

Welcome to [E-XFL.COM](https://www.e-xfl.com)

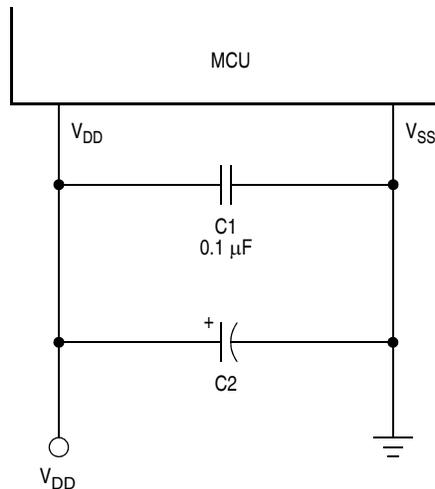
What is "[Embedded - Microcontrollers](#)"?

"[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	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz48vfje



Note: Component values shown represent typical applications.

Figure 1-5. Power Supply Bypassing

1.5.2 Oscillator Pins (OSC1 and OSC2)

OSC1 and OSC2 are the connections for an external crystal, resonator, or clock circuit. See Chapter 4 Clock Generator Module (CGM).

1.5.3 External Reset Pin (\overline{RST})

A low on the \overline{RST} pin forces the MCU to a known startup state. \overline{RST} is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. This pin contains an internal pullup resistor. See Chapter 15 System Integration Module (SIM).

1.5.4 External Interrupt Pin (\overline{IRQ})

\overline{IRQ} is an asynchronous external interrupt pin. This pin contains an internal pullup resistor. See Chapter 8 External Interrupt (IRQ).

1.5.5 CGM Power Supply Pins (V_{DDA} and V_{SSA})

V_{DDA} and V_{SSA} are the power supply pins for the analog portion of the clock generator module (CGM). Decoupling of these pins should be as per the digital supply. See Chapter 4 Clock Generator Module (CGM).

1.5.6 External Filter Capacitor Pin (CGMXFC)

CGMXFC is an external filter capacitor connection for the CGM. See Chapter 4 Clock Generator Module (CGM).

1.5.7 ADC Power Supply/Reference Pins (V_{DDAD}/V_{REFH} and V_{SSAD}/V_{REFL})

V_{DDAD} and V_{SSAD} are the power supply pins to the analog-to-digital converter (ADC). V_{REFH} and V_{REFL} are the reference voltage pins for the ADC. V_{REFH} is the high reference supply for the ADC, and by default the V_{DDAD}/V_{REFH} pin should be externally filtered and connected to the same voltage potential as V_{DD} .

B. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. However, care must be taken to ensure that these operations do not access any address within the FLASH array memory space such as the COP control register (COPCTL) at \$FFFF.

C. It is highly recommended that interrupts be disabled during program/erase operations.

2.6.5 FLASH-1 Page Erase Operation

Use this step-by-step procedure to erase a page (128 bytes) of FLASH-1 memory:

1. Set the ERASE bit and clear the MASS bit in the FLASH-1 control register (FL1CR).
2. Read the FLASH-1 block protect register (FL1BPR).
3. Write any data to any FLASH-1 address within the address range of the page (128 byte block) to be erased.
4. Wait for time, t_{NVS} (minimum 10 μ s).
5. Set the HVEN bit.
6. Wait for time, t_{ERASE} (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for time, t_{NVH} (minimum 5 μ s).
9. Clear the HVEN bit.
10. Wait for a time, t_{RCV} , (typically 1 μ s) after which the memory can be accessed in normal read mode.

NOTES

A. Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.

B. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. However, care must be taken to ensure that these operations do not access any address within the FLASH array memory space such as the COP control register (COPCTL) at \$FFFF.

C. It is highly recommended that interrupts be disabled during program/erase operations.

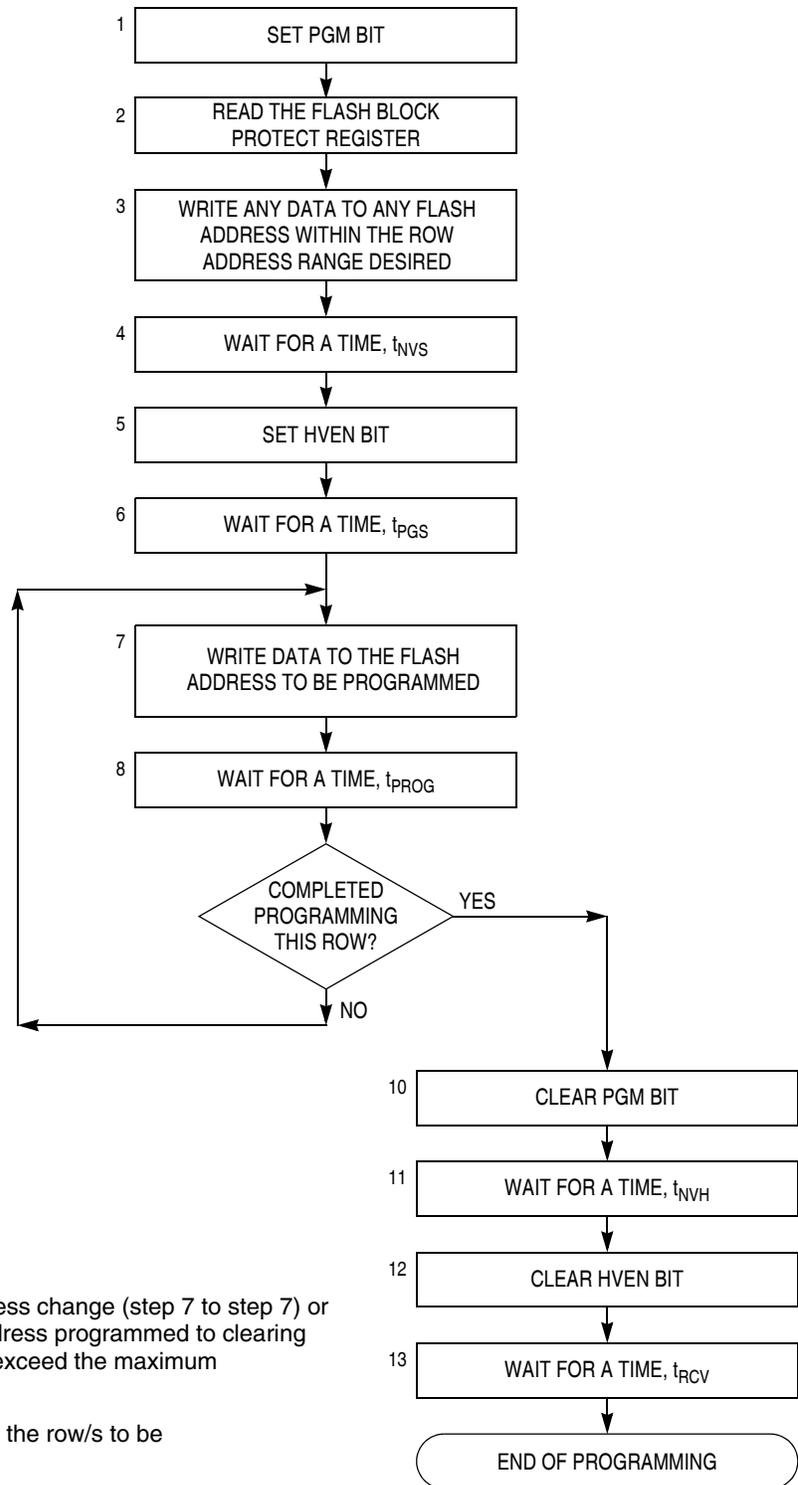
In applications that require more than 1000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a shorter cycle time.

