branch destination
 - Immediate — Operand in next object code byte(s)
 - Direct — Operand in memory at 0x0000–0x00FF
 - Extended — Operand anywhere in 64-Kbyte address space
 - Indexed relative to H:X — Five submodes including auto increment
 - Indexed relative to SP — Improves C efficiency dramatically
- Memory-to-memory data move instructions with four address mode combinations
- Overflow, half-carry, negative, zero, and carry condition codes support conditional branching on the results of signed, unsigned, and binary-coded decimal (BCD) operations
- Efficient bit manipulation instructions
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- STOP and WAIT instructions to invoke low-power operating modes

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

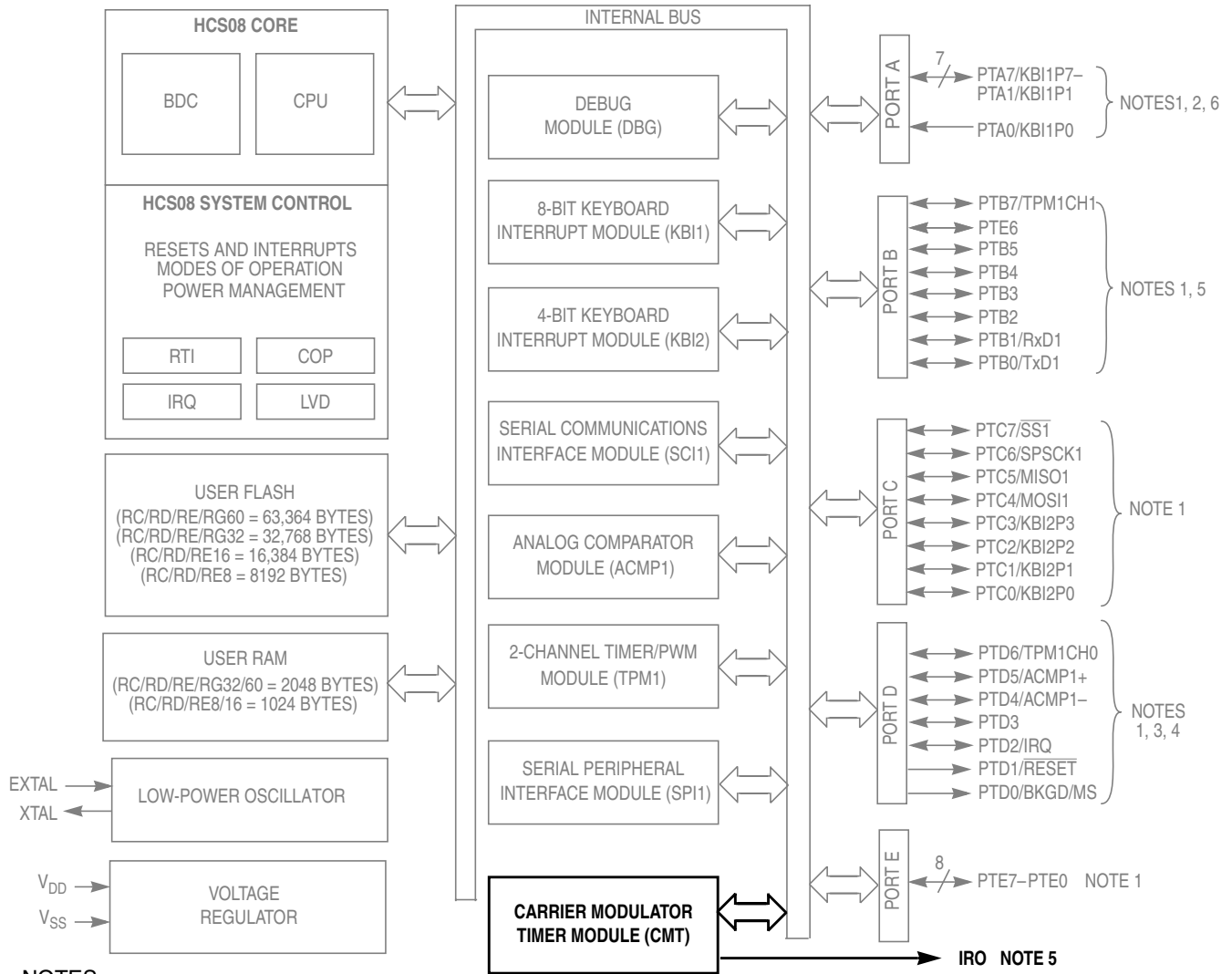
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles ¹
			V	H	I	N	Z	C				
TAP	Transfer Accumulator to CCR	$CCR \leftarrow (A)$							INH	84		1
TAX	Transfer Accumulator to X (Index Register Low)	$X \leftarrow (A)$	-	-	-	-	-	-	INH	97		1
TPA	Transfer CCR to Accumulator	$A \leftarrow (CCR)$	-	-	-	-	-	-	INH	85		1
TST <i>opr8a</i> TSTA TSTX TST <i>opr8,X</i> TST <i>,X</i> TST <i>opr8,SP</i>	Test for Negative or Zero	(M) - 0x00 (A) - 0x00 (X) - 0x00 (M) - 0x00 (M) - 0x00 (M) - 0x00	0	-	-			-	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	4 1 1 4 3 5
TSX	Transfer SP to Index Reg.	$H:X \leftarrow (SP) + 0x0001$	-	-	-	-	-	-	INH	95		2
TXA	Transfer X (Index Reg. Low) to Accumulator	$A \leftarrow (X)$	-	-	-	-	-	-	INH	9F		1
TXS	Transfer Index Reg. to SP	$SP \leftarrow (H:X) - 0x0001$	-	-	-	-	-	-	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit \leftarrow 0; Halt CPU	-	-	0	-	-	-	INH	8F		2+

