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	Not For New Designs
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	56
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gb60acfue

MC9S08GB60A
MC9S08GB32A
MC9S08GT60A
MC9S08GT32A

Data Sheet

HCS08
Microcontrollers

MC9S08GB60A
Rev. 2
07/2008

freescale.com

freescale™
semiconductor

MC9S08GB60A Data Sheet

Covers: MC9S08GB60A
MC9S08GB32A
MC9S08GT60A
MC9S08GT32A

MC9S08GB60A
Rev. 2
07/2008

freescale™
semiconductor

Chapter 8
Central Processor Unit (S08CPUV2)

Table 4-2. Direct-Page Register Summary (Sheet 1 of 3)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	**PTAD**	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
0x0001	**PTAPE**	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
0x0002	**PTASE**	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
0x0003	**PTADD**	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
0x0004	**PTBD**	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0x0005	**PTBPE**	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0x0006	**PTBSE**	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
0x0007	**PTBDD**	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0008	**PTCD**	PTCD7	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0009	**PTCPE**	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0x000A	**PTCSE**	PTCSE7	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
0x000B	**PTCDD**	PTCDD7	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x000C	**PTDD**	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
0x000D	**PTDPE**	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
0x000E	**PTDSE**	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
0x000F	**PTDDD**	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0010	**PTED**	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0011	**PTEPE**	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
0x0012	**PTESE**	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
0x0013	**PTEDD**	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x0014	**IRQSC**	0	0	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
0x0015	Reserved	—	—	—	—	—	—	—	—
0x0016	**KBI1SC**	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
0x0017	**KBI1PE**	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
0x0018	**SCI1BDH**	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0019	**SCI1BDL**	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x001A	**SCI1C1**	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x001B	**SCI1C2**	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x001C	**SCI1S1**	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x001D	**SCI1S2**	0	0	0	0	0	0	0	RAF
0x001E	**SCI1C3**	R8	T8	TXDIR	0	ORIE	NEIE	FEIE	PEIE
0x001F	**SCI1D**	Bit 7	6	5	4	3	2	1	Bit 0
0x0020	**SCI2BDH**	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0021	**SCI2BDL**	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x0022	**SCI2C1**	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x0023	**SCI2C2**	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x0024	**SCI2S1**	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x0025	**SCI2S2**	0	0	0	0	0	0	0	RAF
0x0026	**SCI2C3**	R8	T8	TXDIR	0	ORIE	NEIE	FEIE	PEIE
0x0027	**SCI2D**	Bit 7	6	5	4	3	2	1	Bit 0

5.8.1 Interrupt Pin Request Status and Control Register (IRQSC)

This direct page register includes two unimplemented bits which always read 0, four read/write bits, one read-only status bit, and one write-only bit. These bits are used to configure the IRQ function, report status, and acknowledge IRQ events.

	7	6	5	4	3	2	1	0
R	0	0	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
W						IRQACK		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-2. Interrupt Request Status and Control Register (IRQSC)

