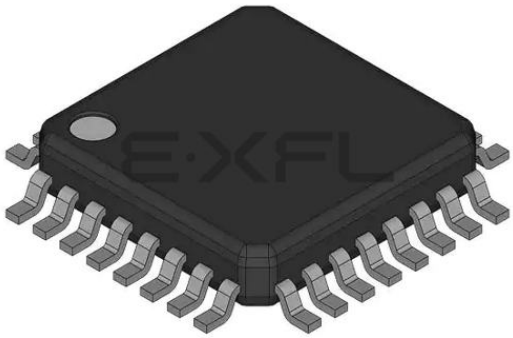


Welcome to E-XFL.COM

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	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/pro/item?MUrl=&PartUrl=mc68hc908gr16vfa

List of Chapters

Chapter 10
Low-Power Modes

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:

- Eight 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 eight pins for sampling external sources at pins PTB7/KBD7–PTB0/KBD0. An analog multiplexer allows the single ADC converter to select one of eight 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

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.

4.4.4 PLL Analog Power Pin (V_{DDA})

V_{DDA} is a power pin used by the analog portions of the PLL. Connect the V_{DDA} pin to the same voltage potential as the V_{DD} pin.

NOTE

Route V_{DDA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

4.4.5 PLL Analog Ground Pin (V_{SSA})

V_{SSA} is a ground pin used by the analog portions of the PLL. Connect the V_{SSA} pin to the same voltage potential as the V_{SS} pin.

NOTE

Route V_{SSA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

4.4.6 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables the oscillator and PLL.

4.4.7 Oscillator Stop Mode Enable Bit (OSCSTOPENB)

OSCSTOPENB is a bit in the CONFIG register that enables the oscillator to continue operating during stop mode. If this bit is set, the Oscillator continues running during stop mode. If this bit is not set (default), the oscillator is controlled by the SIMOSCEN signal which will disable the oscillator during stop mode.

4.4.8 Crystal Output Frequency Signal (CGMXCLK)

CGMXCLK is the crystal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 4-2 shows only the logical relation of CGMXCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of CGMXCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of CGMXCLK can be unstable at startup.

4.4.9 CGM Base Clock Output (CGMOUT)

CGMOUT is the clock output of the CGM. This signal goes to the SIM, which generates the MCU clocks. CGMOUT is a 50 percent duty cycle clock running at twice the bus frequency. CGMOUT is software programmable to be either the oscillator output, CGMXCLK, divided by two or the VCO clock, CGMVCLK, divided by two.

4.4.10 CGM CPU Interrupt (CGMINT)

CGMINT is the interrupt signal generated by the PLL lock detector.

RDS3–RDS0 — Reference Divider Select Bits

These read/write bits control the modulo reference divider that selects the reference division factor, R. (See 4.3.3 PLL Circuits and 4.3.6 Programming the PLL.) RDS7–RDS0 cannot be written when the PLLON bit in the PCTL is set. A value of $00 in the reference divider select register configures the reference divider the same as a value of $01. (See 4.3.7 Special Programming Exceptions.) Reset initializes the register to $01 for a default divide value of 1.

NOTE

The reference divider select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).

NOTE

The default divide value of 1 is recommended for all applications.

PMDS7–PMDS4 — Unimplemented Bits

These bits have no function and always read as 0s.

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.

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

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

A	Accumulator	*n*	Any bit
C	Carry/borrow bit	*opr*	Operand (one or two bytes)
CCR	Condition code register	PC	Program counter
dd	Direct address of operand	PCH	Program counter high byte
dd rr	Direct address of operand and relative offset of branch instruction	PCL	Program counter low byte
DD	Direct to direct addressing mode	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
DIX+	Direct to indexed with post increment addressing mode	rr	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1	Stack pointer, 8-bit offset addressing mode
EXT	Extended addressing mode	SP2	Stack pointer 16-bit offset addressing mode
ff	Offset byte in indexed, 8-bit offset addressing	SP	Stack pointer
H	Half-carry bit	U	Undefined
H	Index register high byte	V	Overflow bit
hh ll	High and low bytes of operand address in extended addressing	X	Index register low byte
I	Interrupt mask	Z	Zero bit
ii	Immediate operand byte	&	Logical AND
IMD	Immediate source to direct destination addressing mode	|	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX+	Indexed, no offset, post increment addressing mode	#	Immediate value
IX+D	Indexed with post increment to direct addressing mode	«	Sign extend
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX1+	Indexed, 8-bit offset, post increment addressing mode	?	If
IX2	Indexed, 16-bit offset addressing mode	:	Concatenated with
M	Memory location	↕	Set or cleared
N	Negative bit	—	Not affected

7.8 Opcode Map

See Table 7-2.