¹ Bus clock frequency is one-half of the CPU clock frequency.

Chapter 8

Carrier Modulator Timer (S08CMTV1)

8.1 Introduction



NOTES:

1. Port pins are software configurable with pullup device if input port
2. PTA0 does not have a clamp diode to VDD. PTA0 should not be driven above VDD. Also, PTA0 does not pullup to VDD when internal pullup is enabled.
3. IRQ pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1)
4. The RESET pin contains integrated pullup device enabled if reset enabled (RSTPE = 1)
5. **High current drive**
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).

Figure 8-1. MC9S08RC/RD/RE/RG Block Diagram

12.2.3 SCI Control Register 2 (SCI1C2)

This register can be read or written at any time.

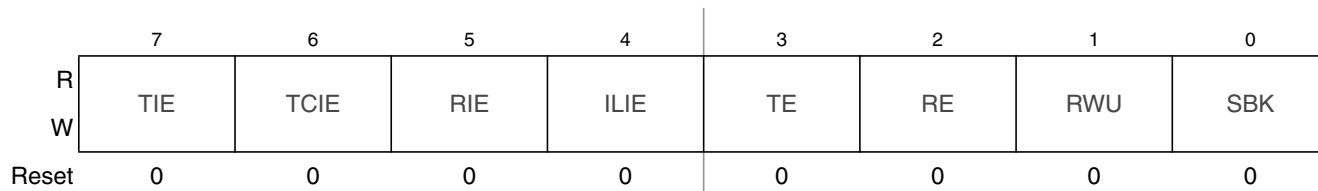


Figure 12-6. SCI Control Register 2 (SCI1C2)

Table 12-4. SCI1C2 Register Field Descriptions

Field	Description
7 TIE	<p>Transmit Interrupt Enable (for TDRE)</p> <p>0 Hardware interrupts from TDRE disabled (use polling). 1 Hardware interrupt requested when TDRE flag is 1.</p>
6 TCIE	<p>Transmission Complete Interrupt Enable (for TC)</p> <p>0 Hardware interrupt requested when TC flag is 1. 1 Hardware interrupts from TC disabled (use polling).</p>
5 RIE	<p>Receiver Interrupt Enable (for RDRF)</p> <p>0 Hardware interrupts from RDRF disabled (use polling). 1 Hardware interrupt requested when RDRF flag is 1.</p>
4 ILIE	<p>Idle Line Interrupt Enable (for IDLE)</p> <p>0 Hardware interrupts from IDLE disabled (use polling). 1 Hardware interrupt requested when IDLE flag is 1.</p>
3 TE	<p>Transmitter Enable</p> <p>0 Transmitter off. 1 Transmitter on.</p> <p>TE must be 1 in order to use the SCI transmitter. Normally, when TE = 1, the SCI forces the TxD pin to act as an output for the SCI system. If LOOPS = 1 and RSRC = 0, the TxD pin reverts to being a port B general-purpose I/O pin even if TE = 1.</p> <p>When the SCI is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single SCI communication line (TxD pin).</p> <p>TE also can be used to queue an idle character by writing TE = 0 then TE = 1 while a transmission is in progress. Refer to Section 12.3.2.1, “Send Break and Queued Idle,” for more details.</p> <p>When TE is written to 0, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin.</p>
2 RE	<p>Receiver Enable — When the SCI receiver is off, the RxD pin reverts to being a general-purpose port I/O pin.</p> <p>0 Receiver off. 1 Receiver on.</p>