2.6.6 FLASH-1 Program Operation

Programming of the FLASH-1 memory is done on a row basis. A row consists of 64 consecutive bytes with address ranges as follows:

- \$XX00 to \$XX3F
- \$XX40 to \$XX7F
- \$XX80 to \$XXBF
- \$XXC0 to \$XXFF

Algorithm for programming a row (64 bytes) of FLASH memory



NOTES:

The time between each FLASH address change (step 7 to step 7) or the time between the last FLASH address programmed to clearing PGM bit (step 7 to step10) must not exceed the maximum programming time, t_{PROG} , maximum.

This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 2-10. FLASH-2 Programming Algorithm Flowchart

Chapter 3

Analog-to-Digital Converter (ADC)

3.1 Introduction

This section describes the 10-bit analog-to-digital converter (ADC).

3.2 Features

Features of the ADC module include:

- 24 channels with multiplexed input
- Linear successive approximation with monotonicity
- 10-bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock
- Left or right justified result
- Left justified sign data mode

3.3 Functional Description

The ADC provides 24 pins for sampling external sources at pins PTG7/AD23–PTG0/AD16, PTA7/KBD7/AD15–PTA0/KBD0/AD8, and PTB7/AD7–PTB0/AD0. An analog multiplexer allows the single ADC converter to select one of 24 ADC channels as ADC voltage in (V_{ADIN}). V_{ADIN} is converted by the successive approximation register-based analog-to-digital converter. When the conversion is completed, ADC places the result in the ADC data register and sets a flag or generates an interrupt. See Figure 3-2.

3.3.1 ADC Port I/O Pins

PTG7/AD23–PTG0/AD16, PTA7/KBD7/AD15–PTA0/KBD0/AD8, and PTB7/AD7–PTB0/AD0 are general-purpose I/O (input/output) pins that share with the ADC channels. The channel select bits define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or data direction register (DDR) will not have any affect on the port pin that is selected by the ADC. A read of a port pin in use by the ADC will return a 0.

Left justified sign data mode is similar to left justified mode with one exception. The MSB of the 10-bit result, AD9 located in ADRH, is complemented. This mode of operation is useful when a result, represented as a signed magnitude from mid-scale, is needed. Finally, 8-bit truncation mode will place the eight MSBs in the ADC data register low, ADRL. The two LSBs are dropped. This mode of operation is used when compatibility with 8-bit ADC designs are required. No interlocking between ADRH and ADRL is present.

NOTE

Quantization error is affected when only the most significant eight bits are used as a result. See Figure 3-3.

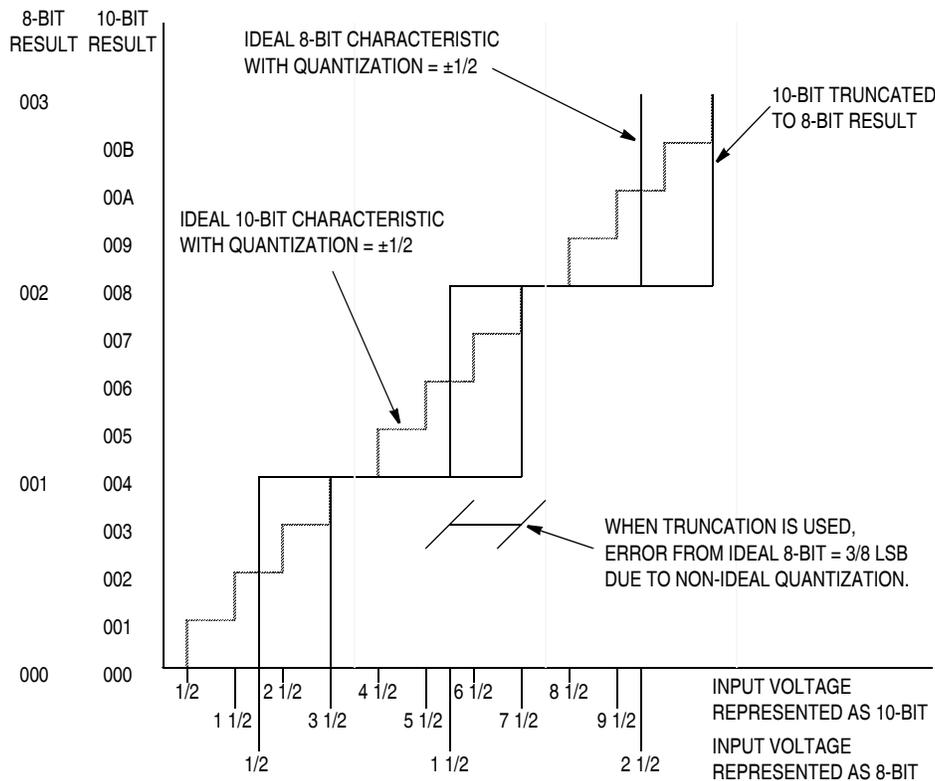


Figure 3-3. Bit Truncation Mode Error

3.4 Monotonicity

The conversion process is monotonic and has no missing codes.