Table 5-2. IRQSC Field Descriptions

Field	Description
5 IRQEDG	**Interrupt Request (IRQ) Edge Select** — This read/write control bit is used to select the polarity of edges or levels on the IRQ pin that cause IRQF to be set. The IRQMOD control bit determines whether the IRQ pin is sensitive to both edges and levels or only edges. When the IRQ pin is enabled as the IRQ input and is configured to detect rising edges, the optional pullup resistor is re-configured as an optional pulldown resistor. 0 IRQ is falling edge or falling edge/low-level sensitive. 1 IRQ is rising edge or rising edge/high-level sensitive.
4 IRQPE	**IRQ Pin Enable** — This read/write control bit enables the IRQ pin function. When this bit is set, the IRQ pin can be used as an interrupt request. Also, when this bit is set, either an internal pull-up or an internal pull-down resistor is enabled depending on the state of the IRQMOD bit. 0 IRQ pin function is disabled. 1 IRQ pin function is enabled.
3 IRQF	**IRQ Flag —** This read-only status bit indicates when an interrupt request event has occurred. 0 No IRQ request. 1 IRQ event detected.
2 IRQACK	**IRQ Acknowledge** — This write-only bit is used to acknowledge interrupt request events (write 1 to clear IRQF). Writing 0 has no meaning or effect. Reads always return 0. If edge-and-level detection is selected (IRQMOD = 1), IRQF cannot be cleared while the IRQ pin remains at its asserted level.
1 IRQIE	**IRQ Interrupt Enable** — This read/write control bit determines whether IRQ events generate a hardware interrupt request. 0 Hardware interrupt requests from IRQF disabled (use polling). 1 Hardware interrupt requested whenever IRQF = 1.
0 IRQMOD	**IRQ Detection Mode** — This read/write control bit selects either edge-only detection or edge-and-level detection. The IRQEDG control bit determines the polarity of edges and levels that are detected as interrupt request events. See Section 5.5.2.2, “Edge and Level Sensitivity” for more details. 0 IRQ event on falling edges or rising edges only. 1 IRQ event on falling edges and low levels or on rising edges and high levels.

Port A can be configured to be keyboard interrupt input pins. Refer to Chapter 9, “Keyboard Interrupt (S08KBIV1),” for more information about using port A pins as keyboard interrupts pins.

6.3.2 Port B and Analog to Digital Converter Inputs

Port B	Bit 7	6	5	4	3	2	1	Bit 0
MCU Pin:	PTB7/ AD1P7	PTB6/ AD1P6	PTB5/ AD1P5	PTB4/ AD1P4	PTB3/ AD1P3	PTB2/ AD1P2	PTB1/ AD1P1	PTB0/ AD1P0

Figure 6-3. Port B Pin Names

Port B is an 8-bit port shared among the ATD inputs and general-purpose I/O. Any pin enabled as an ATD input will be forced to act as an input.

Port B pins are available as general-purpose I/O pins controlled by the port B data (PTBD), data direction (PTBDD), pullup enable (PTBPE), and slew rate control (PTBSE) registers. Refer to Section 6.4, “Parallel I/O Controls,” for more information about general-purpose I/O control.

When the ATD module is enabled, analog pin enables are used to specify which pins on port B will be used as ATD inputs. Refer to Chapter 14, “Analog-to-Digital Converter (S08ATDV3),” for more information about using port B pins as ATD pins.

6.3.3 Port C and SCI2, IIC, and High-Current Drivers

Port C	Bit 7	6	5	3	3	2	1	Bit 0
MCU Pin:	PTC7	PTC6	PTC5	PTC4	PTC3/ SCL1	PTC2/ SDA1	PTC1/ RxD2	PTC0/ TxD2

Figure 6-4. Port C Pin Names

Port C is an 8-bit port which is shared among the SCI2 and IIC1 modules, and general-purpose I/O. When SCI2 or IIC1 modules are enabled, the pin direction will be controlled by the module or function. Port C has high current output drivers.

Port C pins are available as general-purpose I/O pins controlled by the port C data (PTCD), data direction (PTCDD), pullup enable (PTCPE), and slew rate control (PTCSE) registers. Refer to Section 6.4, “Parallel I/O Controls,” for more information about general-purpose I/O control.

When the SCI2 module is enabled, PTC0 serves as the SCI2 module’s transmit pin (TxD2) and PTC1 serves as the receive pin (RxD2). Refer to Chapter 11, “Serial Communications Interface (S08SCIV1),” for more information about using PTC0 and PTC1 as SCI pins

When the IIC module is enabled, PTC2 serves as the IIC modules’s serial data input/output pin (SDA1) and PTC3 serves as the clock pin (SCL1). Refer to Chapter 13, “Inter-Integrated Circuit (S08IICV1),” for more information about using PTC2 and PTC3 as IIC pins.

