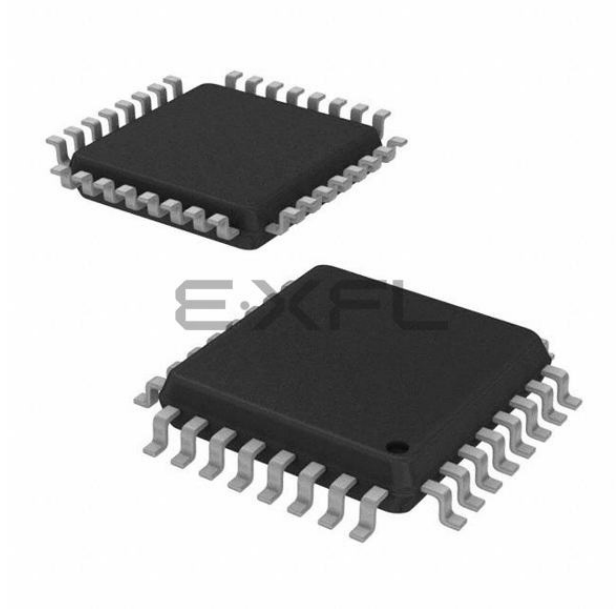


Welcome to E-XFL.COM

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, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/mc68hc908gz60cfj

Revision History (Continued)

Date	Revision Level	Description	Page Number(s)
October, 2006	5.0	12.2 Features — Corrected timer link connection from TIM2 channel 0 to TIM1 channel 0.	135
		12.9 Timer Link — Corrected timer link connection from TIM2 channel 0 to TIM1 channel 0.	147
		21.5 5.0-Vdc Electrical Characteristics and 21.6 3.3-Vdc Electrical Characteristics — Updated DC injection current specification.	317 319
April, 2007	6.0	Figure 2-2. Control, Status, and Data Registers — Changed TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	37
		Chapter 5 Configuration Register (CONFIG) — Changed COPCLK to CGMXCLK and TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	91 92 93
		10.6.2 Stop Mode — Changed COPCLK to CGMXCLK	125
		Figure 14-3. ESCI Module Block Diagram — Changed BUS_CLK to BUS CLOCK and removed reference to 4xBUSCLK	192
		14.4.2 Transmitter — Changed ESCIBDSRC to SCIBDSRC	194
		14.9.1 ESCI Arbiter Control Register and 14.9.3 Bit Time Measurement — Replaced one quarter with one half in the definition for ACLK = 1	217 218
		Figure 17-1. Timebase Block Diagram, 17.5 TBM Interrupt Rate, and Table 17-1. Timebase Divider Selection — Changed TBMCLKSEL to TMBCLKSEL to be compatible with development tool nomenclature	260 261
		21.9 Clock Generation Module (CGM) Characteristics — Updated section to include the following: 21.9.1 CGM Operating Conditions 21.9.2 CGM Component Information 21.9.3 CGM Acquisition/Lock Time Information	 322 322 323

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 μs).
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 μs).
9. Clear the HVEN bit.
10. Wait for a time, t_{RCV}, (typically 1 μs) 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

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.

