E·XFL

Motorola - MC68HC908GT8CFB Datasheet

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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	34
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68hc908gt8cfb

Revision History (Sheet 2 of 2)

Date	Revision Level	Description	Page Number(s)
September, 2004	3.0 (Continued from previous page)	Chapter 15 System Integration Module (SIM) — Clarified SIM features and functionality	179, 182, 183, 184
		15.7.2 SIM Reset Status Register — Clarified SRSR operation	194
		Table 19-1. Monitor Mode Signal Requirements and Options — Reworked	247
		19.2.1 Functional Description — Corrected Break description	237, 240
		19.3 Monitor Module (MON) — Reworked	243
		Chapter 20 Electrical Specifications — Revised/added tables: 20.5 5.0-V DC Electrical Characteristics 20.6 3.0-V DC Electrical Characteristics 20.7 Supply Current Characteristics 20.8 5-V Control Timing 20.9 3-V Control Timing	 257 258 259 260 260
		20.20 Memory Characteristics — Updated memory table	273
		Chapter 20 Electrical Specifications — Added figures: Figure 20-1. RST and IRQ Timing Figure 20-2. RST and IRQ Timing	 260 260
March, 2006	4.0	Appendix A MC68HC08GT16 — Introduces the MC68HC08GT16, the ROM part equivalent to the MC68HC908GT16.	281
April, 2007	5.0	4.2 Functional Description — In the description of the COP Rate Select Bit corrected the values for COP timeout period	57
		Figure 5-1. COP Block Diagram — Replaced BUSCLKX4 with COPCLK	61
		14.9.1 ESCI Arbiter Control Register — Replaced one half with one quarter in definition for ACLK = 0	176
		14.9.3 Bit Time Measurement — Replaced one half with one quarter in definition for ACLK = 0	177
		Revised the following diagrams: Figure 19-10. Forced Monitor Mode (Low) Figure 19-11. Forced Monitor Mode (High) Figure 19-12. Standard Monitor Mode	 245 245 246

Chapter 6
Central Processor Unit (CPU)

6.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

6.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

6.3 CPU Registers

Figure 6-1 shows the five CPU registers. CPU registers are not part of the memory map.

7.3.2.3 Frequency Comparator

The frequency comparator effectively compares the low-frequency base clock (IBASE) to a nominal frequency, f_{NOM}. First, the frequency comparator converts IBASE to a voltage by charging a known capacitor with a current reference for a period dependent on IBASE. This voltage is compared to a voltage reference with comparators, whose outputs are fed to the digital loop filter. The dependence of these outputs on the capacitor size, current reference, and voltage reference causes up to ±25 percent error in f_{NOM}.

7.3.2.4 Digital Loop Filter

The digital loop filter (DLF) uses the outputs of the frequency comparator to adjust the internal clock (ICLK) clock period. The DLF generates the DCO divider control bits (DDIV[3:0]) and the DCO stage control bits (DSTG[7:0]), which are fed to the DCO. The DLF first concatenates the DDIV and DSTG registers (DDIV[3:0]:DSTG[7:0]) and then adds or subtracts a value dependent on the relative error in the low-frequency base clock's period, as shown in Table 7-1. In some extreme error conditions, such as operating at a V_{DD} level which is out of specification, the DLF may attempt to use a value above the maximum ($9FF) or below the minimum ($000). In both cases, the value for DDIV will be between $A and $F. In this range, the DDIV value will be interpreted the same as $9 (the slowest condition). Recovering from this condition requires subtracting (increasing frequency) in the normal fashion until the value is again below $9FF. (If the desired value is $9xx, the value may settle at $Axx through $Fxx. This is an acceptable operating condition.) If the error is less than ±5 percent, the internal clock generator's filter stable indicator (FICGS) is set, indicating relative frequency accuracy to the clock monitor.