3.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating CPU interrupts after each ADC conversion. A CPU interrupt is generated if the COCO bit is a 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

Clock Generator Module (CGM)

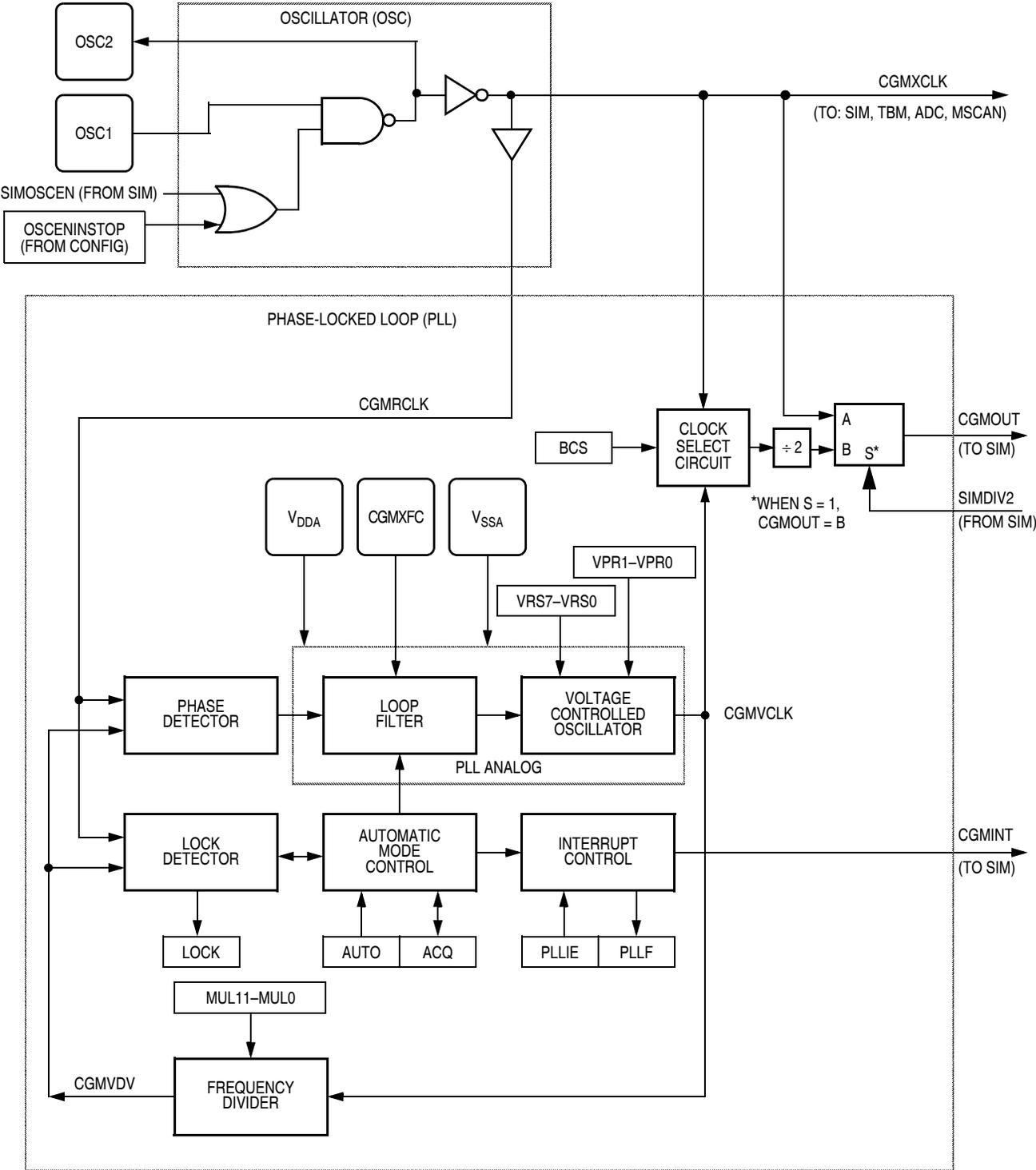


Figure 4-1. CGM Block Diagram

4.3.9 CGM External Connections

In its typical configuration, the CGM requires external components. Five of these are for the crystal oscillator and two or four are for the PLL.

The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in Figure 4-2. Figure 4-2 shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

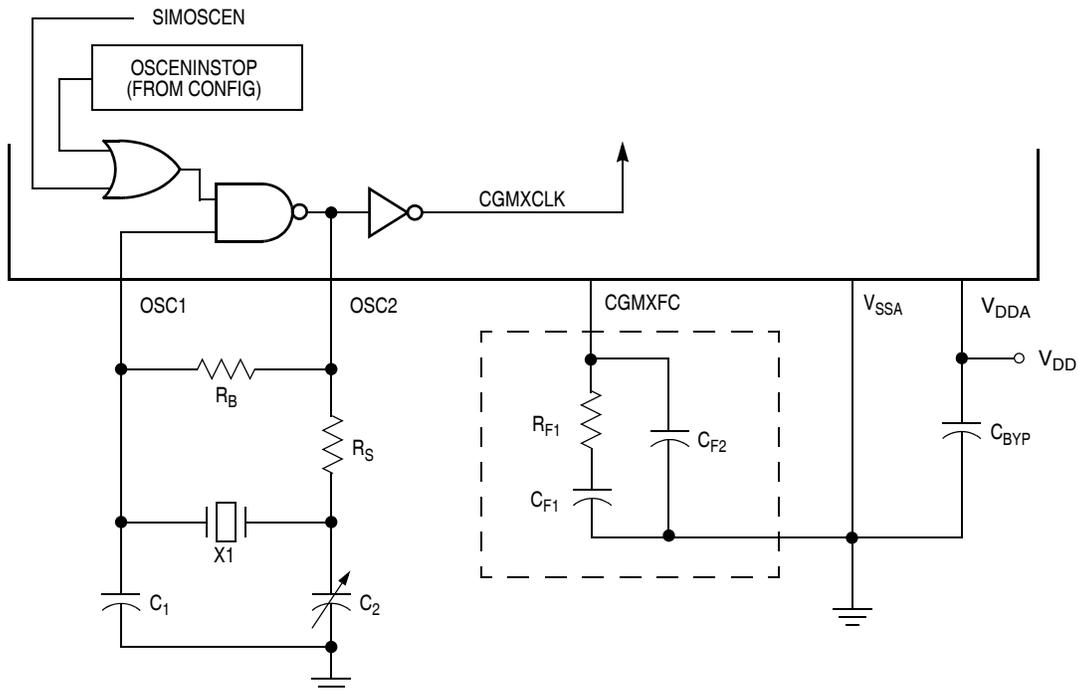
- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines. Refer to the crystal manufacturer's data for more information regarding values for C_1 and C_2 .

