



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	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908gz16vfa

MC68HC908GZ16

MC68HC908GZ8

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://freescale.com/>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
This product incorporates SuperFlash® technology licensed from SST.

© Freescale Semiconductor, Inc., 2005, 2006. All rights reserved.

18.4	Interrupts	251
18.5	TBM Interrupt Rate	252
18.6	Low-Power Modes	253
18.6.1	Wait Mode	253
18.6.2	Stop Mode	253
18.7	Timebase Control Register	254

Chapter 19

Timer Interface Module (TIM)

19.1	Introduction	255
19.2	Features	257
19.3	Pin Name Conventions	257
19.4	Functional Description	257
19.4.1	TIM Counter Prescaler	259
19.4.2	Input Capture	259
19.4.3	Output Compare	260
19.4.3.1	Unbuffered Output Compare	260
19.4.3.2	Buffered Output Compare	260
19.4.4	Pulse Width Modulation (PWM)	261
19.4.4.1	Unbuffered PWM Signal Generation	261
19.4.4.2	Buffered PWM Signal Generation	262
19.4.4.3	PWM Initialization	262
19.5	Interrupts	263
19.6	Low-Power Modes	263
19.6.1	Wait Mode	263
19.6.2	Stop Mode	264
19.7	TIM During Break Interrupts	264
19.8	I/O Signals	264
19.9	I/O Registers	264
19.9.1	TIM Status and Control Register	265
19.9.2	TIM Counter Registers	266
19.9.3	TIM Counter Modulo Registers	267
19.9.4	TIM Channel Status and Control Registers	267
19.9.5	TIM Channel Registers	270

Chapter 20

Development Support

20.1	Introduction	271
20.2	Break Module (BRK)	271
20.2.1	Functional Description	271
20.2.1.1	Flag Protection During Break Interrupts	273
20.2.1.2	TIM During Break Interrupts	273
20.2.1.3	COP During Break Interrupts	273
20.2.2	Break Module Registers	273
20.2.2.1	Break Status and Control Register	274
20.2.2.2	Break Address Registers	274
20.2.2.3	Break Status Register	275
20.2.2.4	Break Flag Control Register	275

General Description

These port pins also have selectable pullups when configured for input mode. The pullups are disengaged when configured for output mode. The pullups are selectable on an individual port bit basis.

1.5.12 Port E I/O Pins (PTE5–PTE2, PTE1/RxD, and PTE0/TxD)

PTE5–PTE0 are general-purpose, bidirectional I/O port pins. PTE1 and PTE0 can also be programmed to be enhanced serial communications interface (ESCI) pins. PTE5–PTE2 are only available on the 48-pin LQFP package. See Chapter 15 Enhanced Serial Communications Interface (ESCI) Module and Chapter 13 Input/Output (I/O) Ports.

1.6 Unused Pin Termination

Input pins and I/O port pins that are not used in the application must be terminated. This prevents excess current caused by floating inputs, and enhances immunity during noise or transient events. Termination methods include:

1. Configuring unused pins as outputs and driving high or low;
2. Configuring unused pins as inputs and enabling internal pull-ups;
3. Configuring unused pins as inputs and using external pull-up or pull-down resistors.

Never connect unused pins directly to V_{DD} or V_{SS} .

Since some general-purpose I/O pins are not available on all packages, these pins must be terminated as well. Either method 1 or 2 above are appropriate.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0560 ↓ \$056F \$0570 ↓ \$057F	MSCAN08 Transmit Buffer 1 See page 137.	MSC08 transmitter buffer 1 Refer to 12.12 Programmer's Model of Message Storage							
	MSCAN08 Transmit Buffer 2 See page 137.	MSC08 transmitter buffer 2 Refer to 12.12 Programmer's Model of Message Storage							
\$FE00	Break Status Register (BSR) See page 275.	Read:	R	R	R	R	R	SBSW	R
		(Note 1)							
		Write:							
		Reset:	0	0	0	0	0	0	0
1. Writing a logic 0 clears SBSW.									
\$FE01	SIM Reset Status Register (SRSR) See page 228.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI
		Write:							
		POR:	1	0	0	0	0	0	0
\$FE02	Reserved	Read:	R	R	R	R	R	R	R
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR) See page 275.	Read:	BCFE	R	R	R	R	R	R
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE04	Interrupt Status Register 1 (INT1) See page 180.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0
		Write:	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2) See page 180.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8
		Write:	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3) See page 180.	Read:	0	0	IF20	IF19	IF18	IF17	IF16
		Write:	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0
\$FE07	Reserved	Read:	R	R	R	R	R	R	R
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE08	FLASH Control Register (FLCR) See page 39.	Read:	0	0	0	0	HVEN	MASS	ERASE
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE09	Break Address Register High (BRKH) See page 274.	Read:	Bit 15	14	13	12	11	10	9
		Write:							Bit 8
		Reset:	0	0	0	0	0	0	0
			= Unimplemented		R = Reserved		U = Unaffected		