Table 7-1. Correction Sizes from DLF to DCO

Frequency Error of IBASE Compared to f_{NOM}	DDVI[3:0]:DSTG[7:0] Correction	Current to New DDIV[3:0]:DSTG[7:0][1]		Relative Correction in DCO	
IBASE < 0.85 f_{NOM}	–32 (–$020)	Minimum	$xFF to $xDF	–2/31	–6.45%
		Maximum	$x20 to $x00	–2/19	–10.5%
0.85 f_{NOM} < IBASE IBASE < 0.95 f_{NOM}	–8 (–$008)	Minimum	$xFF to $xF7	–0.5/31	–1.61%
		Maximum	$x08 to $x00	–0.5/17.5	–2.86%
0.95 f_{NOM} < IBASE IBASE < f_{NOM}	–1 (–$001)	Minimum	$xFF to $xFE	–0.0625/31	–0.202%
		Maximum	$x01 to $x00	–0.0625/17.0625	–0.366%
f_{NOM} < IBASE IBASE < 1.05 f_{NOM}	+1 (+$001)	Minimum	$xFE to $xFF	+0.0625/30.9375	+0.202%
		Maximum	$x00 to $x01	+0.0625/17	+0.368%
1.05 f_{NOM} < IBASE IBASE < 1.15 f_{NOM}	+8 (+$008)	Minimum	$xF7 to $xFF	+0.5/30.5	+1.64%
		Maximum	$x00 to $x08	+0.5/17	+2.94%
1.15 f_{NOM} < IBASE	+32 (+$020)	Minimum	$xDF to $xFF	+2/29	+6.90%
		Maximum	$x00 to $x20	+2/17	+11.8%

[1] x = Maximum error is independent of value in DDIV[3:0]. DDIV increments or decrements when an addition to DSTG[7:0] carries or borrows.

7.4.4.4 Ring Oscillator Fine-Adjust Circuit

The ring oscillator fine-adjust circuit causes the ring oscillator to effectively operate at non-integer numbers of stage delays by operating at two different points for a variable number of cycles specified by the lower five DCO stage control bits (DSTG[4:0]). For example:

- When DSTG[7:5] is %011, the ring oscillator nominally operates at 23 stage delays.
- When DSTG[4:0] is %00000, the ring will always operate at 23 stage delays.
- When DSTG[4:0] is %00001, the ring will operate at 25 stage delays for one of 32 cycles and at 23 stage delays for 31 of 32 cycles.
- Likewise, when DSTG[4:0] is %11111, the ring operates at 25 stage delays for 31 of 32 cycles and at 23 stage delays for one of 32 cycles.
- When DSTG[7:5] is %111, similar results are achieved by including a variable divide-by-two, so the ring operates at 31 stages for some cycles and at 17 stage delays, with a divide-by-two for an effective 34 stage delays, for the remainder of the cycles.

Adjusting the DSTG[0] bit has a 0.202 percent to 0.368 percent effect on the output clock period. This corresponds to the minimum size correction made by the DLF, and the inherent, long-term quantization error in the output frequency.

7.4.5 Switching Internal Clock Frequencies

The frequency of the internal clock (ICLK) may need to be changed for some applications. For example, if the reset condition does not provide the correct frequency, or if the clock is slowed down for a low-power mode (or sped up after a low-power mode), the frequency must be changed by programming the internal clock multiplier factor (N). The frequency of ICLK is N times the frequency of IBASE, which is 307.2 kHz ±25 percent.

Before switching frequencies by changing the N value, the clock monitor must be disabled. This is because when N is changed, the frequency of the low-frequency base clock (IBASE) will change proportionally until the digital loop filter has corrected the error. Since the clock monitor uses IBASE, it could erroneously detect an inactive clock. The clock monitor cannot be re-enabled until the internal clock is stable again (ICGS is set).

The following flow is an example of how to change the clock frequency:

- Verify there is no clock monitor interrupt by reading the CMF bit.
- Turn off the clock monitor.
- If desired, switch to the external clock (see 7.4.1 Switching Clock Sources).
- Change the value of N.
- Switch back to internal (see 7.4.1 Switching Clock Sources), if desired.
- Turn on the clock monitor (see 7.4.2 Enabling the Clock Monitor), if desired.

7.4.6 Nominal Frequency Settling Time

Because the clock period of the internal clock (ICLK) is dependent on the digital loop filter outputs (DDIV and DSTG) which cannot change instantaneously, ICLK temporarily will operate at an incorrect clock period when any operating condition changes. This happens whenever the part is reset, the ICG multiply factor (N) is changed, the ICG trim factor (TRIM) is changed, or the internal clock is enabled after inactivity (stop mode or disabled operation). The time that the ICLK takes to adjust to the correct period is known as the settling time.

clock period tolerance plus 10 percent) must be added. This adjustment can be reduced with trimming. Table 7-3 shows some typical values for settling time.