Figure 4-2 also shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter network

Routing should be done with great care to minimize signal cross talk and noise.



Note: Filter network in box can be replaced with a single capacitor, but will degrade stability.

Figure 4-2. CGM External Connections

4.6 Interrupts

When the AUTO bit is set in the PLL bandwidth control register (PBWC), the PLL can generate a CPU interrupt request every time the LOCK bit changes state. The PLLIE bit in the PLL control register (PCTL) enables CPU interrupts from the PLL. PLLF, the interrupt flag in the PCTL, becomes set whether interrupts are enabled or not. When the AUTO bit is clear, CPU interrupts from the PLL are disabled and PLLF reads as 0.

Software should read the LOCK bit after a PLL interrupt request to see if the request was due to an entry into lock or an exit from lock. When the PLL enters lock, the VCO clock, CGMVCLK, divided by two can be selected as the CGMOUT source by setting BCS in the PCTL. When the PLL exits lock, the VCO clock frequency is corrupt, and appropriate precautions should be taken. If the application is not frequency sensitive, interrupts should be disabled to prevent PLL interrupt service routines from impeding software performance or from exceeding stack limitations.

NOTE

Software can select the CGMVCLK divided by two as the CGMOUT source even if the PLL is not locked (LOCK = 0). Therefore, software should make sure the PLL is locked before setting the BCS bit.

4.7 Special Modes

The WAIT instruction puts the MCU in low power-consumption standby modes.

4.7.1 Wait Mode

The WAIT instruction does not affect the CGM. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL) to save power. Less power-sensitive applications can disengage the PLL without turning it off, so that the PLL clock is immediately available at WAIT exit. This would be the case also when the PLL is to wake the MCU from wait mode, such as when the PLL is first enabled and waiting for LOCK or LOCK is lost.

4.7.2 Stop Mode

If the OSCENINSTOP bit in the CONFIG2 register is cleared (default), then the STOP instruction disables the CGM (oscillator and phase locked loop) and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If the OSCENINSTOP bit in the CONFIG2 register is set, then the phase locked loop is shut off but the oscillator will continue to operate in stop mode.

4.7.3 CGM During Break Interrupts

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 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 the PLLF bit during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write the PLL control register during the break state without affecting the PLLF bit.

7.7 Instruction Set Summary

Table 7-1 provides a summary of the M68HC08 instruction set.

Table 7-1. Instruction Set Summary (Sheet 1 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles	
			V	H	I	N	Z					C
ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X ADC opr,SP ADC opr,SP	Add with Carry	$A \leftarrow (A) + (M) + (C)$	†	†	-	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd ll hh ll ee ff ff ff ff ee	2 3 4 4 3 2 4 5
ADD #opr ADD opr ADD opr ADD opr,X ADD opr,X ADD ,X ADD opr,SP ADD opr,SP	Add without Carry	$A \leftarrow (A) + (M)$	†	†	-	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ff ee	2 3 4 4 3 2 4 5
AIS #opr	Add Immediate Value (Signed) to SP	$SP \leftarrow (SP) + (16 \ll M)$	-	-	-	-	-	-	IMM	A7	ii	2
AIX #opr	Add Immediate Value (Signed) to H:X	$H:X \leftarrow (H:X) + (16 \ll M)$	-	-	-	-	-	-	IMM	AF	ii	2
AND #opr AND opr AND opr AND opr,X AND opr,X AND ,X AND opr,SP AND opr,SP	Logical AND	$A \leftarrow (A) \& (M)$	0	-	-	†	†	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ff ee	2 3 4 4 3 2 4 5
ASL opr ASLA ASLX ASL opr,X ASL ,X ASL opr,SP	Arithmetic Shift Left (Same as LSL)		†	-	-	†	†	†	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR opr ASRA ASRX ASR opr,X ASR opr,X ASR opr,SP	Arithmetic Shift Right		†	-	-	†	†	†	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC rel	Branch if Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (C) = 0$	-	-	-	-	-	-	REL	24	rr	3
BCLR n, opr	Clear Bit n in M	$M_n \leftarrow 0$	-	-	-	-	-	-	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BCS rel	Branch if Carry Bit Set (Same as BLO)	$PC \leftarrow (PC) + 2 + rel ? (C) = 1$	-	-	-	-	-	-	REL	25	rr	3
BEQ rel	Branch if Equal	$PC \leftarrow (PC) + 2 + rel ? (Z) = 1$	-	-	-	-	-	-	REL	27	rr	3
BGE opr	Branch if Greater Than or Equal To (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 0$	-	-	-	-	-	-	REL	90	rr	3
BGT opr	Branch if Greater Than (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (Z) \mid (N \oplus V) = 0$	-	-	-	-	-	-	REL	92	rr	3
BHCC rel	Branch if Half Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (H) = 0$	-	-	-	-	-	-	REL	28	rr	3
BHCS rel	Branch if Half Carry Bit Set	$PC \leftarrow (PC) + 2 + rel ? (H) = 1$	-	-	-	-	-	-	REL	29	rr	3
BHI rel	Branch if Higher	$PC \leftarrow (PC) + 2 + rel ? (C) \mid (Z) = 0$	-	-	-	-	-	-	REL	22	rr	3