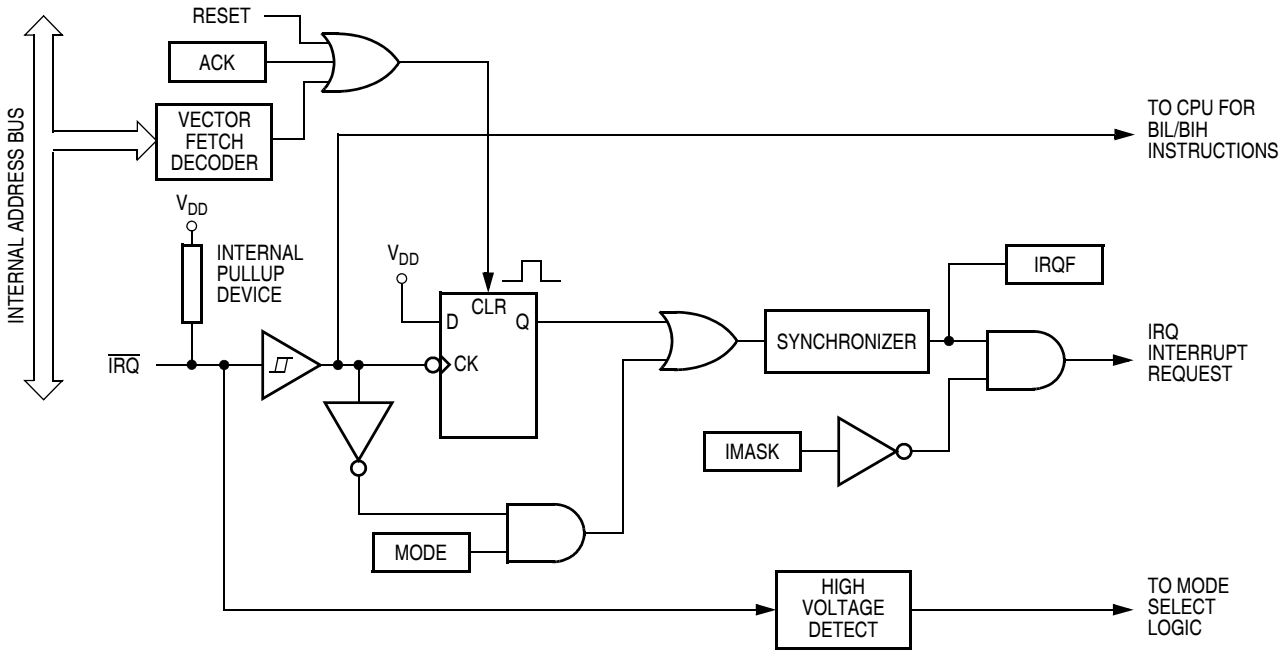


Figure 8-2. IRQ Module Block Diagram

When an interrupt pin is both falling-edge and low-level triggered (MODE = 1), the interrupt remains set until both of these events occur:

- Vector fetch or software clear
- Return of the interrupt pin to 1

The vector fetch or software clear may occur before or after the interrupt pin returns to 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the INTSCR mask all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK bit is clear.

NOTE

The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001D	IRQ Status and Control Register (INTSCR) See page 103.	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-3. IRQ I/O Register Summary

10.6 Computer Operating Properly Module (COP)

10.6.1 Wait Mode

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

10.6.2 Stop Mode

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

The STOP bit in the CONFIG1 register enables the STOP instruction. To prevent inadvertently turning off the COP with a STOP instruction, disable the STOP instruction by clearing the STOP bit.

10.7 External Interrupt Module (IRQ)

10.7.1 Wait Mode

The external interrupt (IRQ) module remains active in wait mode. Clearing the IMASK1 bit in the IRQ status and control register enables $\overline{\text{IRQ}}$ CPU interrupt requests to bring the MCU out of wait mode.

10.7.2 Stop Mode

The IRQ module remains active in stop mode. Clearing the IMASK1 bit in the IRQ status and control register enables $\overline{\text{IRQ}}$ CPU interrupt requests to bring the MCU out of stop mode.