Table 7-3. Typical Settling Time Examples

τ_1	τ_2	N	τ_{15}	τ_5	τ_{tot}
1/ (6.45 MHz)	1/ (25.8 MHz)	84	430 μs	535 μs	850 μs
1/ (25.8 MHz)	1/ (6.45 MHz)	21	107 μs	212 μs	525 μs
1/ (25.8 MHz)	1/ (307.2 kHz)	1	141 μs	246 μs	560 μs
1/ (307.2 kHz)	1/ (25.8 MHz)	84	11.9 ms	12.0 ms	12.3 ms

7.4.7 Trimming Frequency on the Internal Clock Generator

The unadjusted frequency of the low-frequency base clock (IBASE), when the comparators in the frequency comparator indicate zero error, will vary as much as ±25 percent due to process, temperature, and voltage dependencies. These dependencies are in the voltage and current references, the offset of the comparators, and the internal capacitor.

The method of changing the unadjusted operating point is by changing the size of the capacitor. This capacitor is designed with 639 equally sized units. Of that number, 384 of these units are always connected. The remaining 255 units are put in by adjusting the ICG trim factor (TRIM). The default value for TRIM is $80, or 128 units, making the default capacitor size 512. Each unit added or removed will adjust the output frequency by about ±0.195 percent of the unadjusted frequency (adding to TRIM will decrease frequency). Therefore, the frequency of IBASE can be changed to ±25 percent of its unadjusted value, which is enough to cancel the process variability mentioned before.

The best way to trim the internal clock is to use the timer to measure the width of an input pulse on an input capture pin (this pulse must be supplied by the application and should be as long or wide as possible). Considering the prescale value of the timer and the theoretical (zero error) frequency of the bus (307.2 kHz *N/4), the error can be calculated. This error, expressed as a percentage, can be divided by 0.195 percent and the resultant factor added or subtracted from TRIM. This process should be repeated to eliminate any residual error.

7.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power- consumption standby modes.

7.5.1 Wait Mode

The ICG remains active in wait mode. If enabled, the ICG interrupt to the CPU can bring the MCU out of wait mode.

In some applications, low power-consumption is desired in wait mode and a high-frequency clock is not needed. In these applications, reduce power consumption by either selecting a low-frequency external clock and turn the internal clock generator off or reduce the bus frequency by minimizing the ICG multiplier factor (N) before executing the WAIT instruction.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0006	Data Direction Register C (DDRC) See page 130.	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:	▨							
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD) See page 133.	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE) See page 135.	Read:	0	0	0	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:	▨	▨	▨					
		Reset:	Unaffected by reset							
$000C	Data Direction Register E (DDRE) See page 136.	Read:	0	0	0	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:	▨	▨	▨					
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 127.	Read:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Port C Input Pullup Enable Register (PTCPUE) See page 131.	Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
		Write:	▨							
		Reset:	0	0	0	0	0	0	0	0
$000F	Port D Input Pullup Enable Register (PTDPUE) See page 134.	Read:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 12-1. I/O Port Register Summary (Continued)

PTCPUE6–PTCPUE0 — Port C Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

1 = Corresponding port C pin configured to have internal pullup
0 = Corresponding port C pin internal pullup disconnected

12.5 Port D

Port D is an 8-bit special-function port that shares four of its pins with the serial peripheral interface (SPI) module and four of its pins with two timer interface (TIM1 and TIM2) modules. Port D also has software configurable pullup devices if configured as an input port.

12.5.1 Port D Data Register

The port D data register (PTD) contains a data latch for each of the eight port D pins.

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Reset:	Unaffected by reset							
Alternative Function:	T2CH1	T2CH0	$\overline{\text{T1CH1}}$	T1CH0	SPSCK	$\overline{\text{MOSI}}$	MISO	$\overline{\text{SS}}$

Figure 12-13. Port D Data Register (PTD)

PTD7–PTD0 — Port D Data Bits

These read/write bits are software-programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

T2CH1 and T2CH0 — Timer 2 Channel I/O Bits

The PTD7/T2CH1–PTD6/T2CH0 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTD7/T2CH1–PTD6/T2CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 18 Timer Interface Module (TIM).

T1CH1 and T1CH0 — Timer 1 Channel I/O Bits

The PTD7/T1CH1–PTD6/T1CH0 pins are the TIM1 input capture/output compare pins. The edge/level select bits, ELSxB and ELSxA, determine whether the PTD7/T1CH1–PTD6/T1CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 18 Timer Interface Module (TIM).