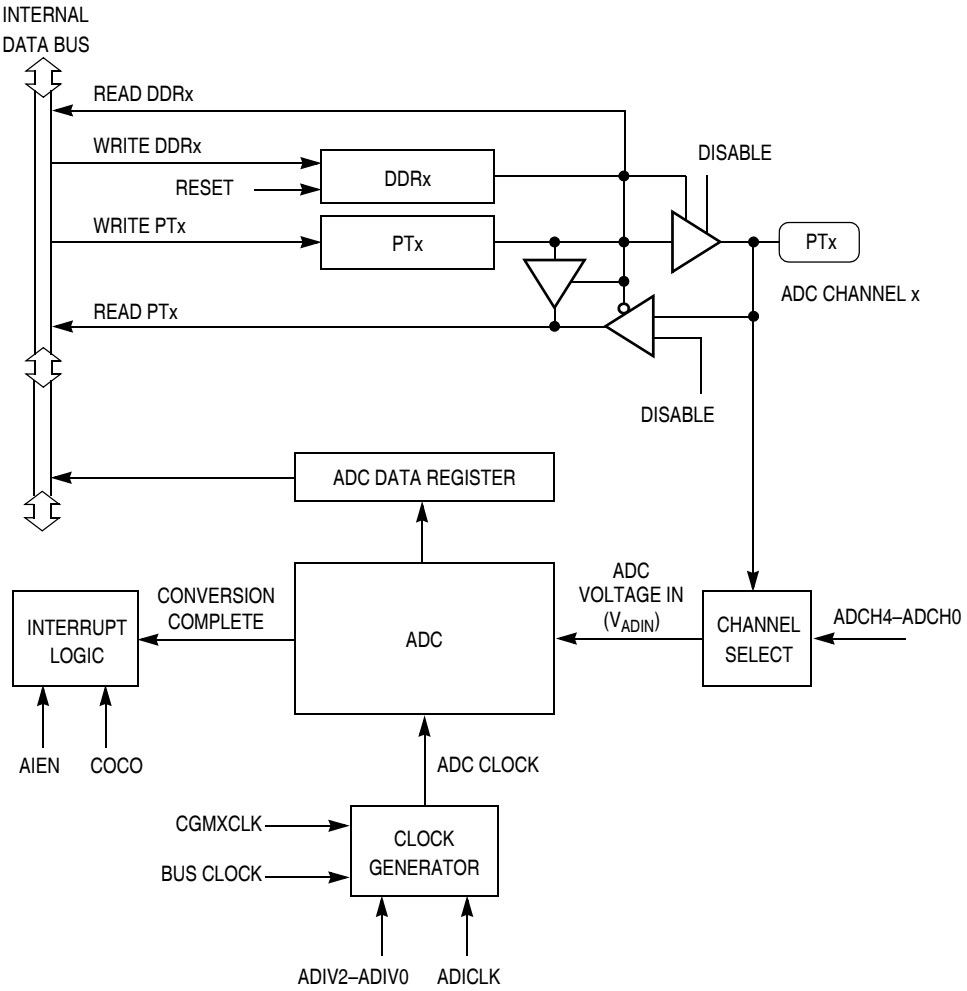


Figure 3-2. ADC Block Diagram

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.

LVI5OR3 — LVI 5-V or 3-V Operating Mode Bit

LVI5OR3 selects the voltage operating mode of the LVI module (see Chapter 11 Low-Voltage Inhibit (LVI)). The voltage mode selected for the LVI should match the operating V_{DD} (see Chapter 21 Electrical Specifications) for the LVI's voltage trip points for each of the modes.

1 = LVI operates in 5-V mode
0 = LVI operates in 3-V mode

NOTE

The LVI5OR3 bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 CGMXCLK cycles instead of a 4096-CGMXCLK cycle delay.

1 = Stop mode recovery after 32 CGMXCLK cycles
0 = Stop mode recovery after 4096 CGMXCLK cycles

NOTE

Exiting stop mode by any reset will result in the long stop recovery.

The short stop recovery delay can be enabled when using a crystal or resonator and the OSCENINSTOP bit is set. The short stop recovery delay can be enabled when an external oscillator is used, regardless of the OSCENINSTOP setting.

The short stop recovery delay must be disabled when the OSCENINSTOP bit is clear and a crystal or resonator is used.

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. See Chapter 6 Computer Operating Properly (COP) Module.

1 = COP module disabled
0 = COP module enabled

6.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register. See Chapter 5 Configuration Register (CONFIG).

6.3.7 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register. See Chapter 5 Configuration Register (CONFIG).

6.4 COP Control Register

The COP control register (COPCTL) is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address: $FFFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 6-2. COP Control Register (COPCTL)

6.5 Interrupts

The COP does not generate central processor unit (CPU) interrupt requests.

6.6 Monitor Mode

When monitor mode is entered with V_{TST} on the $\overline{IRQ}$ pin, the COP is disabled as long as V_{TST} remains on the $\overline{IRQ}$ pin or the $\overline{RST}$ pin. When monitor mode is entered by having blank reset vectors and not having V_{TST} on the $\overline{IRQ}$ pin, the COP is automatically disabled until a POR occurs.

6.7 Low-Power Modes

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

6.7.1 Wait Mode

The COP remains active during wait mode. If COP is enabled, a reset will occur at COP timeout.

6.7.2 Stop Mode

Stop mode turns off the CGMXCLK input to the COP and clears the SIM counter. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

10.3 Break Module (BRK)

10.3.1 Wait Mode

The break (BRK) module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the SBSW bit in the break status register is set.

10.3.2 Stop Mode

The break module is inactive in stop mode. The STOP instruction does not affect break module register states.

10.4 Central Processor Unit (CPU)

10.4.1 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

10.4.2 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.

10.5 Clock Generator Module (CGM)

10.5.1 Wait Mode

The clock generator module (CGM) remains active in wait mode. 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). Less power-sensitive applications can disengage the PLL without turning it off. Applications that require the PLL to wake the MCU from wait mode also can deselect the PLL output without turning off the PLL.