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

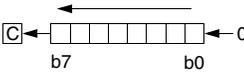
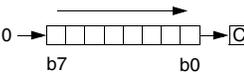
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr,X</i> JMP <i>opr,X</i> JMP ,X	Jump	PC ← Jump Address	-	-	-	-	-	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr,X</i> JSR <i>opr,X</i> JSR ,X	Jump to Subroutine	PC ← (PC) + <i>n</i> (<i>n</i> = 1, 2, or 3) Push (PCL); SP ← (SP) - 1 Push (PCH); SP ← (SP) - 1 PC ← Unconditional Address	-	-	-	-	-	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr,X</i> LDA <i>opr,X</i> LDA ,X LDA <i>opr,SP</i> LDA <i>opr,SP</i>	Load A from M	A ← (M)	0	-	-	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ee ff ff ee ff	2 3 4 4 3 2 4 5
LDHX # <i>opr</i> LDHX <i>opr</i>	Load H:X from M	H:X ← (M:M + 1)	0	-	-	†	†	IMM DIR	45 55	ii jj dd	3 4
LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr,X</i> LDX <i>opr,X</i> LDX ,X LDX <i>opr,SP</i> LDX <i>opr,SP</i>	Load X from M	X ← (M)	0	-	-	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL <i>opr</i> LSLA LSLX LSL <i>opr,X</i> LSL ,X LSL <i>opr,SP</i>	Logical Shift Left (Same as ASL)		†	-	-	†	†	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR <i>opr</i> LSRA LSRX LSR <i>opr,X</i> LSR ,X LSR <i>opr,SP</i>	Logical Shift Right		†	-	-	0	†	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5
MOV <i>opr,opr</i> MOV <i>opr,X+</i> MOV # <i>opr,opr</i> MOV X+, <i>opr</i>	Move	(M) _{Destination} ← (M) _{Source} H:X ← (H:X) + 1 (IX+D, DIX+)	0	-	-	†	†	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	-	0	-	-	0	INH	42		5
NEG <i>opr</i> NEGA NEGX NEG <i>opr,X</i> NEG ,X NEG <i>opr,SP</i>	Negate (Two's Complement)	M ← -(M) = \$00 - (M) A ← -(A) = \$00 - (A) X ← -(X) = \$00 - (X) M ← -(M) = \$00 - (M) M ← -(M) = \$00 - (M)	†	-	-	†	†	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	4 1 1 4 3 5
NOP	No Operation	None	-	-	-	-	-	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	-	-	-	-	-	INH	62		3
ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr,X</i> ORA <i>opr,X</i> ORA ,X ORA <i>opr,SP</i> ORA <i>opr,SP</i>	Inclusive OR A and M	A ← (A) (M)	0	-	-	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) - 1	-	-	-	-	-	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) - 1	-	-	-	-	-	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) - 1	-	-	-	-	-	INH	89		2

Table 7-2. Opcode Map

MSB LSB	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory							
	DIR	DIR	REL	DIR	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
	0	1	2	3	4	5	6	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
0	BRSET0 3 DIR	BSET0 2 DIR	BRA 2 REL	NEG 2 DIR	NEGA 1 INH	NEGX 1 INH	NEG 2 IX1	NEG 3 SP1	NEG 1 IX	RTI 1 INH	BGE 2 REL	SUB 2 IMM	SUB 2 DIR	SUB 3 EXT	SUB 3 IX2	SUB 4 SP2	SUB 2 IX1	SUB 3 SP1	SUB 1 IX
1	BRCLR0 3 DIR	BCLR0 2 DIR	BRN 2 REL	CBEQ 3 DIR	CBEQA 3 IMM	CBEQX 3 IMM	CBEQ 3 IX1+	CBEQ 4 SP1	CBEQ 2 IX+	RTS 1 INH	BLT 2 REL	CMP 2 IMM	CMP 2 DIR	CMP 3 EXT	CMP 3 IX2	CMP 4 SP2	CMP 2 IX1	CMP 3 SP1	CMP 1 IX
2	BRSET1 3 DIR	BSET1 2 DIR	BHI 2 REL		MUL 1 INH	DIV 1 INH	NSA 1 INH		DAA 1 INH		BGT 2 REL	SBC 2 IMM	SBC 2 DIR	SBC 3 EXT	SBC 3 IX2	SBC 4 SP2	SBC 2 IX1	SBC 3 SP1	SBC 1 IX
3	BRCLR1 3 DIR	BCLR1 2 DIR	BLS 2 REL	COM 2 DIR	COMA 1 INH	COMX 1 INH	COM 2 IX1	COM 3 SP1	COM 1 IX	SWI 1 INH	BLE 2 REL	CPX 2 IMM	CPX 2 DIR	CPX 3 EXT	CPX 3 IX2	CPX 4 SP2	CPX 2 IX1	CPX 3 SP1	CPX 1 IX
4	BRSET2 3 DIR	BSET2 2 DIR	BCC 2 REL	LSR 2 DIR	LSRA 1 INH	LSRX 1 INH	LSR 2 IX1	LSR 3 SP1	LSR 1 IX	TAP 1 INH	TXS 1 INH	AND 2 IMM	AND 2 DIR	AND 3 EXT	AND 3 IX2	AND 4 SP2	AND 2 IX1	AND 3 SP1	AND 1 IX
5	BRCLR2 3 DIR	BCLR2 2 DIR	BCS 2 REL	STHX 2 DIR	LDHX 3 IMM	LDHX 2 DIR	CPHX 3 IMM		CPHX 2 DIR	TPA 1 INH	TSX 1 INH	BIT 2 IMM	BIT 2 DIR	BIT 3 EXT	BIT 3 IX2	BIT 4 SP2	BIT 2 IX1	BIT 3 SP1	BIT 1 IX
6	BRSET3 3 DIR	BSET3 2 DIR	BNE 2 REL	ROR 2 DIR	RORA 1 INH	RORX 1 INH	ROR 2 IX1	ROR 3 SP1	ROR 1 IX	PULA 1 INH		LDA 2 IMM	LDA 2 DIR	LDA 3 EXT	LDA 3 IX2	LDA 4 SP2	LDA 2 IX1	LDA 3 SP1	LDA 1 IX
7	BRCLR3 3 DIR	BCLR3 2 DIR	BEQ 2 REL	ASR 2 DIR	ASRA 1 INH	ASRX 1 INH	ASR 2 IX1	ASR 3 SP1	ASR 1 IX	PSHA 1 INH	TAX 1 INH	AIS 2 IMM	STA 2 DIR	STA 3 EXT	STA 3 IX2	STA 4 SP2	STA 2 IX1	STA 3 SP1	STA 1 IX
8	BRSET4 3 DIR	BSET4 2 DIR	BHCC 2 REL	LSL 2 DIR	LSLA 1 INH	LSLX 1 INH	LSL 2 IX1	LSL 3 SP1	LSL 1 IX	PULX 1 INH	CLC 1 INH	EOR 2 IMM	EOR 2 DIR	EOR 3 EXT	EOR 3 IX2	EOR 4 SP2	EOR 2 IX1	EOR 3 SP1	EOR 1 IX
9	BRCLR4 3 DIR	BCLR4 2 DIR	BHCS 2 REL	ROL 2 DIR	ROLA 1 INH	ROLX 1 INH	ROL 2 IX1	ROL 3 SP1	ROL 1 IX	PSHX 1 INH	SEC 1 INH	ADC 2 IMM	ADC 2 DIR	ADC 3 EXT	ADC 3 IX2	ADC 4 SP2	ADC 2 IX1	ADC 3 SP1	ADC 1 IX
A	BRSET5 3 DIR	BSET5 2 DIR	BPL 2 REL	DEC 2 DIR	DECA 1 INH	DECX 1 INH	DEC 2 IX1	DEC 3 SP1	DEC 1 IX	PULH 1 INH	CLI 1 INH	ORA 2 IMM	ORA 2 DIR	ORA 3 EXT	ORA 3 IX2	ORA 4 SP2	ORA 2 IX1	ORA 3 SP1	ORA 1 IX
B	BRCLR5 3 DIR	BCLR5 2 DIR	BMI 2 REL	DBNZ 3 DIR	DBNZA 2 INH	DBNZX 2 INH	DBNZ 3 IX1	DBNZ 4 SP1	DBNZ 2 IX	PSHH 1 INH	SEI 1 INH	ADD 2 IMM	ADD 2 DIR	ADD 3 EXT	ADD 3 IX2	ADD 4 SP2	ADD 2 IX1	ADD 3 SP1	ADD 1 IX
C	BRSET6 3 DIR	BSET6 2 DIR	BMC 2 REL	INC 2 DIR	INCA 1 INH	INCX 1 INH	INC 2 IX1	INC 3 SP1	INC 1 IX	CLRH 1 INH	RSP 1 INH		JMP 2 DIR	JMP 3 EXT	JMP 3 IX2		JMP 2 IX1		JMP 1 IX
D	BRCLR6 3 DIR	BCLR6 2 DIR	BMS 2 REL	TST 2 DIR	TSTA 1 INH	TSTX 1 INH	TST 2 IX1	TST 3 SP1	TST 1 IX		NOP 1 INH	BSR 2 REL	JSR 2 DIR	JSR 3 EXT	JSR 3 IX2		JSR 2 IX1		JSR 1 IX
E	BRSET7 3 DIR	BSET7 2 DIR	BIL 2 REL		MOV 3 DD	MOV 2 DIX+	MOV 3 IMD		MOV 2 IX+D	STOP 1 INH	*	LDX 2 IMM	LDX 2 DIR	LDX 3 EXT	LDX 3 IX2	LDX 4 SP2	LDX 2 IX1	LDX 3 SP1	LDX 1 IX
F	BRCLR7 3 DIR	BCLR7 2 DIR	BIH 2 REL	CLR 2 DIR	CLRA 1 INH	CLRX 1 INH	CLR 2 IX1	CLR 3 SP1	CLR 1 IX	WAIT 1 INH	TXA 1 INH	AIX 2 IMM	STX 2 DIR	STX 3 EXT	STX 3 IX2	STX 4 SP2	STX 2 IX1	STX 3 SP1	STX 1 IX