SPSCK — SPI Serial Clock

The PTD3/SPSCK pin is the serial clock input of the SPI module. When the SPE bit is clear, the PTD3/SPSCK pin is available for general-purpose I/O.

MOSI — Master Out/Slave In

The PTD2/MOSI pin is the master out/slave in terminal of the SPI module. When the SPE bit is clear, the PTD2/MOSI pin is available for general-purpose I/O.

MISO — Master In/Slave Out

The PTD1/MISO pin is the master in/slave out terminal of the SPI module. When the SPI enable bit, SPE, is clear, the SPI module is disabled, and the PTD0/$\overline{\text{SS}}$ pin is available for general-purpose I/O.

- Mode fault bit (MODF) — The MODF bit is set in a slave SPI if the $\overline{SS}$ pin goes high during a transmission with the mode fault enable bit (MODFEN) set. In a master SPI, the MODF bit is set if the $\overline{SS}$ pin goes low at any time with the MODFEN bit set. The error interrupt enable bit, ERRIE, enables MODF CPU interrupt requests. MODF, MODFEN, and ERRIE are in the SPI status and control register.
- Overflow bit (OVRF) — The OVRF bit is set if software does not read the byte in the receive data register before the next full byte enters the shift register. The error interrupt enable bit, ERRIE, enables OVRF CPU interrupt requests. OVRF and ERRIE are in the SPI status and control register.

13.3.2.8 Serial Communications Interface (SCI)

SCI CPU interrupt sources:

- SCI transmitter empty bit (SCTE) — SCTE is set when the SCI data register transfers a character to the transmit shift register. The SCI transmit interrupt enable bit, SCTIE, enables transmitter CPU interrupt requests. SCTE is in SCI status register 1. SCTIE is in SCI control register 2.
- Transmission complete bit (TC) — TC is set when the transmit shift register and the SCI data register are empty and no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, enables transmitter CPU interrupt requests. TC is in SCI status register 1. TCIE is in SCI control register 2.
- SCI receiver full bit (SCRF) — SCRF is set when the receive shift register transfers a character to the SCI data register. The SCI receive interrupt enable bit, SCRIE, enables receiver CPU interrupts. SCRF is in SCI status register 1. SCRIE is in SCI control register 2.
- Idle input bit (IDLE) — IDLE is set when 10 or 11 consecutive 1s shift in from the RxD pin. The idle line interrupt enable bit, ILIE, enables IDLE CPU interrupt requests. IDLE is in SCI status register 1. ILIE is in SCI control register 2.
- Receiver overrun bit (OR) — OR is set when the receive shift register shifts in a new character before the previous character was read from the SCI data register. The overrun interrupt enable bit, ORIE, enables OR to generate SCI error CPU interrupt requests. OR is in SCI status register 1. ORIE is in SCI control register 3.
- Noise flag (NF) — NF is set when the SCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, enables NF to generate SCI error CPU interrupt requests. NF is in SCI status register 1. NEIE is in SCI control register 3.
- Framing error bit (FE) — FE is set when a 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, enables FE to generate SCI error CPU interrupt requests. FE is in SCI status register 1. FEIE is in SCI control register 3.
- Parity error bit (PE) — PE is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, enables PE to generate SCI error CPU interrupt requests. PE is in SCI status register 1. PEIE is in SCI control register 3.

13.3.2.9 $\overline{KBD0}$–$\overline{KBD7}$ Pins

A logic 0 on a keyboard interrupt pin latches an external interrupt request.

13.3.2.10 Analog-to-Digital Converter (ADC)

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

14.4.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 14-3.