10.5.2 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.

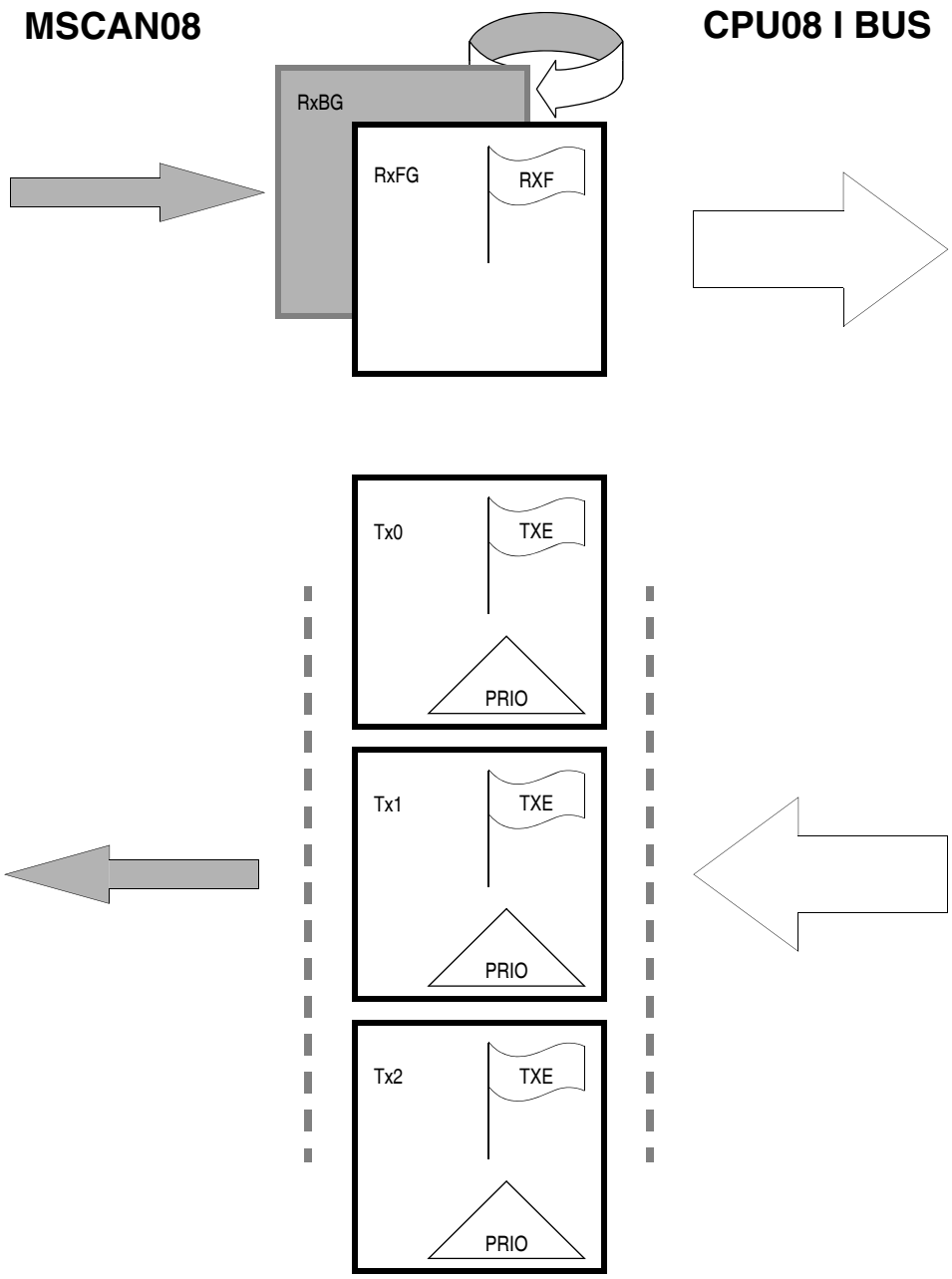


Figure 12-3. User Model for Message Buffer Organization

12.4.3 Transmit Structures

The MSCAN08 has a triple transmit buffer scheme to allow multiple messages to be set up in advance and to achieve an optimized real-time performance. The three buffers are arranged as shown in Figure 12-3.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see 12.12 Programmer's Model of Message Storage). An additional transmit buffer priority register (TBPR) contains an 8-bit "local priority" field (PRIO) (see 12.12.5 Transmit Buffer Priority Registers).

MSCAN08 bus activity can wake the MCU from CPU stop/MSCAN08 power-down mode. However, until the oscillator starts up and synchronization is achieved the MSCAN08 will not respond to incoming data.