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

*Pre-byte for stack pointer indexed instructions

Low Byte of Opcode in Hexadecimal

MSB	0
LSB	5 BRSET0 3 DIR

High Byte of Opcode in Hexadecimal
 Cycles
 Opcode Mnemonic
 Number of Bytes / Addressing Mode

MSCAN08 Controller (MSCAN08)

CLKSRC — Clock Source

This flag defines which clock source the MSCAN08 module is driven from (see 12.10 Clock System).

1 = The MSCAN08 clock source is CGMOUT (see Figure 12-8).

0 = The MSCAN08 clock source is CGMXCLK/2 (see Figure 12-8).

NOTE

The CMCR1 register can be written only if the SFTRES bit in the MSCAN08 module control register is set

12.13.3 MSCAN08 Bus Timing Register 0

Address: \$0502

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 12-18. Bus Timing Register 0 (CBTR0)

SJW1 and SJW0 — Synchronization Jump Width

The synchronization jump width (SJW) defines the maximum number of time quanta (T_q) clock cycles by which a bit may be shortened, or lengthened, to achieve resynchronization on data transitions on the bus (see Table 12-6).

Table 12-6. Synchronization Jump Width

SJW1	SJW0	Synchronization Jump Width
0	0	1 T_q cycle
0	1	2 T_q cycle
1	0	3 T_q cycle
1	1	4 T_q cycle

BRP5–BRP0 — Baud Rate Prescaler

These bits determine the time quanta (T_q) clock, which is used to build up the individual bit timing, according to Table 12-7.

Table 12-7. Baud Rate Prescaler

BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	Prescaler Value (P)
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
:	:	:	:	:	:	:
1	1	1	1	1	1	64

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch), or when the receiver detects an idle character. Polling RPF before disabling the ESCI module or entering stop mode can show whether a reception is in progress.

- 1 = Reception in progress
- 0 = No reception in progress

14.8.6 ESCI Data Register

The ESCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the ESCI data register.

Address: \$0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by reset							

Figure 14-16. ESCI Data Register (SCDR)

R7/T7:R0/T0 — Receive/Transmit Data Bits

Reading address \$0018 accesses the read-only received data bits, R7:R0. Writing to address \$0018 writes the data to be transmitted, T7:T0. Reset has no effect on the ESCI data register.

NOTE

Do not use read-modify-write instructions on the ESCI data register.

14.8.7 ESCI Baud Rate Register

The ESCI baud rate register (SCBR) together with the ESCI prescaler register selects the baud rate for both the receiver and the transmitter.

NOTE

There are two prescalers available to adjust the baud rate. One in the ESCI baud rate register and one in the ESCI prescaler register.

Address: \$0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LINT	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
Write:								
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-17. ESCI Baud Rate Register (SCBR)

LINT — LIN Transmit Enable

This read/write bit selects the enhanced ESCI features for the local interconnect network (LIN) protocol as shown in Table 14-6. Reset clears LINT.

ARUN— Arbiter Counter Running Flag

This read-only bit indicates the arbiter counter is running. Reset clears ARUN.

- 1 = Arbiter counter running
- 0 = Arbiter counter stopped

AROVFL— Arbiter Counter Overflow Bit

This read-only bit indicates an arbiter counter overflow. Clear AROVFL by writing any value to SCIACTL. Writing 0s to AM1 and AM0 resets the counter keeps it in this idle state. Reset clears AROVFL.