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 8)

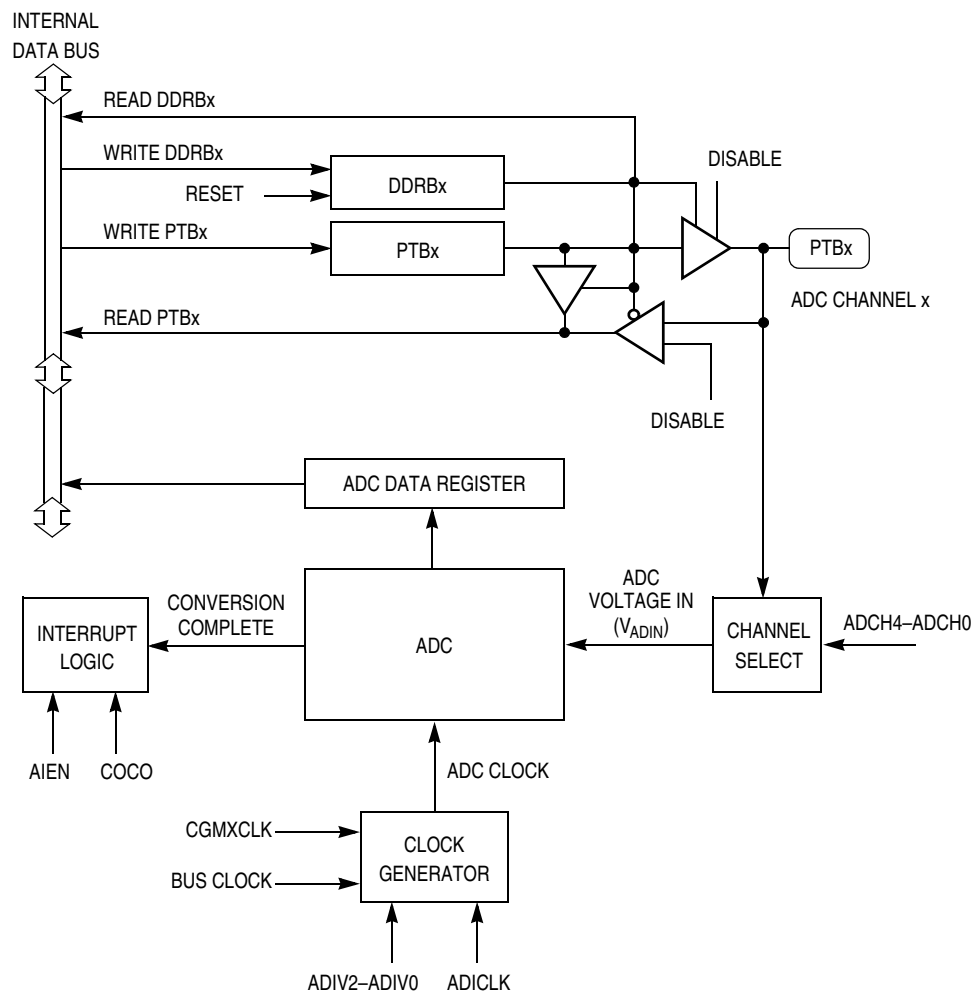


Figure 3-2. ADC Block Diagram

3.3.2 Voltage Conversion

When the input voltage to the ADC equals V_{REFH} , the ADC converts the signal to \$3FF (full scale). If the input voltage equals V_{REFL} , the ADC converts it to \$000. Input voltages between V_{REFH} and V_{REFL} are a straight-line linear conversion.

NOTE

The ADC input voltage must always be greater than V_{SSAD} and less than V_{DDAD} .