12.8.4 CPU Wait Mode

The MSCAN08 module remains active during CPU wait mode. The MSCAN08 will stay synchronized to the CAN bus and generates transmit, receive, and error interrupts to the CPU, if enabled. Any such interrupt will bring the MCU out of wait mode.

12.8.5 Programmable Wakeup Function

The MSCAN08 can be programmed to apply a low-pass filter function to the CAN_{RX} input line while in internal sleep mode (see information on control bit WUPM in 12.13.2 MSCAN08 Module Control Register 1). This feature can be used to protect the MSCAN08 from wakeup due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic inference within noisy environments.

12.9 Timer Link

The MSCAN08 will generate a timer signal whenever a valid frame has been received. Because the CAN specification defines a frame to be valid if no errors occurred before the EOF field has been transmitted successfully, the timer signal will be generated right after the EOF. A pulse of one bit time is generated. As the MSCAN08 receiver engine also receives the frames being sent by itself, a timer signal also will be generated after a successful transmission.

The previously described timer signal can be routed into the on-chip timer interface module (TIM). This signal is connected to channel 0 of timer interface module 1 (TIM1) under the control of the timer link enable (TLNKEN) bit in CMCR0.

After timer n has been programmed to capture rising edge events, it can be used under software control to generate 16-bit time stamps which can be stored with the received message.

12.10 Clock System

Figure 12-8 shows the structure of the MSCAN08 clock generation circuitry and its interaction with the clock generation module (CGM). With this flexible clocking scheme the MSCAN08 is able to handle CAN bus rates ranging from 10 kbps up to 1 Mbps.

The clock source bit (CLKSRC) in the MSCAN08 module control register (CMCR1) (see 12.13.1 MSCAN08 Module Control Register 0) defines whether the MSCAN08 is connected to the output of the crystal oscillator or to the PLL output.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met.

NOTE

If the system clock is generated from a PLL, it is recommended to select the crystal clock source rather than the system clock source due to jitter considerations, especially at faster CAN bus rates.

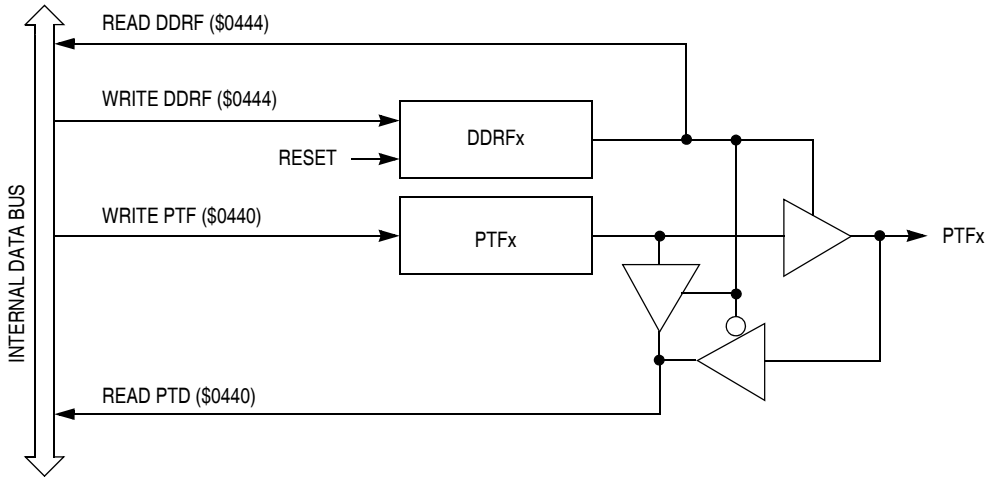


Figure 13-22. Port F I/O Circuit

When bit DDRFx is a 1, reading address $0440 reads the PTFx data latch. When bit DDRFx is a 0, reading address $0440 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 13-7 summarizes the operation of the port F pins.

Table 13-7. Port F Pin Functions

DDRF Bit	PTF Bit	I/O Pin Mode	Accesses to DDRF	Accesses to PTF	
			Read/Write	Read	WritE
0	X(1)	Input, Hi-Z(2)	DDRF7–DDRF0	Pin	PTF7–PTF0(3)
1	X	Output	DDRF7–DDRF0	PTF7–PTF0	PTF7–PTF0

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

13.9 Port G

Port G is an 8-bit special-function port that shares all eight of its pins with the analog-to-digital converter (ADC) module.