7.4.4 Example #3: No External Crystal Connection, 5.4 MHz Bus Frequency

In this example, the FLL will be used (in FEI mode) to multiply the internal 243 kHz (approximate) reference clock up to 10.8 MHz to achieve 5.4 MHz bus frequency. This system will also use the trim function to fine tune the frequency based on an external reference signal.

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

The clock scheme will be FLL engaged, internal (FEI). So

$$f_{ICGOUT} = (f_{IRG} / 7) * P * N / R ; P = 64, f_{IRG} = 243 \text{ kHz} \quad \textit{Eqn. 7-5}$$

Solving for N / R gives:

$$N / R = 10.8 \text{ MHz} /(243/7 \text{ kHz} * 64) = 4.86 ; \text{ We can choose N = 10 and R = 2.} \quad \textit{Eqn. 7-6}$$

A trim procedure will be required to hone the frequency to exactly 5.4 MHz. An example of the trim procedure is shown in example #4.

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

ICGC1 = $28 (%00101000)

Bit 7	HGO	0	Configures oscillator for low-power operation
Bit 6	RANGE	0	Configures oscillator for low-frequency range; FLL prescale factor is 64
Bit 5	REFS	1	Oscillator using crystal or resonator requested (bit is really a don't care)
Bits 4:3	CLKS	01	FLL engaged, internal reference clock mode
Bit 2	OSCSTEN	0	Disables the oscillator in stop modes
Bit 1	LOCD	0	Loss-of-clock detection enabled
Bit 0		0	Unimplemented or reserved, always reads zero

ICGC2 = $31 (%00110001)

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bit 6:4	MFD	011	Sets the MFD multiplication factor to 10
Bit 3	LOCRE	0	Generates an interrupt request on loss of clock
Bit 2:0	RFD	001	Sets the RFD division factor to ÷2

ICGS1 = $xx

This is read only except for clearing interrupt flag

ICGS2 = $xx

This is read only; good idea to read this before performing time critical operations

ICGFLTLU/L = $xx

Not used in this example

ICGTRM = $xx

Bit 7:0	TRIM	Only need to write when trimming internal oscillator; done in separate operation (see example #4)

Chapter 8
Central Processor Unit (S08CPUV2)

8.1 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).

8.1.1 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

9.3.1 KBI Status and Control Register (KBI1SC)

	7	6	5	4	3	2	1	0
R	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
W						KBACK		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-4. KBI Status and Control Register (KBI1SC)

Table 9-1. KBI1SC Register 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 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. KBF can be used as a software pollable flag (KBIE = 0) or it can generate a hardware interrupt request to the CPU (KBIE = 1). 0 No KBI interrupt pending 1 KBI interrupt pending
2 KBACK	**Keyboard Interrupt Acknowledge** — This write-only bit (reads always return 0) is used to clear the KBF status flag by writing a 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
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: • 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

associated TPM act as center-aligned PWM channels. When CPWMS = 0, each channel can independently be configured to operate in input capture, output compare, or buffered edge-aligned PWM mode.

The following sections describe the main 16-bit counter and each of the timer operating modes (input capture, output compare, edge-aligned PWM, and center-aligned PWM). Because details of pin operation and interrupt activity depend on the operating mode, these topics are covered in the associated mode sections.

10.5.1 Counter

All timer functions are based on the main 16-bit counter (TPMxCNTH:TPMxCNTL). This section discusses selection of the clock source, up-counting vs. up-/down-counting, end-of-count overflow, and manual counter reset.

After any MCU reset, CLKSB:CLKSA = 0:0 so no clock source is selected and the TPM is inactive. Normally, CLKSB:CLKSA would be set to 0:1 so the bus clock drives the timer counter. The clock source for each of the TPM can be independently selected to be off, the bus clock (BUSCLK), the fixed system clock (XCLK), or an external input through the TPMxCH0 pin. The maximum frequency allowed for the external clock option is one-fourth the bus rate. Refer to Section 10.7.1, “Timer x Status and Control Register (TPMxSC),” and Table 10-2 for more information about clock source selection.