Connect the V_{DDAD} pin to the same voltage potential as the V_{DD} pin, and connect the V_{SSAD} pin to the same voltage potential as the V_{SS} pin.

The V_{DDAD} pin should be routed carefully for maximum noise immunity.

The following conditions apply when in manual mode:

- \overline{ACQ} is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the \overline{ACQ} bit must be clear.
- Before entering tracking mode ($\overline{ACQ} = 1$), software must wait a given time, t_{ACQ} (See 4.8 Acquisition/Lock Time Specifications.), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL} , after entering tracking mode before selecting the PLL as the clock source to CGMOUT (BCS = 1).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

4.3.6 Programming the PLL

Use the following procedure to program the PLL. For reference, the variables used and their meaning are shown in Table 4-1.

Table 4-1. Variable Definitions

Variable	Definition
f_{BUSDES}	Desired bus clock frequency
$f_{VCLKDES}$	Desired VCO clock frequency
f_{RCLK}	Chosen reference crystal frequency
f_{VCLK}	Calculated VCO clock frequency
f_{BUS}	Calculated bus clock frequency
f_{NOM}	Nominal VCO center frequency
f_{VRS}	Programmed VCO center frequency

NOTE

The round function in the following equations means that the real number should be rounded to the nearest integer number.

1. Choose the desired bus frequency, f_{BUSDES} .
2. Calculate the desired VCO frequency (four times the desired bus frequency).

$$f_{VCLKDES} = 4 \times f_{BUSDES}$$

3. Choose a practical PLL (crystal) reference frequency, f_{RCLK} . Typically, the reference crystal is 1–8 MHz.

Frequency errors to the PLL are corrected at a rate of f_{RCLK} .

For stability and lock time reduction, this rate must be as fast as possible. The VCO frequency must be an integer multiple of this rate. The relationship between the VCO frequency, f_{VCLK} , and the reference frequency, f_{RCLK} , is:

$$f_{VCLK} = (N) (f_{RCLK})$$

N, the range multiplier, must be an integer.

Clock Generator Module (CGM)

In cases where desired bus frequency has some tolerance, choose f_{RCLK} to a value determined either by other module requirements (such as modules which are clocked by CGMXCLK), cost requirements, or ideally, as high as the specified range allows. See Chapter 21 Electrical Specifications. After choosing N, the actual bus frequency can be determined using equation in 2 above.

4. Select a VCO frequency multiplier, N.

$$N = \text{round}\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right)$$

5. Calculate and verify the adequacy of the VCO and bus frequencies f_{VCLK} and f_{BUS} .

$$f_{VCLK} = (N) \times f_{RCLK}$$

$$f_{BUS} = (f_{VCLK})/4$$

6. Select the VCO's power-of-two range multiplier E, according to Table 4-2.

Table 4-2. Power-of-Two Range Selectors

Frequency Range	E
$0 < f_{VCLK} \leq 8 \text{ MHz}$	0
$8 \text{ MHz} < f_{VCLK} \leq 16 \text{ MHz}$	1
$16 \text{ MHz} < f_{VCLK} \leq 32 \text{ MHz}$	$2^{(1)}$

1. Do not program E to a value of 3.

7. Select a VCO linear range multiplier, L, where $f_{NOM} = 71.4 \text{ kHz}$

$$L = \text{Round}\left(\frac{f_{VCLK}}{2^E \times f_{NOM}}\right)$$

8. Calculate and verify the adequacy of the VCO programmed center-of-range frequency, f_{VRS} . The center-of-range frequency is the midpoint between the minimum and maximum frequencies attainable by the PLL.

$$f_{VRS} = (L \times 2^E) f_{NOM}$$

9. For proper operation,

$$|f_{VRS} - f_{VCLK}| \leq \frac{f_{NOM} \times 2^E}{2}$$

10. Verify the choice of N, E, and L by comparing f_{VCLK} to f_{VRS} and $f_{VCLKDES}$. For proper operation, f_{VCLK} must be within the application's tolerance of $f_{VCLKDES}$, and f_{VRS} must be as close as possible to f_{VCLK} .

NOTE

Exceeding the recommended maximum bus frequency or VCO frequency can crash the MCU.

PLLIE — PLL Interrupt Enable Bit

This read/write bit enables the PLL to generate an interrupt request when the LOCK bit toggles, setting the PLL flag, PLLF. When the AUTO bit in the PLL bandwidth control register (PBWC) is clear, PLLIE cannot be written and reads as logic 0. Reset clears the PLLIE bit.