13.9.1 Port G Data Register

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

Address:	$0441							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTG7	PTG6	PTG5	PTG4	PTG3	PTG2	PTG1	PTG0
Reset:	Unaffected by reset							
Alternate Function:	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16

Figure 13-23. Port G Data Register (PTG)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	Break Status Register (BSR) See page 237.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note(1)	
		Reset:	0	0	0	0	0	0	0	0
			1. Writing a 0 clears SBSW.							
$FE01	SIM Reset Status Register (SRSR) See page 237.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE03	Break Flag Control Register (BFCR) See page 238.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1) See page 231.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2) See page 233.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3) See page 233.	Read:	IF22	IF32	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE07	Interrupt Status Register 4 (INT4) See page 233.	Read:	0	0	0	0	0	0	IF24	IF23
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
			= Unimplemented			R	= Reserved			

Figure 15-2. SIM I/O Register Summary

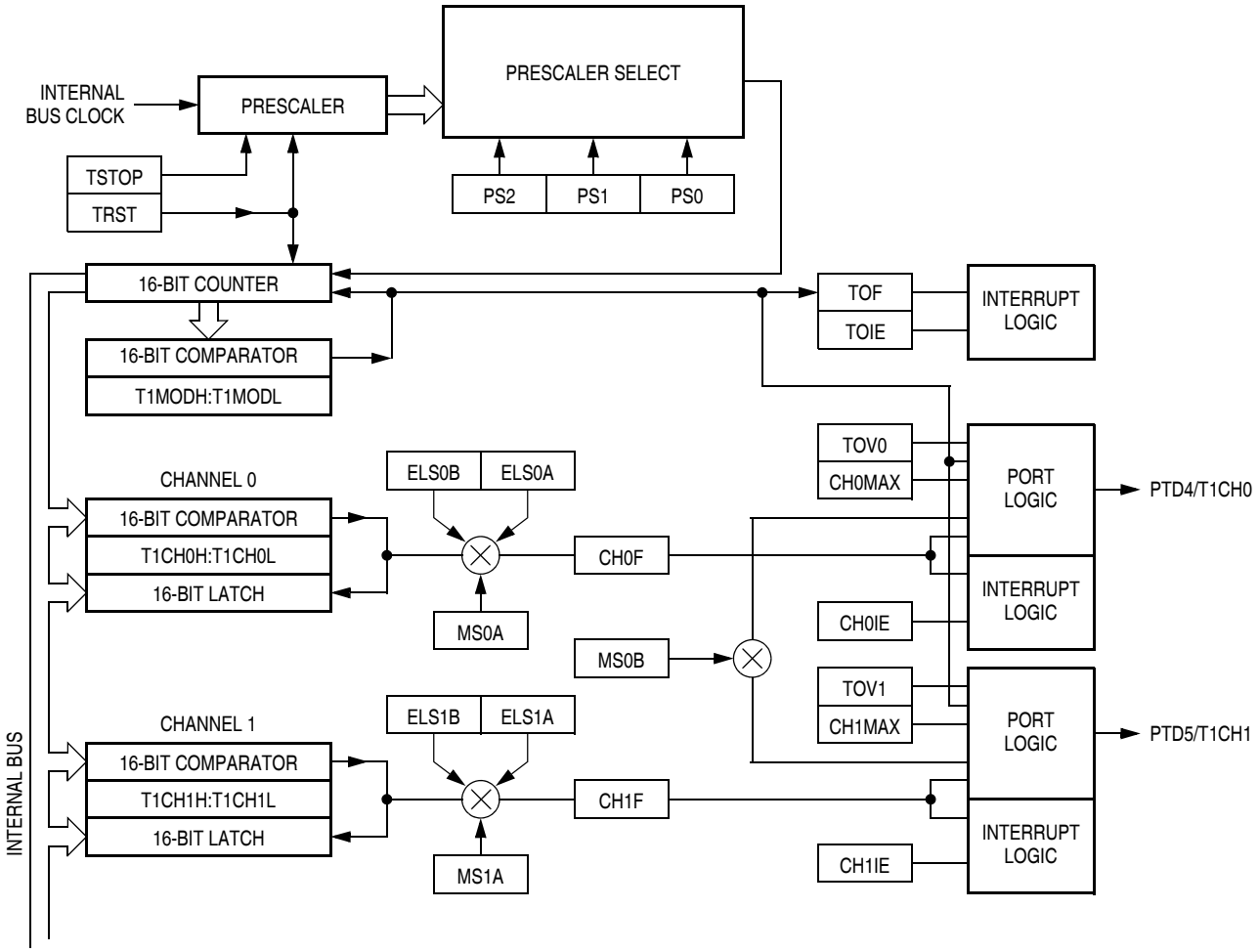


Figure 18-2. TIM1 Block Diagram

18.3.3 Output Compare

With the output compare function, the TIM1 can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM1 can set, clear, or toggle the channel pin. Output compares can generate TIM1 CPU interrupt requests.