10.8 Keyboard Interrupt Module (KBI)

10.8.1 Wait Mode

The keyboard interrupt (KBI) module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

10.8.2 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

10.9 Low-Voltage Inhibit Module (LVI)

10.9.1 Wait Mode

If enabled, the low-voltage inhibit (LVI) module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

LVISTOP, LVIPWRD, LVI5OR3, and LVIRSTD are in the configuration register (CONFIG1). See Figure 5-2. Configuration Register 1 (CONFIG1) for details of the LVI's configuration bits. Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR}, which causes the MCU to exit reset. See 15.3.2.5 Low-Voltage Inhibit (LVI) Reset for details of the interaction between the SIM and the LVI. The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR).

An LVI reset also drives the $\overline{RST}$ pin low to provide low-voltage protection to external peripheral devices.

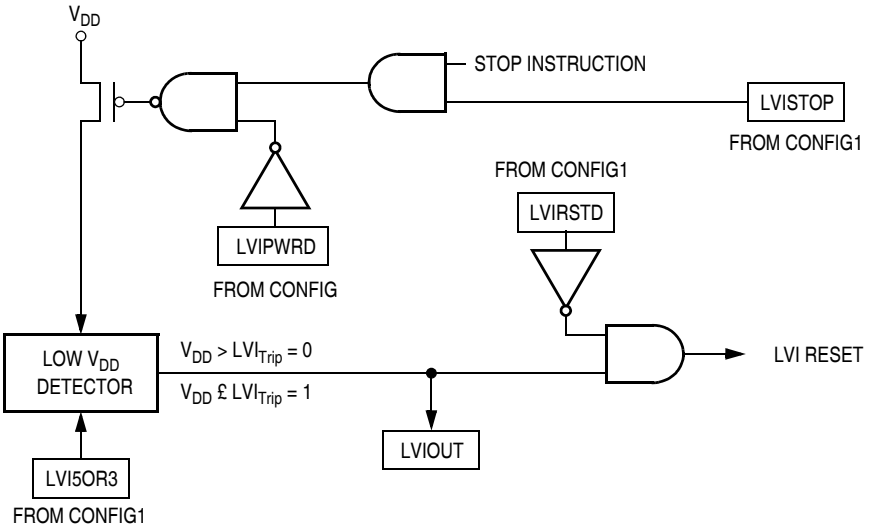


Figure 11-1. LVI Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE0C	LVI Status Register (LVISR) See page 119.	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-2. LVI I/O Register Summary

11.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register, the LVIPWRD bit must be at 0 to enable the LVI module, and the LVIRSTD bit must be at 1 to disable LVI resets.

11.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register, the LVIPWRD and LVIRSTD bits must be at 0 to enable the LVI module and to enable LVI resets.

11.3.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the hysteresis voltage, V_{HYS}.

11.3.4 LVI Trip Selection

The LVI5OR3 bit in the configuration register selects whether the LVI is configured for 5-V or 3-V protection.

NOTE

The microcontroller is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF} [5 V] or V_{TRIPF} [3 V]) may be lower than this. See Chapter 20 Electrical Specifications for the actual trip point voltages.

11.4 LVI Status Register

The LVI status register (LVISR) indicates if the V_{DD} voltage was detected below the V_{TRIPF} level.

Address: $FE0C

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

= Unimplemented

Figure 11-3. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage (see Table 11-1). Reset clears the LVIOUT bit.

Table 11-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

11.5 LVI Interrupts

The LVI module does not generate interrupt requests.

11.6 Low-Power Modes

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0005	Data Direction Register B (DDRB) See page 126.	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC) See page 128.	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD) See page 131.	Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE) See page 133.	Read:	0	0	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$000C	Data Direction Register E (DDRE) See page 134.	Read:	0	0	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 125.	Read: Write:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Reset:	0	0	0	0	0	0	0	0
$000E	Port C Input Pullup Enable Register (PTCPUE) See page 129.	Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F	Port D Input Pullup Enable Register (PTDPUE) See page 132.	Read: Write:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

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

14.9 ESCI Arbiter

The ESCI module comprises an arbiter module designed to support software for communication tasks as bus arbitration, baud rate recovery and break time detection. The arbiter module consists of an 9-bit counter with 1-bit overflow and control logic. The CPU can control operation mode via the ESCI arbiter control register (SCIACTL).