When the microcontroller is in active background mode, the TPM temporarily suspends all counting until the microcontroller returns to normal user operating mode. During stop mode, all TPM clocks are stopped; therefore, the TPM is effectively disabled until clocks resume. During wait mode, the TPM continues to operate normally.

The main 16-bit counter has two counting modes. When center-aligned PWM is selected (CPWMS = 1), the counter operates in up-/down-counting mode. Otherwise, the counter operates as a simple up-counter. As an up-counter, the main 16-bit counter counts from $0000 through its terminal count and then continues with $0000. The terminal count is $FFFF or a modulus value in TPMxMODH:TPMxMODL.

When center-aligned PWM operation is specified, the counter counts upward from $0000 through its terminal count and then counts downward to $0000 where it returns to up-counting. Both $0000 and the terminal count value (value in TPMxMODH:TPMxMODL) are normal length counts (one timer clock period long).

An interrupt flag and enable are associated with the main 16-bit counter. The timer overflow flag (TOF) is a software-accessible indication that the timer counter has overflowed. The enable signal selects between software polling (TOIE = 0) where no hardware interrupt is generated, or interrupt-driven operation (TOIE = 1) where a static hardware interrupt is automatically generated whenever the TOF flag is 1.

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the main 16-bit counter counts from $0000 through $FFFF and overflows to $0000 on the next counting clock. TOF becomes set at the transition from $FFFF to $0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to $0000. When the main 16-bit counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The $0000 count value corresponds to the center of a period.)

Chapter 13
Inter-Integrated Circuit (S08IICV1)

13.1 Introduction

The MC9S08GBxxA/GTxxA series of microcontrollers provides one inter-integrated circuit (IIC) module for communication with other integrated circuits. The two pins associated with this module, SDA1 and SCL1 share port C pins 2 and 3, respectively. All functionality as described in this section is available on MC9S08GBxxA/GTxxA. When the IIC is enabled, the direction of pins is controlled by module configuration. If the IIC is disabled, both pins can be used as general-purpose I/O.

Table 14-1. Signal Properties

Name	Function
AD7–AD0	Channel input pins
V_{REFH}	High reference voltage for ATD converter
V_{REFL}	Low reference voltage for ATD converter
V_{DDAD}	ATD power supply voltage
V_{SSAD}	ATD ground supply voltage

14.2.1.1 Channel Input Pins — AD1P7–AD1P0

The channel pins are used as the analog input pins of the ATD. Each pin is connected to an analog switch which serves as the signal gate into the sample submodule.

14.2.1.2 ATD Reference Pins — V_{REFH}, V_{REFL}

These pins serve as the source for the high and low reference potentials for the converter. Separation from the power supply pins accommodates the filtering necessary to achieve the accuracy of which the system is capable.

14.2.1.3 ATD Supply Pins — V_{DDAD}, V_{SSAD}

These two pins are used to supply power and ground to the analog section of the ATD. Dedicated power is required to isolate the sensitive analog circuitry from the normal levels of noise present on digital power supplies.

NOTE

V_{DDAD1} and V_{DD} must be at the same potential. Likewise, V_{SSAD1} and V_{SS} must be at the same potential.

14.3 Functional Description

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

A write to the ATD1SC register initiates a new conversion. A write to the ATD1C register will interrupt the current conversion but it will not initiate a new conversion. A write to the ATD1PE register will also abort the current conversion but will not initiate a new conversion. If a conversion is already running when a write to the ATD1SC register is made, it will be aborted and a new one will be started.