18.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 18.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM1 overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM1 may pass the new value before it is written.

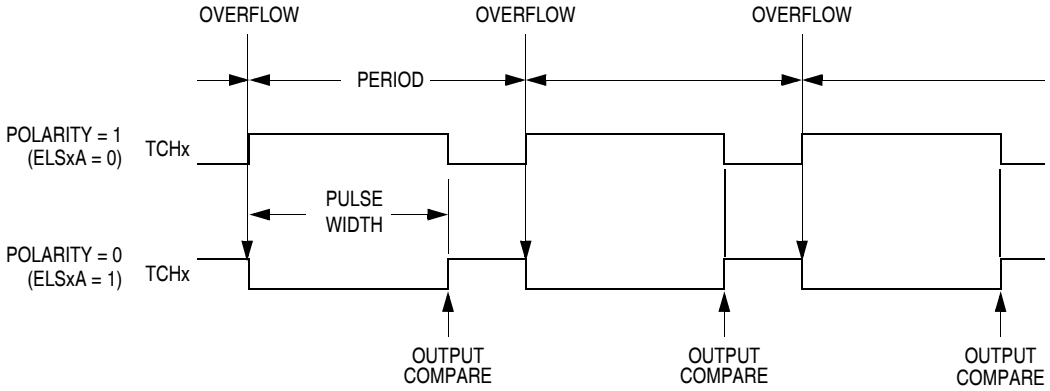


Figure 18-4. PWM Period and Pulse Width

18.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 18.3.4 Pulse Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM1 overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM1 may pass the new value before it is written to the timer channel (T1CHxH:T1CHxL) registers.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM1 overflow interrupts and write the new value in the TIM1 overflow interrupt routine. The TIM1 overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

Chapter 20
Development Support

20.1 Introduction

This section describes the break module, the monitor module (MON), and the monitor mode entry methods.

20.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features of the break module include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

20.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal ($\overline{\text{BKPT}}$) to the system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a 1 to the BRKA bit in the break status and control register.

When a CPU generated address matches the contents of the break address registers, the break interrupt is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

Figure 20-2 shows the structure of the break module.

Figure 20-3 provides a summary of the I/O registers.

When the internal address bus matches the value written in the break address registers or when software writes a 1 to the BRKA bit in the break status and control register, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC and $FFFD ($FEFC and $FEFD in monitor mode)

The break interrupt timing is:

- When a break address is placed at the address of the instruction opcode, the instruction is not executed until after completion of the break interrupt routine.
- When a break address is placed at an address of an instruction operand, the instruction is executed before the break interrupt.
- When software writes a 1 to the BRKA bit, the break interrupt occurs just before the next instruction is executed.

By updating a break address and clearing the BRKA bit in a break interrupt routine, a break interrupt can be generated continuously.

CAUTION

A break address should be placed at the address of the instruction opcode. When software does not change the break address and clears the BRKA bit in the first break interrupt routine, the next break interrupt will not be generated after exiting the interrupt routine even when the internal address bus matches the value written in the break address registers.

20.2.1.1 Flag Protection During Break Interrupts

The system integration module (SIM) controls whether or not module status bits can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See 15.7.3 Break Flag Control Register and the **Break Interrupts** subsection for each module.

20.2.1.2 TIM During Break Interrupts

A break interrupt stops the timer counter and inhibits input captures.

20.2.1.3 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

20.2.2 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- Break status register (BSR)
- Break flag control register (BFCR)

20.2.2.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

Address: $FE0B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 20-4. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = Break address match
0 = No break address match

20.2.2.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: $FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 20-5. Break Address Register High (BRKH)

Address: $FE0A

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

Figure 20-6. Break Address Register Low (BRKL)

20.3.2 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations $FFF6–$FFFD. Locations $FFF6–$FFFD contain user-defined data.

NOTE

Do not leave locations $FFF6–$FFFD blank. For security reasons, program locations $FFF6–$FFFD even if they are not used for vectors.

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations $FFF6–$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. See Figure 20-17.

Upon power-on reset, if the received bytes of the security code do not match the data at locations $FFF6–$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading a FLASH location returns an invalid value and trying to execute code from FLASH causes an illegal address reset. After receiving the eight security bytes from the host, the MCU transmits a break character, signifying that it is ready to receive a command.

NOTE

The MCU does not transmit a break character until after the host sends the eight security bytes.