has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full ($RDRF = 1$), it gets the data from the receive data register by reading SCI1D. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user's program that handles receive data. Refer to Section 12.3.4, "Interrupts and Status Flags," for more details about flag clearing.

12.3.3.1 Data Sampling Technique

The SCI receiver uses a $16\times$ baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD1 serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The $16\times$ baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

12.3.3.2 Receiver Wakeup Operation

Receiver wakeup is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCI1C2. When $RWU = 1$, it inhibits setting of the status flags associated with the receiver, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

Figure 15-2 shows an external host transmitting a logic 1 or 0 to the BKGD pin of a target HCS08 MCU. The host is asynchronous to the target so there is a 0-to-1 cycle delay from the host-generated falling edge to where the target perceives the beginning of the bit time. Ten target BDC clock cycles later, the target senses the bit level on the BKGD pin. Typically, the host actively drives the pseudo-open-drain BKGD pin during host-to-target transmissions to speed up rising edges. Because the target does not drive the BKGD pin during the host-to-target transmission period, there is no need to treat the line as an open-drain signal during this period.

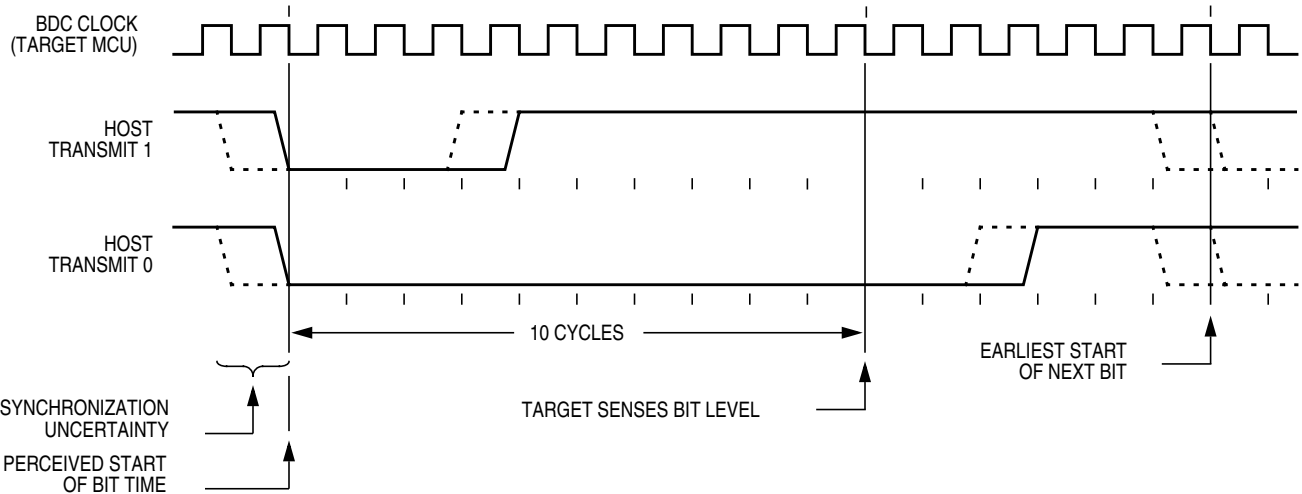


Figure 15-2. BDC Host-to-Target Serial Bit Timing

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

15.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

15.4.1.1 BDC Status and Control Register (BDCSCR)

This register can be read or written by serial BDC commands (READ_STATUS and WRITE_CONTROL) but is not accessible to user programs because it is not located in the normal memory map of the MCU.