14.3.1 Mode Control

The ATD has a mode control unit to communicate with the sample and hold (S/H) machine and the SAR machine when necessary to collect samples and perform conversions. The mode control unit signals the S/H machine to begin collecting a sample and for the SAR machine to begin receiving a sample. At the end of the sample period, the S/H machine signals the SAR machine to begin the analog-to-digital conversion process. The conversion process is terminated when the SAR machine signals the end of

Table 14-5. Clock Prescaler Values

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

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

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

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

14.6.2 ATD Status and Control (ATD1SC)

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

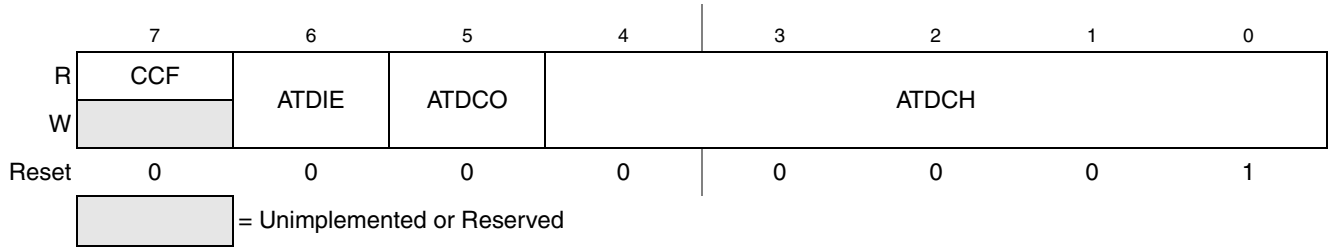

Figure 14-6. ATD Status and Control Register (ATD1SC)

When no debugger pod is connected to the 6-pin BDM interface connector, the internal pullup on BKGD chooses normal operating mode. When a debug pod is connected to BKGD it is possible to force the MCU into active background mode after reset. The specific conditions for forcing active background depend upon the HCS08 derivative (refer to the introduction to this Development Support section). It is not necessary to reset the target MCU to communicate with it through the background debug interface.

15.2.2 Communication Details

The BDC serial interface requires the external controller to generate a falling edge on the BKGD pin to indicate the start of each bit time. The external controller provides this falling edge whether data is transmitted or received.

BKGD is a pseudo-open-drain pin that can be driven either by an external controller or by the MCU. Data is transferred MSB first at 16 BDC clock cycles per bit (nominal speed). The interface times out if 512 BDC clock cycles occur between falling edges from the host. Any BDC command that was in progress when this timeout occurs is aborted without affecting the memory or operating mode of the target MCU system.

The custom serial protocol requires the debug pod to know the target BDC communication clock speed.

The clock switch (CLKSW) control bit in the BDC status and control register allows the user to select the BDC clock source. The BDC clock source can either be the bus or the alternate BDC clock source.

The BKGD pin can receive a high or low level or transmit a high or low level. The following diagrams show timing for each of these cases. Interface timing is synchronous to clocks in the target BDC, but asynchronous to the external host. The internal BDC clock signal is shown for reference in counting cycles.

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.

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.

Appendix A
Electrical Characteristics

A.1 Introduction

This section contains electrical and timing specifications.

A.2 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only, and functional operation at the maxima is not guaranteed. Stress beyond the limits specified in Table A-1 may affect device reliability or cause permanent damage to the device. For functional operating conditions, refer to the remaining tables in this section.

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (for instance, either V_{SS} or V_{DD}) or the programmable pull-up resistor associated with the pin is enabled.

Table A-1. Absolute Maximum Ratings

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +3.8	V
Maximum current into V_{DD}	I_{DD}	120	mA
Digital input voltage	V_{In}	–0.3 to V_{DD} + 0.3	V
Instantaneous maximum current Single pin limit (applies to all port pins)[1, 2, 3]	I_D	± 25	mA
Storage temperature range	T_{stg}	–55 to 150	°C

1 Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive (V_{DD}) and negative (V_{SS}) clamp voltages, then use the larger of the two resistance values.