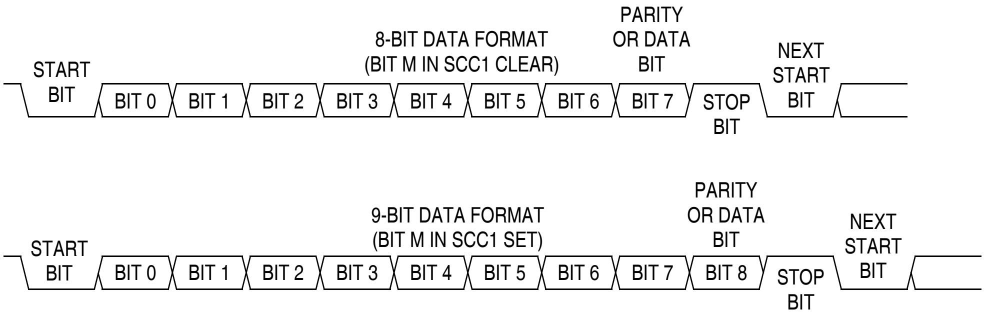

Figure 14-3. SCI Data Formats

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0009	ESCI Prescaler Register (SCPSC) See page 172.	Read:	PDS2	PDS1	PDS0	PSSB4	PSSB3	PSSB2	PSSB1	PSSB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000A	ESCI Arbiter Control Register (SCIACTL) See page 176.	Read:	AM1	ALOST	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
		Write:		▨			▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$000B	ESCI Arbiter Data Register (SCIADAT) See page 177.	Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$0013	ESCI Control Register 1 (SCC1) See page 163.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0014	ESCI Control Register 2 (SCC2) See page 165.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	ESCI Control Register 3 (SCC3) See page 167.	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:	▨							
		Reset:	U	0	0	0	0	0	0	0
$0016	ESCI Status Register 1 (SCS1) See page 168.	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	1	1	0	0	0	0	0	0
$0017	ESCI Status Register 2 (SCS2) See page 170.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$0018	ESCI Data Register (SCDR) See page 171.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0019	ESCI Baud Rate Register (SCBR) See page 171.	Read:	R	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 14-4. ESCI I/O Register Summary

14.8.1 ESCI Control Register 1

ESCI control register 1 (SCC1):

- Enables loop mode operation
- Enables the ESCI
- Controls output polarity
- Controls character length
- Controls ESCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 14-10. ESCI Control Register 1 (SCC1)

LOOPS — Loop Mode Select Bit

This read/write bit enables loop mode operation. In loop mode the RxD pin is disconnected from the ESCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. Reset clears the LOOPS bit.

1 = Loop mode enabled
0 = Normal operation enabled

ENSCI — Enable ESCI Bit

This read/write bit enables the ESCI and the ESCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in ESCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.

1 = ESCI enabled
0 = ESCI disabled

TXINV — Transmit Inversion Bit

This read/write bit reverses the polarity of transmitted data. Reset clears the TXINV bit.

1 = Transmitter output inverted
0 = Transmitter output not inverted

NOTE

Setting the TXINV bit inverts all transmitted values including idle, break, start, and stop bits.

M — Mode (Character Length) Bit

This read/write bit determines whether ESCI characters are eight or nine bits long (See Table 14-5).The ninth bit can serve as a receiver wakeup signal or as a parity bit. Reset clears the M bit.

1 = 9-bit ESCI characters
0 = 8-bit ESCI characters

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

1 = Transmitter enabled
0 = Transmitter disabled

NOTE

Writing to the TE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

NOTE

Writing to the RE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them. Reset clears the SBK bit.

1 = Transmit break characters
0 = No break characters being transmitted

NOTE

Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the ESCI to send a break character instead of a preamble.

16.8 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is 0. Whenever SPE is 0, the following occurs:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.
- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is defaulted back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR register
- All control bits in the SPSCR register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

16.9 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

16.9.1 Wait Mode

The SPI module remains active after the execution of a WAIT instruction. In wait mode the SPI module registers are not accessible by the CPU. Any enabled CPU interrupt request from the SPI module can bring the MCU out of wait mode.

If SPI module functions are not required during wait mode, reduce power consumption by disabling the SPI module before executing the WAIT instruction.

To exit wait mode when an overflow condition occurs, enable the OVRF bit to generate CPU interrupt requests by setting the error interrupt enable bit (ERRIE). See 16.7 Interrupts.

16.9.2 Stop Mode

The SPI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions. SPI operation resumes after an external interrupt. If stop mode is exited by reset, any transfer in progress is aborted, and the SPI is reset.

16.10 SPI During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. BCFE in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. See Chapter 15 System Integration Module (SIM).

To allow software to clear status bits during a break interrupt, write a 1 to BCFE. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

ERRIE — Error Interrupt Enable Bit

This read/write bit enables the MODF and OVRF bits to generate CPU interrupt requests. Reset clears the ERRIE bit.