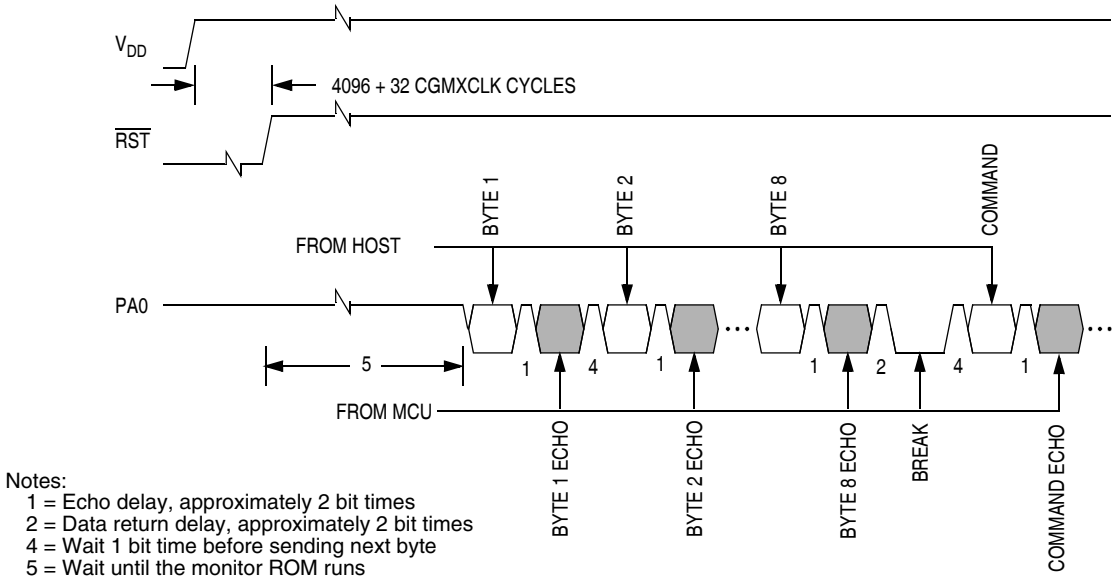


Figure 20-17. Monitor Mode Entry Timing

To determine whether the security code entered is correct, check to see if bit 6 of RAM address $40 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

If the security sequence fails, the device should be reset by a power-on reset and brought up in monitor mode to attempt another entry. After failing the security sequence, the FLASH module can also be mass erased by executing an erase routine that was downloaded into internal RAM. The mass erase operation clears the security code locations so that all eight security bytes become $FF (blank).

21.7 5.0-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option[2]	f_{OSC}	 1 dc	 8 32	MHz
Internal operating frequency	f_{OP} (f_{Bus})	—	8	MHz
Internal clock period ($1/f_{OP}$)	t_{CYC}	125	—	ns
RESET input pulse width low	t_{RL}	100	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)	t_{ILIH}	100	—	ns
$\overline{IRQ}$ interrupt pulse period[3]	t_{ILIL}	Note 3	—	t_{CYC}

1. V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.
2. No more than 10% duty cycle deviation from 50%.
3. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

21.8 3.3-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option[2]	f_{OSC}	 1 dc	 8 16	MHz
Internal operating frequency	f_{OP} (f_{Bus})	—	4	MHz
Internal clock period ($1/f_{OP}$)	t_{CYC}	250	—	ns
RESET input pulse width low	t_{RL}	200	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)	t_{ILIH}	200	—	ns
$\overline{IRQ}$ interrupt pulse period[3]	t_{ILIL}	Note 3	—	t_{CYC}

1. V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.
2. No more than 10% duty cycle deviation from 50%.
3. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