14.9.1 ESCI Arbiter Control Register

Address: $000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AM1	ALOST	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-19. ESCI Arbiter Control Register (SCIACTL)

AM1 and AM0 — Arbiter Mode Select Bits

These read/write bits select the mode of the arbiter module as shown in Table 14-12. Reset clears AM1 and AM0.

Table 14-12. ESCI Arbiter Selectable Modes

AM[1:0]	ESCI Arbiter Mode
0 0	Idle / counter reset
0 1	Bit time measurement
1 0	Bus arbitration
1 1	Reserved / do not use

ALOST — Arbitration Lost Flag

This read-only bit indicates loss of arbitration. Clear ALOST by writing a 0 to AM1. Reset clears ALOST.

ACLK — Arbiter Counter Clock Select Bit

This read/write bit selects the arbiter counter clock source. Reset clears ACLK.

1 = Arbiter counter is clocked with one half of the ESCI input clock generated by the ESCI prescaler
0 = Arbiter counter is clocked with the bus clock divided by four

NOTE

For ACLK = 1, the arbiter input clock is driven from the ESCI prescaler. The prescaler can be clocked by either the bus clock or CGMXCLK depending on the state of the ESCIBDSRC bit in CONFIG2.

AFIN— Arbiter Bit Time Measurement Finish Flag

This read-only bit indicates bit time measurement has finished. Clear AFIN by writing any value to SCIACTL. Reset clears AFIN.

1 = Bit time measurement has finished
0 = Bit time measurement not yet finished

The clock phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE

Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI enable bit (SPE).

16.5.2 Transmission Format When CPHA = 0

Figure 16-5 shows an SPI transmission in which CPHA is 0. 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 at 0, 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 16.7.2 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.

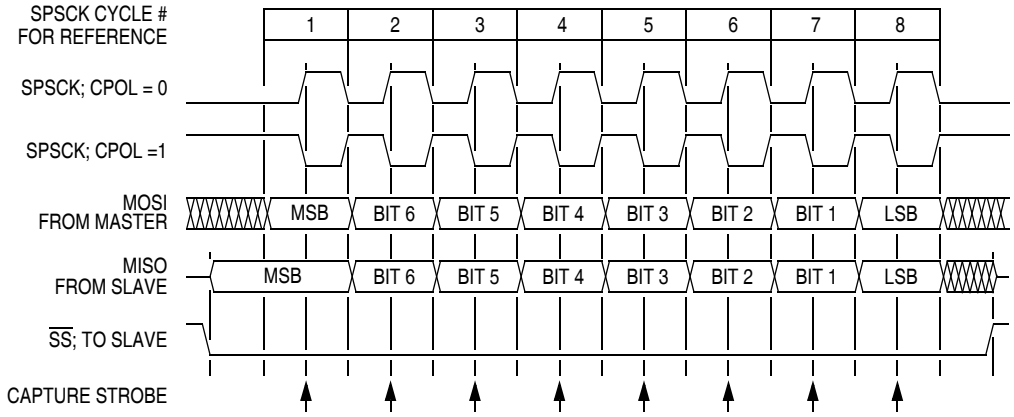


Figure 16-5. Transmission Format (CPHA = 0)

16.6 Queuing Transmission Data

The double-buffered transmit data register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag (SPTE) indicates when the transmit data buffer is ready to accept new data. Write to the transmit data register only when the SPTE bit is high. Figure 16-9 shows the timing associated with doing back-to-back transmissions with the SPI (SPSCK has CPHA: CPOL = 1:0).