1 = MODF and OVRF can generate CPU interrupt requests
0 = MODF and OVRF cannot generate CPU interrupt requests

OVRF — Overflow Bit

This clearable, read-only flag is set if software does not read the byte in the receive data register before the next full byte enters the shift register. In an overflow condition, the byte already in the receive data register is unaffected, and the byte that shifted in last is lost. Clear the OVRF bit by reading the SPI status and control register with OVRF set and then reading the receive data register. Reset clears the OVRF bit.

1 = Overflow
0 = No overflow

MODF — Mode Fault Bit

This clearable, read-only flag is set in a slave SPI if the $\overline{SS}$ pin goes high during a transmission with MODFEN set. In a master SPI, the MODF flag is set if the $\overline{SS}$ pin goes low at any time with the MODFEN bit set. Clear MODF by reading the SPI status and control register (SPSCR) with MODF set and then writing to the SPI control register (SPCR). Reset clears the MODF bit.

1 = $\overline{SS}$ pin at inappropriate logic level
0 = $\overline{SS}$ pin at appropriate logic level

SPTE — SPI Transmitter Empty Bit

This clearable, read-only flag is set each time the transmit data register transfers a byte into the shift register. SPTE generates an SPTE CPU interrupt request if SPTIE in the SPI control register is set also.

NOTE

Do not write to the SPI data register unless SPTE is high.

During an SPTE CPU interrupt, the CPU clears SPTE by writing to the transmit data register.

Reset sets the SPTE bit.

1 = Transmit data register empty
0 = Transmit data register not empty

MODFEN — Mode Fault Enable Bit

This read/write bit, when set, allows the MODF flag to be set. If the MODF flag is set, clearing MODFEN does not clear the MODF flag. If the SPI is enabled as a master and the MODFEN bit is 0, then the $\overline{SS}$ pin is available as a general-purpose I/O.

If the MODFEN bit is 1, then the $\overline{SS}$ pin is not available as a general-purpose I/O. When the SPI is enabled as a slave, the $\overline{SS}$ pin is not available as a general-purpose I/O regardless of the value of MODFEN. See 16.11.4 SS (Slave Select).

If the MODFEN bit is 0, the level of the $\overline{SS}$ pin does not affect the operation of an enabled SPI configured as a master. For an enabled SPI configured as a slave, having MODFEN low only prevents the MODF flag from being set. It does not affect any other part of SPI operation. See 16.6.2 Mode Fault Error.

17.6 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

17.6.1 Wait Mode

The timebase module remains active after execution of the WAIT instruction. In wait mode, the timebase register is not accessible by the CPU.

If the timebase functions are not required during wait mode, reduce the power consumption by stopping the timebase before enabling the WAIT instruction.

17.6.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the oscillator has been enabled to operate during stop mode through the OSCSTOPEN bit in the CONFIG register. The timebase module can be used in this mode to generate a periodic wakeup from stop mode.

If the oscillator has not been enabled to operate in stop mode, the timebase module will not be active during STOP mode. In stop mode the timebase register is not accessible by the CPU.

If the timebase functions are not required during stop mode, reduce the power consumption by stopping the timebase before enabling the STOP instruction.

18.6.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

18.7 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. See 15.7.3 SIM Break Flag Control Register.

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

18.8 I/O Signals

Port D shares four of its pins with the TIM. The four TIM channel I/O pins are T1CH0, T1CH1, T2CH0, and T2CH1 as described in 18.3 Pin Name Conventions.

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T1CH0 and T2CH0 can be configured as buffered output compare or buffered PWM pins.

18.9 I/O Registers

NOTE

References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC AND T2SC.

These I/O registers control and monitor operation of the TIM:
- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0, TSC1)
- TIM channel registers (TCH0H:TCH0L, TCH1H:TCH1L)

19.2.2.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. This register is only used in emulation mode.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note[1]	
Reset:							0	

R = Reserved

[1] Writing a 0 clears SBSW.

Figure 19-7. SIM Break Status Register (SBSR)

SBSW — SIM Break Stop/Wait

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

1 = Wait mode was exited by break interrupt
0 = Wait mode was not exited by break interrupt

19.2.2.4 Break Flag Control Register

The SIM break flag control register (SBFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

Address: $FE03

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BCFE	R	R	R	R	R	R	R
Write:								
Reset:	0							

R = Reserved

Figure 19-8. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break
0 = Status bits not clearable during break

If monitor mode was entered with V_{TST} on $\overline{\text{IRQ}}$, then the COP is disabled as long as V_{TST} is applied to either $\overline{\text{IRQ}}$ or $\overline{\text{RST}}$.