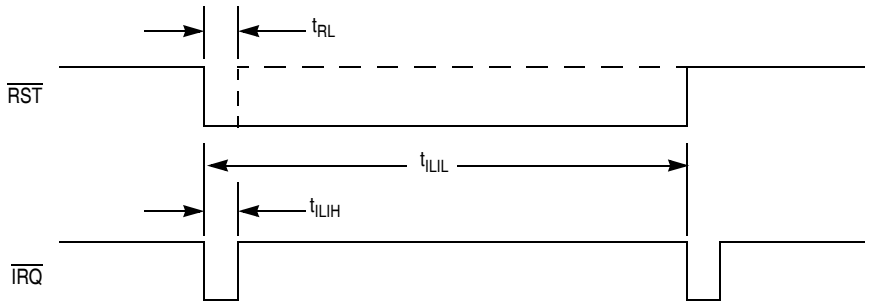
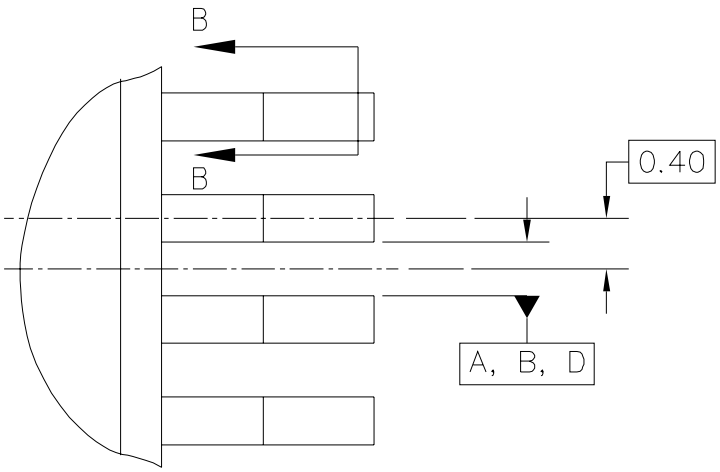
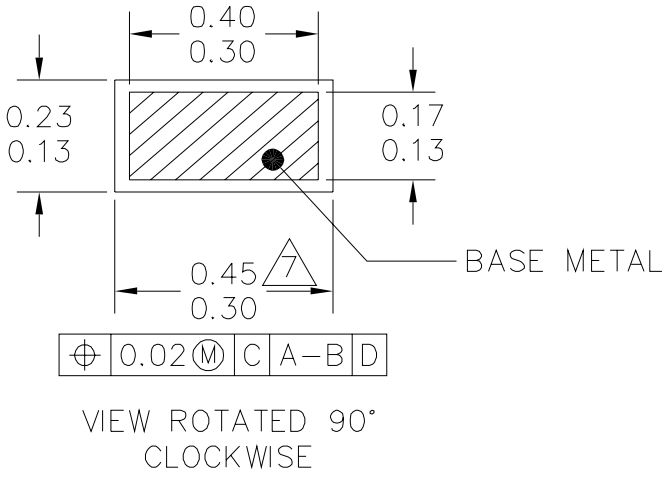


Figure 21-1. $\overline{RST}$ and $\overline{IRQ}$ Timing

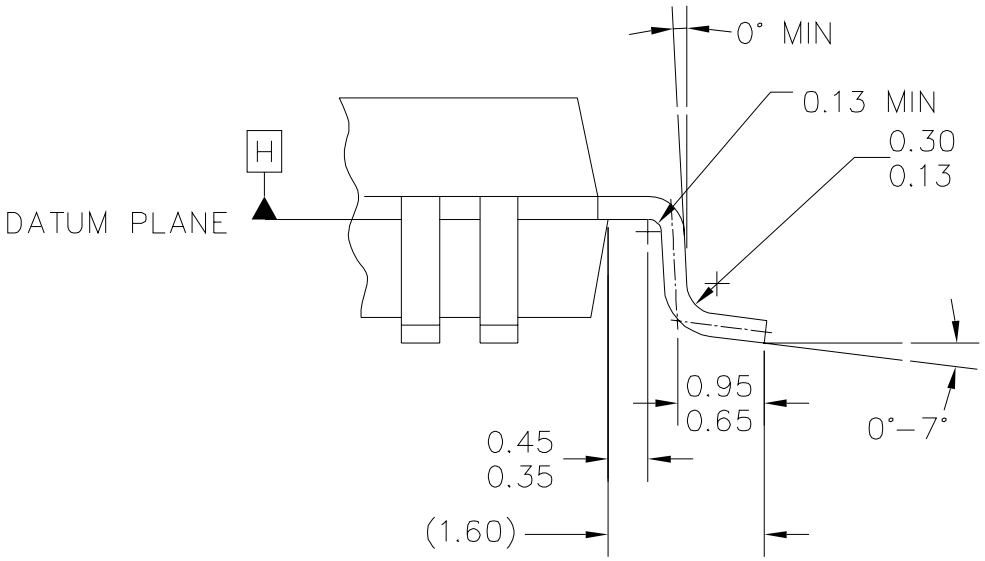


DETAIL "A"



VIEW ROTATED 90° CLOCKWISE

SECTION B–B



DETAIL "C"

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42844B	REV: A
64LD QFP (14 X 14)	CASE NUMBER: 840B-02	06 APR 2005
	STANDARD: NON-JEDEC	

1. $FFF6–$FFFD used for eight security bytes

Figure A-2. MC68HC908GZ48 Memory Map