2 All functional non-supply pins are internally clamped to V_{SS} and V_{DD}.

3 Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{In} > V_{DD}$) is greater than I_{DD}, the injection current may flow out of V_{DD} and could result in external power supply going out of regulation. Ensure external V_{DD} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if the clock rate is very low which would reduce overall power consumption.

A.8.1 ICG Frequency Specifications

Table A-9. ICG Frequency Specifications
(V_{DDA} = V_{DDA} (min) to V_{DDA} (max), Temperature Range = –40 to 85°C Ambient)

Characteristic	Symbol	Min	Typical	Max	Unit
Oscillator crystal or resonator[4] (REFS = 1) (Fundamental mode crystal or ceramic resonator) Low range High range High Gain, FBE (HGO=1,CLKS = 10) High Gain, FEE (HGO=1,CLKS = 11) Low Power, FBE (HGO=0, CLKS=10) Low Power, FEE (HGO=0, CLKS=11)	 flo fhi_byp fhi_eng flp_byp flp_eng	 32 1 2 1 2	 — — — — —	 100 16 10 10 10	 kHz MHz MHz MHz MHz
Input clock frequency (CLKS = 11, REFS = 0) Low range High range	 f_{lo} f_{hi_eng}	 32 2	 — —	 100 10	 kHz MHz
Input clock frequency (CLKS = 10, REFS = 0)	f_{Extal}	0	—	40	MHz
Internal reference frequency (untrimmed)	$f_{ICGIRCLK}$	182.25	243	303.75	kHz
Duty cycle of input clock[4] (REFS = 0)	t_{dc}	40	—	60	%
Output clock ICGOUT frequency CLKS = 10, REFS = 0 All other cases	f_{ICGOUT}	 f_{Extal} (min) f_{lo} (min)		 f_{Extal} (max) $f_{ICGDCLKmax}$ (max)	MHz
Minimum DCO clock (ICGDCLK) frequency	$f_{ICGDCLKmin}$	8	—		MHz
Maximum DCO clock (ICGDCLK) frequency	$f_{ICGDCLKmax}$		—	40	MHz
Self-clock mode (ICGOUT) frequency [1]	f_{Self}	$f_{ICGDCLKmin}$		$f_{ICGDCLKmax}$	MHz
Self-clock mode reset (ICGOUT) frequency	f_{Self_reset}	5.5	8	10.5	MHz
Loss of reference frequency [2] Low range High range	f_{LOR}	 5 50		 25 500	kHz
Loss of DCO frequency [3]	f_{LOD}	 0.5		 1.5	MHz
Crystal start-up time [4, 5] Low range High range	 t_{CSTL} t_{CSTH}	 — —	 430 4	 — —	ms
FLL lock time [4, 6] Low range High range	 t_{Lockl} t_{Lockh}	 — —		 2 2	ms
FLL frequency unlock range	n_{Unlock}	–4*N		4*N	counts
FLL frequency lock range	n_{Lock}	–2*N		2*N	counts
ICGOUT period jitter, [4, 7] measured at f_{ICGOUT} Max Long term jitter (averaged over 2 ms interval)	C_{Jitter}	 —		 0.2	% f_{ICG}
Internal oscillator deviation from trimmed frequency[8] V_{DD} = 1.8 – 3.6 V, (constant temperature) V_{DD} = 3.0 V ±10%, –40° C to 85° C	ACC_{int}	 — —	 ±0.5 ±0.5	 ±2 ±2	%

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 Chapter 4, "Memory."

Table A-13. Flash Characteristics

Characteristic	Symbol	Min	Typical	Max	Unit
Supply voltage for program/erase	$V_{prog/erase}$	1.8		3.6	V
Supply voltage for read operation $0 < f_{Bus} < 8$ MHz $0 < f_{Bus} < 20$ MHz	V_{Read}	 1.8 2.08		 3.6 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°C to + 85°C T = 25°C		10,000	 100,000	 —	cycles
Data retention[4]	t_{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.*