This condition states that as long as V_{TST} is maintained on the $\overline{\text{IRQ}}$ pin after entering monitor mode, or if V_{TST} is applied to $\overline{\text{RST}}$ after the initial reset to get into monitor mode (when V_{TST} was applied to $\overline{\text{IRQ}}$), then the COP will be disabled. In the latter situation, after V_{TST} is applied to the $\overline{\text{RST}}$ pin, V_{TST} can be removed from the $\overline{\text{IRQ}}$ pin in the interest of freeing the $\overline{\text{IRQ}}$ for normal functionality in monitor mode.

19.3.1.2 Forced Monitor Mode

If entering monitor mode without high voltage on $\overline{\text{IRQ}}$ (where applied voltage is either V_{DD} or V_{SS}), then all port C pin requirements and conditions, including the PTC3 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

If $\overline{\text{IRQ}} = V_{DD}$ on monitor mode entry, an external oscillator of 9.8304 MHz is required for a 9600 baud rate.

If $\overline{\text{IRQ}} = V_{SS}$ on monitor mode entry, the ICG generates a 9600 baud rate using the trimmed ICG value in the ICGTR5 register. If the ICGTR5 register is blank, the baud rate will be a nominal 9600 baud which may not be adequate for standard PC serial communication.

When forced monitor mode is entered, the COP is always disabled regardless of the state of $\overline{\text{IRQ}}$ or $\overline{\text{RST}}$.

NOTE

If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial POR reset. Once the part has been programmed, the traditional method of applying a voltage, V_{TST}, to $\overline{\text{IRQ}}$ must be used to enter monitor mode.

19.3.1.3 Monitor Vectors

In monitor mode, the MCU uses different vectors for reset, SWI (software interrupt), and break interrupt than those for user mode. The alternate vectors are in the $FE page instead of the $FF page and allow code execution from the internal monitor firmware instead of user code.

NOTE

Exiting monitor mode after it has been initiated by having a blank reset vector requires a power-on reset (POR). Pulling $\overline{\text{RST}}$ low will not exit monitor mode in this situation.

Table 19-2 summarizes the differences between user mode and monitor mode.

Table 19-2. Mode Differences

Modes	Functions					
	Reset Vector High	**Reset Vector Low**	**Break Vector High**	**Break Vector Low**	**SWI Vector High**	**SWI Vector Low**
User	$FFFE	$FFFF	$FFFC	$FFFD	$FFFC	$FFFD
Monitor	$FEFE	$FEFF	$FEFC	$FEFD	$FEFC	$FEFD

$FE20 ↓ $FF7D	MONITOR ROM (MONITOR BLOCK) 350 BYTES
$FF7E	RESERVED
$FF7F	MONITOR ROM (JUMP TABLE) 1 BYTE
$FF80	ICG TRIM REGISTER 5V (ICGTR5) FIX VALUE
$FF81	ICG TRIM REGISTER 3V (ICGTR3) FIX VALUE
$FF82 ↓ $FF96	MONITOR ROM (JUMP TABLE) 21 BYTES
$FF97 ↓ $FFDB	UNIMPLEMENTED 69 BYTES
$FFDC ↓ $FFFF[1]	ROM VECTORS 36 BYTES

[1] $FFF6–$FFFD reserved for eight security bytes

Table A-2. MC68HC08GT16 Memory Map (Continued)

A.5 ICG Trim Registers

The two ICG trim registers at $FF80 and $FF81 are masked with fixed values on the MC68HC08GT16. These are not trim values. These registers are available for compatibility with the MC68HC908GT16 only.

On the MC68HC908GT16, these two trim registers are used to store user trim values.

A.6 Monitor ROM

The monitor program on the MC68HC08GT16 is for device testing only.

A.7 ADC Reference Pins (V_{REFH} and V_{REFL})

V_{REFH} must be connected to the same voltage potential as the analog supply pin, V_{DDA}. V_{REFL} must be connected to the same voltage potential as the analog supply pin, V_{SSA}.

A.8 Electrical Specifications

Electrical specifications for the MC68HC908GT16 apply to the MC68HC08GT16, except for the parameters indicated below.