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
W								
Normal Reset	0	0	0	0	0	0	0	0
Reset in Active BDM:	1	1	0	0	1	0	0	0

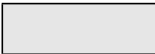
 = Unimplemented or Reserved

Figure 15-5. BDC Status and Control Register (BDCSCR)

Table 15-2. BDCSCR Register Field Descriptions

Field	Description
7 ENBDM	Enable BDM (Permit Active Background Mode) — Typically, this bit is written to 1 by the debug host shortly after the beginning of a debug session or whenever the debug host resets the target and remains 1 until a normal reset clears it. 0 BDM cannot be made active (non-intrusive commands still allowed) 1 BDM can be made active to allow active background mode commands
6 BDMACT	Background Mode Active Status — This is a read-only status bit. 0 BDM not active (user application program running) 1 BDM active and waiting for serial commands
5 BKPTEN	BDC Breakpoint Enable — If this bit is clear, the BDC breakpoint is disabled and the FTS (force tag select) control bit and BDCBKPT match register are ignored. 0 BDC breakpoint disabled 1 BDC breakpoint enabled
4 FTS	Force/Tag Select — When FTS = 1, a breakpoint is requested whenever the CPU address bus matches the BDCBKPT match register. When FTS = 0, a match between the CPU address bus and the BDCBKPT register causes the fetched opcode to be tagged. If this tagged opcode ever reaches the end of the instruction queue, the CPU enters active background mode rather than executing the tagged opcode. 0 Tag opcode at breakpoint address and enter active background mode if CPU attempts to execute that instruction 1 Breakpoint match forces active background mode at next instruction boundary (address need not be an opcode)
3 CLKSW	Select Source for BDC Communications Clock — CLKSW defaults to 0, which selects the alternate BDC clock source. 0 Alternate BDC clock source 1 MCU bus clock

15.4.3.8 Debug Trigger Register (DBGT)

This register can be read any time, but may be written only if ARM = 0, except bits 4 and 5 are hard-wired to 0s.

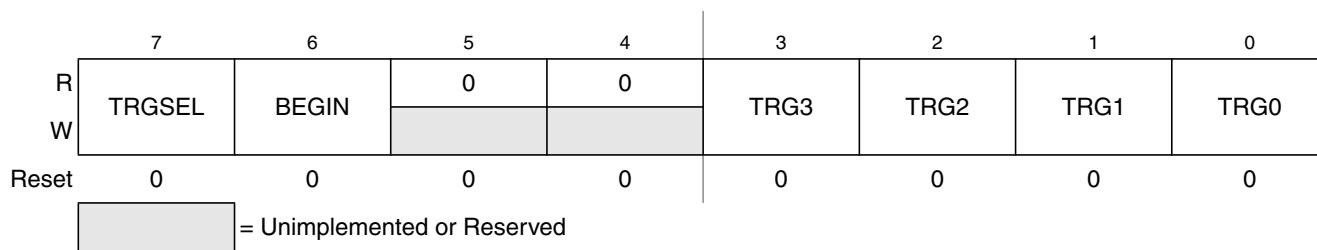


Figure 15-8. Debug Trigger Register (DBGT)

Table 15-5. DBGT Register Field Descriptions

Field	Description
7 TRGSEL	<p>Trigger Type — Controls whether the match outputs from comparators A and B are qualified with the opcode tracking logic in the debug module. If TRGSEL is set, a match signal from comparator A or B must propagate through the opcode tracking logic and a trigger event is only signalled to the FIFO logic if the opcode at the match address is actually executed.</p> <p>0 Trigger on access to compare address (force) 1 Trigger if opcode at compare address is executed (tag)</p>
6 BEGIN	<p>Begin/End Trigger Select — Controls whether the FIFO starts filling at a trigger or fills in a circular manner until a trigger ends the capture of information. In event-only trigger modes, this bit is ignored and all debug runs are assumed to be begin traces.</p> <p>0 Data stored in FIFO until trigger (end trace) 1 Trigger initiates data storage (begin trace)</p>
3:0 TRG[3:0]	<p>Select Trigger Mode — Selects one of nine triggering modes, as described below.</p> <p>0000 A-only 0001 A OR B 0010 A Then B 0011 Event-only B (store data) 0100 A then event-only B (store data) 0101 A AND B data (full mode) 0110 A AND NOT B data (full mode) 0111 Inside range: $A \leq \text{address} \leq B$ 1000 Outside range: $\text{address} < A$ or $\text{address} > B$ 1001 – 1111 (No trigger)</p>

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*.