- 1 = Arbiter counter overflow has occurred
- 0 = No arbiter counter overflow has occurred

ARD8— Arbiter Counter MSB

This read-only bit is the MSB of the 9-bit arbiter counter. Clear ARD8 by writing any value to SCIACTL. Reset clears ARD8.

14.9.2 ESCI Arbiter Data Register

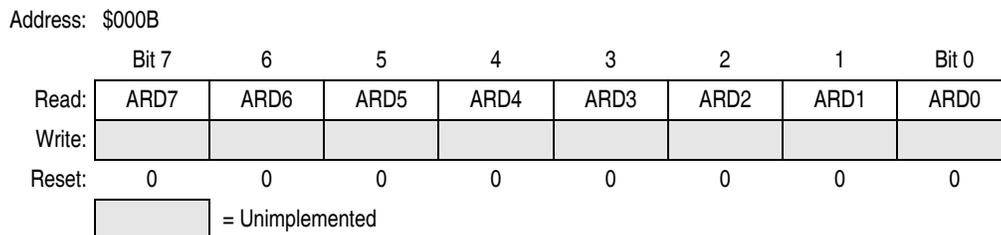


Figure 14-20. ESCI Arbiter Data Register (SCIADAT)

ARD7–ARD0 — Arbiter Least Significant Counter Bits

These read-only bits are the eight LSBs of the 9-bit arbiter counter. Clear ARD7–ARD0 by writing any value to SCIACTL. Writing 0s to AM1 and AM0 permanently resets the counter and keeps it in this idle state. Reset clears ARD7–ARD0.

14.9.3 Bit Time Measurement

Two bit time measurement modes, described here, are available according to the state of ACLK.

1. **ACLK = 0** — The counter is clocked with the bus clock divided by four. The counter is started when a falling edge on the RxD pin is detected. The counter will be stopped on the next falling edge. ARUN is set while the counter is running, AFIN is set on the second falling edge on RxD (for instance, the counter is stopped). This mode is used to recover the received baud rate. See Figure 14-21.
2. **ACLK = 1** — The counter is clocked with one half of the ESCI input clock generated by the ESCI prescaler. The counter is started when a 0 is detected on RxD (see Figure 14-22). A 0 on RxD on enabling the bit time measurement with ACLK = 1 leads to immediate start of the counter (see Figure 14-23). The counter will be stopped on the next rising edge of RxD. This mode is used to measure the length of a received break.

14.9.4 Arbitration Mode

If AM[1:0] is set to 10, the arbiter module operates in arbitration mode. On every rising edge of SCI_TxD (output of the ESCI module, internal chip signal), the counter is started. When the counter reaches \$38 (ACLK = 0) or \$08 (ACLK = 1), RxD is statically sensed. If in this case, RxD is sensed low (for example,

Serial Peripheral Interface (SPI) Module

input (\overline{SS}) is low, so that only the selected slave drives to the master. The \overline{SS} pin of the master is not shown but is assumed to be inactive. The \overline{SS} pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 16.6.2 Mode Fault Error.) When $CPHA = 0$, the first SPSCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCK edge, and a falling edge on the \overline{SS} pin is used to start the slave data transmission. The slave's \overline{SS} pin must be toggled back to high and then low again between each byte transmitted as shown in Figure 16-6.

When $CPHA = 0$ for a slave, the falling edge of \overline{SS} indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data. Once the transmission begins, no new data is allowed into the shift register from the transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the falling edge of \overline{SS} . Any data written after the falling edge is stored in the transmit data register and transferred to the shift register after the current transmission.

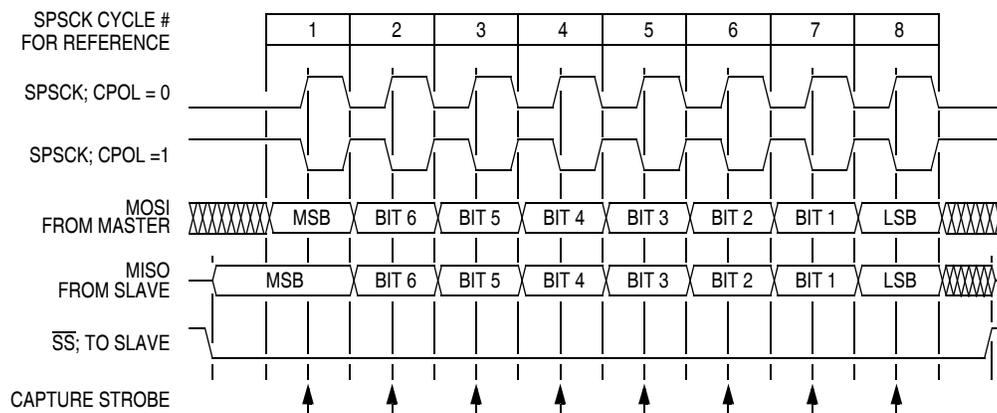


Figure 16-5. Transmission Format (CPHA = 0)

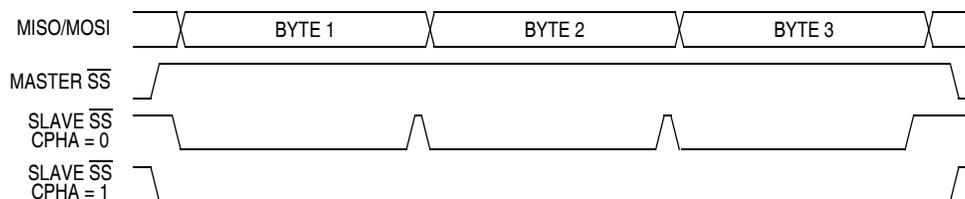


Figure 16-6. CPHA/ \overline{SS} Timing

16.4.3 Transmission Format When $CPHA = 1$

Figure 16-7 shows an SPI transmission in which $CPHA = 1$. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SPSCK: one for $CPOL = 0$ and another for $CPOL = 1$. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SPSCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The \overline{SS} line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input (\overline{SS}) is low, so that only the selected slave drives to the master. The \overline{SS} pin of the master is not shown but is assumed to be inactive. The \overline{SS}