- 1 = PLL interrupts enabled
- 0 = PLL interrupts disabled

PLLF — PLL Interrupt Flag Bit

This read-only bit is set whenever the LOCK bit toggles. PLLF generates an interrupt request if the PLLIE bit also is set. PLLF always reads as logic 0 when the AUTO bit in the PLL bandwidth control register (PBWC) is clear. Clear the PLLF bit by reading the PLL control register. Reset clears the PLLF bit.

- 1 = Change in lock condition
- 0 = No change in lock condition

NOTE

Do not inadvertently clear the PLLF bit. Any read or read-modify-write operation on the PLL control register clears the PLLF bit.

PLLON — PLL On Bit

This read/write bit activates the PLL and enables the VCO clock, CGMVCLK. PLLON cannot be cleared if the VCO clock is driving the base clock, CGMOUT (BCS = 1). (See 4.3.8 Base Clock Selector Circuit.) Reset sets this bit so that the loop can stabilize as the MCU is powering up.

- 1 = PLL on
- 0 = PLL off

BCS — Base Clock Select Bit

This read/write bit selects either the crystal oscillator output, CGMXCLK, or the VCO clock, CGMVCLK, as the source of the CGM output, CGMOUT. CGMOUT frequency is one-half the frequency of the selected clock. BCS cannot be set while the PLLON bit is clear. After toggling BCS, it may take up to three CGMXCLK and three CGMVCLK cycles to complete the transition from one source clock to the other. During the transition, CGMOUT is held in stasis. (See 4.3.8 Base Clock Selector Circuit.) Reset clears the BCS bit.

- 1 = CGMVCLK divided by two drives CGMOUT
- 0 = CGMXCLK divided by two drives CGMOUT

NOTE

PLLON and BCS have built-in protection that prevents the base clock selector circuit from selecting the VCO clock as the source of the base clock if the PLL is off. Therefore, PLLON cannot be cleared when BCS is set, and BCS cannot be set when PLLON is clear. If the PLL is off (PLLON = 0), selecting CGMVCLK requires two writes to the PLL control register. (See 4.3.8 Base Clock Selector Circuit.).

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 OSCSTOPENB bit in the CONFIG 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 OSCSTOPENB bit in the CONFIG 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 16.7.3 Break Flag Control Register.

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

4.8 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

4.8.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5 percent acquisition time tolerance. If a command instructs the system to change from 0 Hz to 1 MHz, the acquisition time is the time taken for the frequency to reach $1\text{ MHz} \pm 50\text{ kHz}$. Fifty kHz = 5% of the 1-MHz step input. If the system is operating at 1 MHz and suffers a –100-kHz noise hit, the acquisition time is the time taken to return from 900 kHz to $1\text{ MHz} \pm 5\text{ kHz}$. Five kHz = 5% of the 100-kHz step input.

Other systems refer to acquisition and lock times as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the acquisition or lock time varies according to the original error in the output. Minor errors may not even be registered. Typical PLL applications prefer to use this definition because the system requires the output frequency to be within a certain tolerance of the desired frequency regardless of the size of the initial error.

4.8.2 Parametric Influences on Reaction Time

Acquisition and lock times are designed to be as short as possible while still providing the highest possible stability. These reaction times are not constant, however. Many factors directly and indirectly affect the acquisition time.

The most critical parameter which affects the reaction times of the PLL is the reference frequency, f_{RCLK} . This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are required to reduce the frequency error. Therefore, the slower the reference the longer it takes to make these corrections. This parameter is under user control via the choice of crystal frequency f_{XCLK} . (See 4.3.3 PLL Circuits and 4.3.6 Programming the PLL.)

Another critical parameter is the external filter network. The PLL modifies the voltage on the VCO by adding or subtracting charge from capacitors in this network. Therefore, the rate at which the voltage changes for a given frequency error (thus change in charge) is proportional to the capacitance. The size of the capacitor also is related to the stability of the PLL. If the capacitor is too small, the PLL cannot make small enough adjustments to the voltage and the system cannot lock. If the capacitor is too large, the PLL may not be able to adjust the voltage in a reasonable time. (See 4.8.3 Choosing a Filter.)