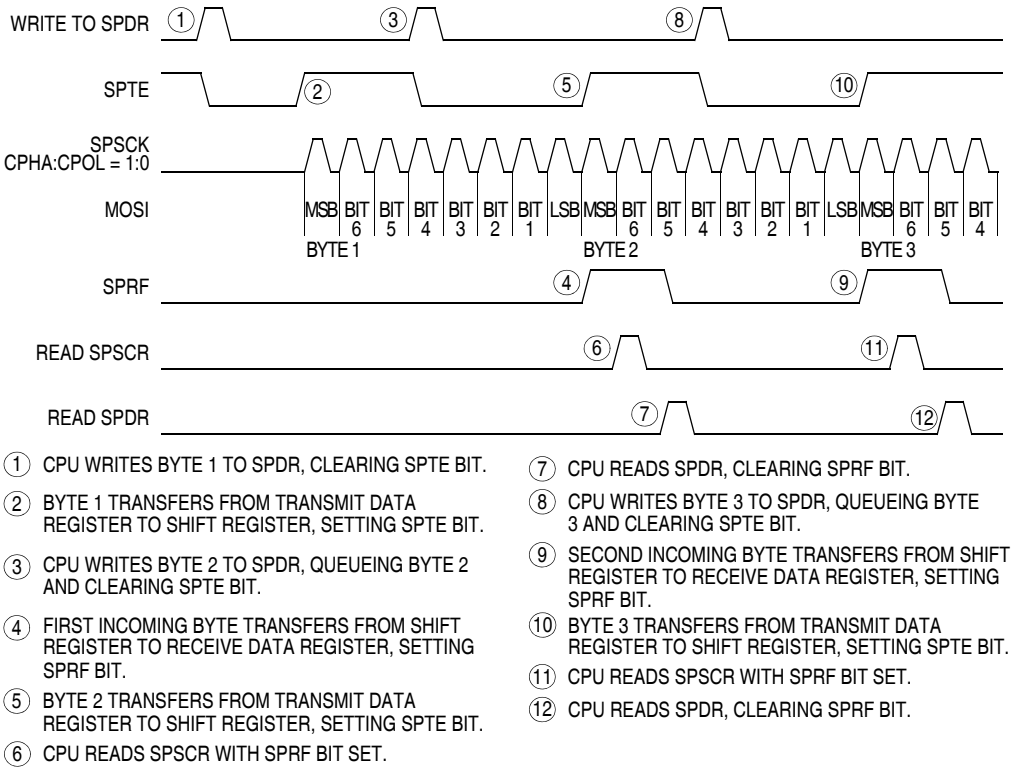


Figure 16-9. SPRF/SPTE CPU Interrupt Timing

The transmit data buffer allows back-to-back transmissions without the slave precisely timing its writes between transmissions as in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the shift register is the next data word to be transmitted.

For an idle master or idle slave that has no data loaded into its transmit buffer, the SPTE is set again no more than two bus cycles after the transmit buffer empties into the shift register. This allows the user to queue up a 16-bit value to send. For an already active slave, the load of the shift register cannot occur until the transmission is completed. This implies that a back-to-back write to the transmit data register is not possible. The SPTE indicates when the next write can occur.

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to port D, and pin PTDx/TCHx is available as a general-purpose I/O pin. Table 18-3 shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 18-3. Mode, Edge, and Level Selection

MSxB:MSxA	ELSxB:ELSxA	Mode	Configuration
X0	00	Output preset	Pin under port control; initial output level high
X1	00		Pin under port control; initial output level low
00	01	Input capture	Capture on rising edge only
00	10		Capture on falling edge only
00	11		Capture on rising or falling edge
01	01	Output compare or PWM	Toggle output on compare
01	10		Clear output on compare
01	11		Set output on compare
1X	01	Buffered output compare or buffered PWM	Toggle output on compare
1X	10		Clear output on compare
1X	11		Set output on compare

NOTE

Before enabling a TIM channel register for input capture operation, make sure that the PTD/TCHx pin is stable for at least two bus clocks.

TOVx — Toggle On Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM counter overflows. When channel x is an input capture channel, TOVx has no effect. Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM counter overflow.
0 = Channel x pin does not toggle on TIM counter overflow.

NOTE

When TOVx is set, a TIM counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 18-12 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

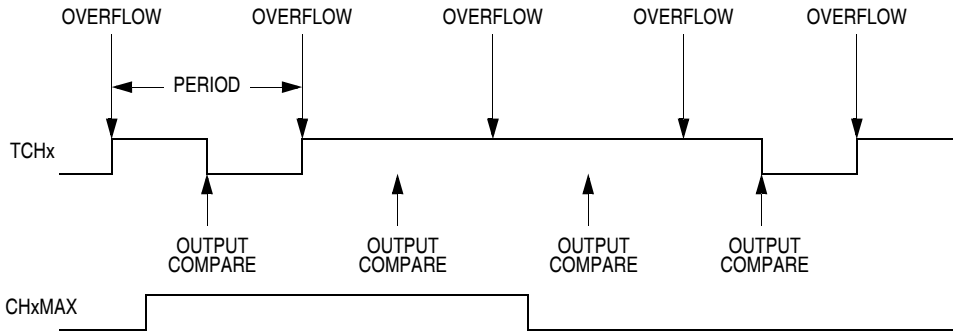


Figure 18-12. CHxMAX Latency

18.9.5 TIM Channel Registers

These read/write registers contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Address: T1CH0H, $0026 and T2CH0H, $0031