Table 16-2. SPI Configuration

SPE	SPMSTR	MODFEN	SPI Configuration	Function of \overline{SS} Pin
0	X ⁽¹⁾	X	Not enabled	General-purpose I/O; \overline{SS} ignored by SPI
1	0	X	Slave	Input-only to SPI
1	1	0	Master without MODF	General-purpose I/O; \overline{SS} ignored by SPI
1	1	1	Master with MODF	Input-only to SPI

1. X = Don't care

16.12 I/O Registers

Three registers control and monitor SPI operation:

- SPI control register (SPCR)
- SPI status and control register (SPSCR)
- SPI data register (SPDR)

16.12.1 SPI Control Register

The SPI control register:

- Enables SPI module interrupt requests
- Configures the SPI module as master or slave
- Selects serial clock polarity and phase
- Configures the SPSCK, MOSI, and MISO pins as open-drain outputs
- Enables the SPI module

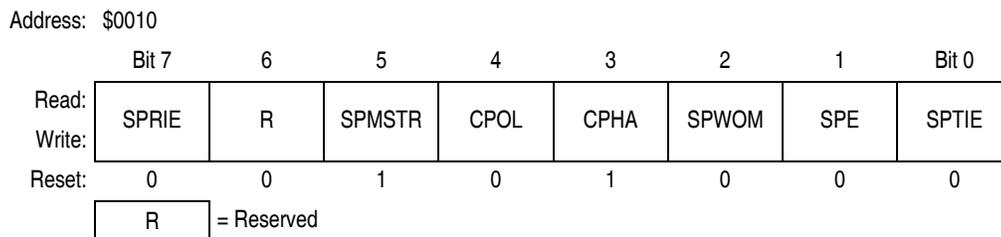


Figure 16-14. SPI Control Register (SPCR)

SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables CPU interrupt requests generated by the SPRF bit. The SPRF bit is set when a byte transfers from the shift register to the receive data register. Reset clears the SPRIE bit.

- 1 = SPRF CPU interrupt requests enabled
- 0 = SPRF CPU interrupt requests disabled

SPMSTR — SPI Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset sets the SPMSTR bit.

- 1 = Master mode
- 0 = Slave mode

20.3.1.4 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.



Figure 20-12. Monitor Data Format

20.3.1.5 Break Signal

A start bit (0) followed by nine 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of approximately two bits and then echoes back the break signal.

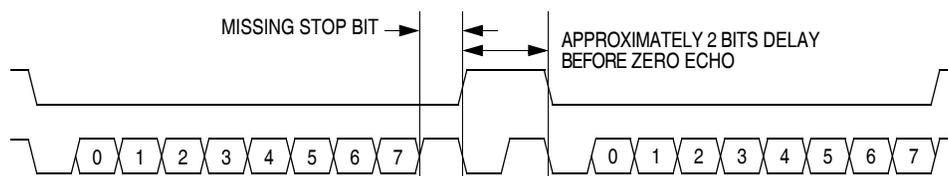


Figure 20-13. Break Transaction

20.3.1.6 Baud Rate

The communication baud rate is controlled by the crystal frequency or external clock and the state of the PTB4 pin (when \overline{IRQ} is set to V_{TST}) upon entry into monitor mode. If monitor mode was entered with V_{DD} on \overline{IRQ} and the reset vector blank, then the baud rate is independent of PTB4.

Table 20-1 also lists external frequencies required to achieve a standard baud rate of 7200 bps. The effective baud rate is the bus frequency divided by 278. If using a crystal as the clock source, be aware of the upper frequency limit that the internal clock module can handle. See 21.7 5.0-Volt Control Timing or 21.8 3.3-Volt Control Timing for this limit.

20.3.1.7 Commands

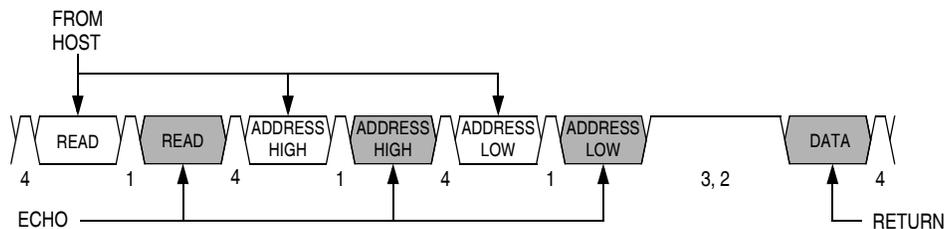
The monitor ROM firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

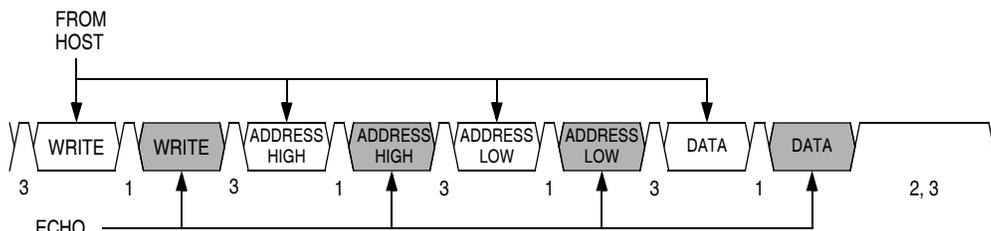
NOTE

Wait one bit time after each echo before sending the next byte.



- Notes:
- 1 = Echo delay, approximately 2 bit times
 - 2 = Data return delay, approximately 2 bit times
 - 3 = Cancel command delay, 11 bit times
 - 4 = Wait 1 bit time before sending next byte.

Figure 20-14. Read Transaction



- Notes:
- 1 = Echo delay, approximately 2 bit times
 - 2 = Cancel command delay, 11 bit times
 - 3 = Wait 1 bit time before sending next byte.

Figure 20-15. Write Transaction

A brief description of each monitor mode command is given in Table 20-3 through Table 20-8.

Table 20-3. READ (Read Memory) Command

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	\$4A
Command Sequence	



NOTES:

1. DIMENSIONS AND TOLERANCING PER ASME Y14.5M-1994.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE AB IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS T, U, AND Z TO BE DETERMINED AT DATUM PLANE AB.



5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE AC.



6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE AB.



7. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.350.

8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076.



9. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: LQFP, 48 LEAD, 0.50 PITCH (7.0 X 7.0 X 1.4)	DOCUMENT NO: 98ASH00962A	REV: G	
	CASE NUMBER: 932-03	14 APR 2005	
	STANDARD: JEDEC MS-026-BBC		