Also important is the operating voltage potential applied to V_{DDA} . The power supply potential alters the characteristics of the PLL. A fixed value is best. Variable supplies, such as batteries, are acceptable if they vary within a known range at very slow speeds. Noise on the power supply is not acceptable, because it causes small frequency errors which continually change the acquisition time of the PLL.

Temperature and processing also can affect acquisition time because the electrical characteristics of the PLL change. The part operates as specified as long as these influences stay within the specified limits. External factors, however, can cause drastic changes in the operation of the PLL. These factors include noise injected into the PLL through the filter capacitor, filter capacitor leakage, stray impedances on the circuit board, and even humidity or circuit board contamination.

4.8.3 Choosing a Filter

As described in 4.8.2 Parametric Influences on Reaction Time, the external filter network is critical to the stability and reaction time of the PLL. The PLL is also dependent on reference frequency and supply voltage.

Figure 4-9 shows two types of filter circuits. In low-cost applications, where stability and reaction time of the PLL are not critical, the three component filter network shown in Figure 4-9 (B) can be replaced by a single capacitor, C_F , as shown in shown in Figure 4-9 (A). Refer to Table 4-5 for recommended filter components at various reference frequencies. For reference frequencies between the values listed in the table, extrapolate to the nearest common capacitor value. In general, a slightly larger capacitor provides more stability at the expense of increased lock time.

7.3.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	V	1	1	H	I	N	Z	C
Write:								
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 7-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

- 1 = Overflow
- 0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

- 1 = Carry between bits 3 and 4
- 0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

- 1 = Interrupts disabled
- 0 = Interrupts enabled

NOTE

To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

- 1 = Negative result
- 0 = Non-negative result

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of \$00.

1 = Zero result

0 = Non-zero result

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag.

1 = Carry out of bit 7

0 = No carry out of bit 7

7.4 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations defined by the instruction set.

Refer to the *CPU08 Reference Manual* (document order number CPU08RM/AD) for a description of the instructions and addressing modes and more detail about the architecture of the CPU.

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 WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

7.5.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

7.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with \$FFFC:\$FFFD or with \$FEFC:\$FEFD in monitor mode

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

Keyboard Interrupt Module (KBI)

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

- 1 = Keyboard interrupt requests masked
- 0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

- 1 = Keyboard interrupt requests on falling edges and low levels
- 0 = Keyboard interrupt requests on falling edges only

9.7.2 Keyboard Interrupt Enable Register

The keyboard interrupt enable register enables or disables each port A pin to operate as a keyboard interrupt pin.

Address: \$001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 9-5. Keyboard Interrupt Enable Register (INTKBIER)

KBIE7–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch interrupt requests. Reset clears the keyboard interrupt enable register.

- 1 = PTAx pin enabled as keyboard interrupt pin
- 0 = PTAx pin not enabled as keyboard interrupt pin

Chapter 14

Resets and Interrupts

14.1 Introduction

Resets and interrupts are responses to exceptional events during program execution. A reset re-initializes the microcontroller (MCU) to its startup condition. An interrupt vectors the program counter to a service routine.

14.2 Resets

A reset immediately returns the MCU to a known startup condition and begins program execution from a user-defined memory location.

14.2.1 Effects

A reset:

- Immediately stops the operation of the instruction being executed
- Initializes certain control and status bits
- Loads the program counter with a user-defined reset vector address from locations \$FFFE and \$FFFF
- Selects CGMXCLK divided by four as the bus clock

14.2.2 External Reset

A logic 0 applied to the $\overline{\text{RST}}$ pin for a time, t_{RL} , generates an external reset. An external reset sets the PIN bit in the system integration module (SIM) reset status register.

14.2.3 Internal Reset

Sources:

- Power-on reset (POR)
- Computer operating properly (COP)
- Low-power reset circuits
- Illegal opcode
- Illegal address

All internal reset sources pull the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles to allow resetting of external devices. The MCU is held in reset for an additional 32 CGMXCLK cycles after releasing the $\overline{\text{RST}}$ pin.

After every instruction, the CPU checks all pending interrupts if the I bit is not set. If more than one interrupt is pending when an instruction is done, the highest priority interrupt is serviced first. In the example shown in Figure 14-4, if an interrupt is pending upon exit from the interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

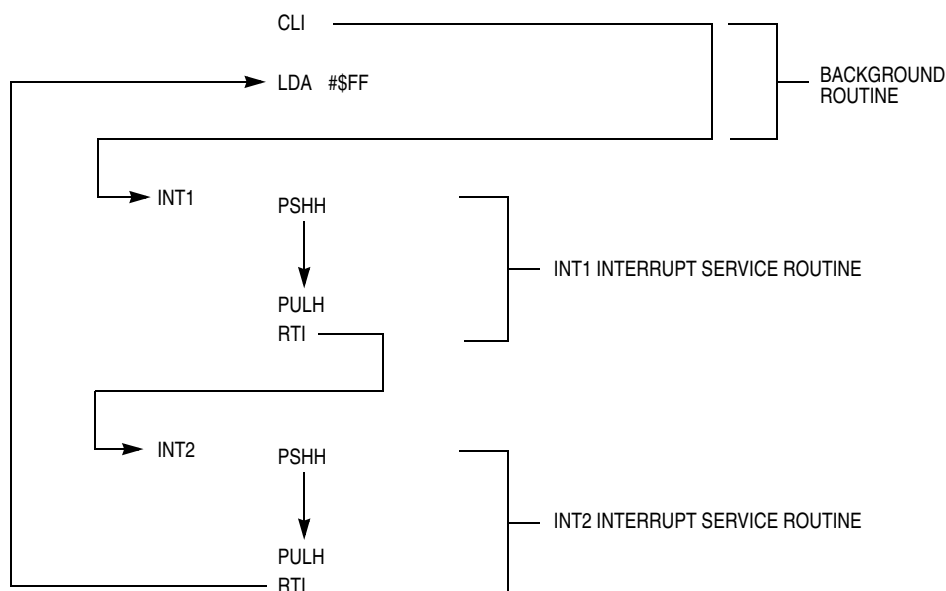


Figure 14-4. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE

To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, save the H register and then restore it prior to exiting the routine.

See Figure 14-5 for a flowchart depicting interrupt processing.

14.3.2 Sources

The sources in Table 14-1 can generate CPU interrupt requests.

14.3.2.1 Software Interrupt (SWI) Instruction

The software interrupt (SWI) instruction causes a non-maskable interrupt.

NOTE

*A software interrupt pushes PC onto the stack. An SWI does **not** push PC – 1, as a hardware interrupt does.*

14.3.2.2 Break Interrupt

The break module causes the CPU to execute an SWI instruction at a software-programmable break point.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$\left| \frac{170 - 163}{170} \right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 15-9 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

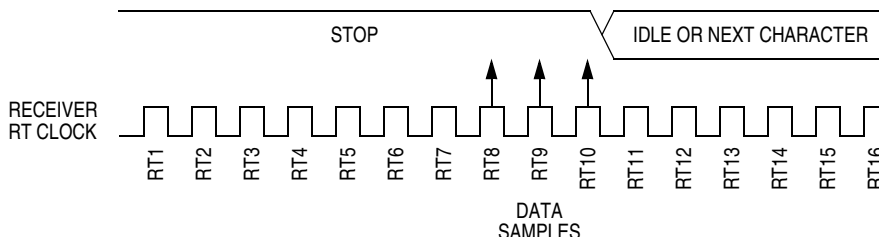


Figure 15-9. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times \times 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 15-9, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times \times 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left| \frac{154 - 160}{154} \right| \times 100 = 3.90\%.$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times \times 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 15-9, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times \times 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%.$$

15.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

ILAD — Illegal Address Reset Bit (opcode fetches only)

1 = Last reset caused by an opcode fetch from an illegal address
0 = POR or read of SRSR

MODRST — Monitor Mode Entry Module Reset Bit

1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while $\overline{IRQ} = V_{DD}$
0 = POR or read of SRSR

LVI — Low-Voltage Inhibit Reset Bit

1 = Last reset caused by the LVI circuit
0 = POR or read of SRSR

16.7.3 Break Flag Control Register

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

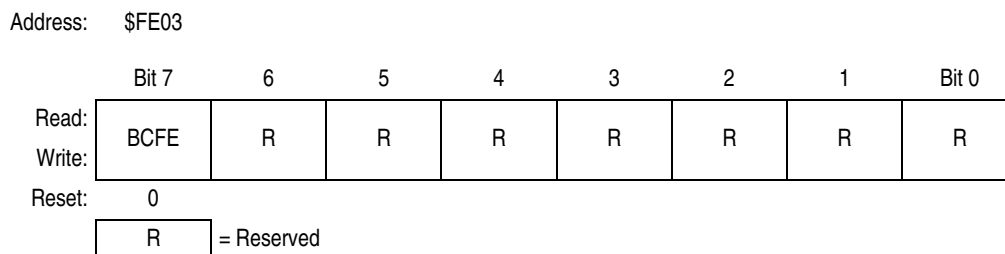


Figure 16-22. Break Flag Control Register (BFCR)

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

The CPU can always read the state of the \overline{SS} pin by configuring the appropriate pin as an input and reading the port data register. See Table 17-3.

Table 17-3. SPI Configuration

SPE	SPMSTR	MODFEN	SPI Configuration	State of \overline{SS} Logic
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

17.12.5 CGND (Clock Ground)

CGND is the ground return for the serial clock pin, SPSCCK, and the ground for the port output buffers. It is internally connected to V_{SS} as shown in Table 17-1.

17.13 I/O Registers

Three registers control and monitor SPI operation:

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

17.13.1 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 SPSCCK, MOSI, and MISO pins as open-drain outputs
- Enables the SPI module

Address: \$0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
Write:								
Reset:	0	0	1	0	1	0	0	0
	R		= Reserved					

Figure 17-14. 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

18.7 Timebase Control Register

The timebase has one register, the timebase control register (TBCR), which is used to enable the timebase interrupts and set the rate.

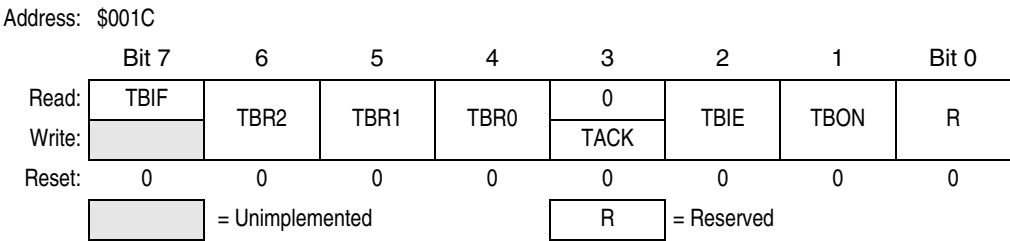


Figure 18-2. Timebase Control Register (TBCR)

TBIF — Timebase Interrupt Flag

This read-only flag bit is set when the timebase counter has rolled over.

- 1 = Timebase interrupt pending
- 0 = Timebase interrupt not pending

TBR2–TBR0 — Timebase Divider Selection Bits

These read/write bits select the tap in the counter to be used for timebase interrupts as shown in Table 18-1.

NOTE

Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).

TACK— Timebase Acknowledge Bit

The TACK bit is a write-only bit and always reads as 0. Writing a logic 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a logic 0 to this bit has no effect.

- 1 = Clear timebase interrupt flag
- 0 = No effect

TBIE — Timebase Interrupt Enabled Bit

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

- 1 = Timebase interrupt is enabled.
- 0 = Timebase interrupt is disabled.

TBON — Timebase Enabled Bit

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

- 1 = Timebase is enabled.
- 0 = Timebase is disabled and the counter initialized to 0s.

Timer Interface Module (TIM)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0020	Timer 1 Status and Control Register (T1SC) See page 265.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$0021	Timer 1 Counter Register High (T1CNTH) See page 266.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0022	Timer 1 Counter Register Low (T1CNTL) See page 266.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0023	Timer 1 Counter Modulo Register High (T1MODH) See page 267.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0024	Timer 1 Counter Modulo Register Low (T1MODL) See page 267.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0025	Timer 1 Channel 0 Status and Control Register (T1SC0) See page 267.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0026	Timer 1 Channel 0 Register High (T1CH0H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
\$0027	Timer 1 Channel 0 Register Low (T1CH0L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
\$0028	Timer 1 Channel 1 Status and Control Register (T1SC1) See page 267.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0029	Timer 1 Channel 1 Register High (T1CH1H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
\$002A	Timer 1 Channel 1 Register Low (T1CH1L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
\$002B	Timer 2 Status and Control Register (T2SC) See page 265.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$002C	Timer 2 Counter Register High (T2CNTH) See page 266.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0

□ = Unimplemented

Figure 19-3. TIM I/O Register Summary (Sheet 1 of 2)