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

Figure 18-13. TIM Channel 0 Register High (TCH0H)

Address: T1CH0L, $0027 and T2CH0L $0032

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

Figure 18-14. TIM Channel 0 Register Low (TCH0L)

Address: T1CH1H, $0029 and T2CH1H, $0034

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

Figure 18-15. TIM Channel 1 Register High (TCH1H)

Address: T1CH1L, $002A and T2CH1L, $0035

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

Figure 18-16. TIM Channel 1 Register Low (TCH1L)

20.5 5.0-Vdc Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage					
(I_{Load} = –2.0 mA) all I/O pins	V_{OH}	V_{DD} – 0.8	—	—	V
(I_{Load} = –10.0 mA) all I/O pins	V_{OH}	V_{DD} – 1.5	—	—	V
(I_{Load} = –20.0 mA) pins PTC0–PTC4 only	V_{OH}	V_{DD} – 1.5	—	—	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OH1}	—	—	50	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OH2}	—	—	50	mA
Maximum total I_{OH} for all port pins	I_{OHT}	—	—	100	mA
Output low voltage					
(I_{Load} = 1.6 mA) all I/O pins	V_{OL}	—	—	0.4	V
(I_{Load} = 10 mA) all I/O pins	V_{OL}	—	—	1.5	V
(I_{Load} = 20 mA) pins PTC0–PTC4 only	V_{OL}	—	—	1.5	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OL1}	—	—	50	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OL2}	—	—	50	mA
Maximum total I_{OL} for all port pins	I_{OLT}	—	—	100	mA
Input high voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
V_{DD} supply current					
Run[3]	I_{DD}	—	20	30	mA
Wait[4]		—	6	12	mA
Stop[5]					
25°C		—	3	—	µA
25°C with TBM enabled[6]		—	20	—	µA
25°C with LVI and TBM enabled[6]		—	300	—	µA
–40°C to 125°C with TBM enabled[6]		—	50	—	µA
–40°C to 125°C with LVI and TBM enabled[6]		—	500	—	µA
DC injection current, all ports	I_{INJ}	–2	—	+2	mA
Total dc current injection (sum of all I/O)	I_{INJTOT}	–25	—	+25	mA
I/O ports Hi-Z leakage current[7]	I_{IL}	–10	—	+10	µA
Input current	I_{In}	–1	—	+1	µA
Pullup resistors (as input only) Ports PTA7/$\overline{KBD7}$–PTA0/$\overline{KBD0}$, PTC6–PTC0, PTD7/T2CH1–PTD0/$\overline{SS}$	R_{PU}	20	45	65	kΩ
Capacitance	C_{Out}	—	—	12	pF
Ports (as input or output)	C_{In}	—	—	8	

Continued on next page

20.6 3.3-Vdc Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage					
(I_{Load} = –0.6 mA) all I/O pins	V_{OH}	V_{DD} – 0.3	—	—	V
(I_{Load} = –4.0 mA) all I/O pins	V_{OH}	V_{DD} – 1.0	—	—	V
(I_{Load} = –10.0 mA) pins PTC0–PTC4 only	V_{OH}	V_{DD} – 1.0	—	—	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OH1}	—	—	30	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OH2}	—	—	30	mA
Maximum total I_{OH} for all port pins	I_{OHT}	—	—	60	mA
Output low voltage					
(I_{Load} = 1.6 mA) all I/O pins	V_{OL}	—	—	0.3	V
(I_{Load} = 10 mA) all I/O pins	V_{OL}	—	—	1.0	V
(I_{Load} = 20 mA) pins PTC0–PTC4 only	V_{OL}	—	—	0.8	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OL1}	—	—	30	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OL2}	—	—	30	mA
Maximum total I_{OL} for all port pins	I_{OLT}	—	—	60	mA
Input high voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current					
Run[3]	I_{DD}	—	8	12	mA
Wait[4]		—	3	6	mA
Stop[5]					
25°C		—	2	—	µA
25°C with TBM enabled[6]		—	12	—	µA
25°C with LVI and TBM enabled[6]		—	200	—	µA
–40°C to 125°C with TBM enabled[6]		—	30	—	µA
–40°C to 125°C with LVI and TBM enabled[6]		—	300	—	µA
DC injection current, all ports	I_{INJ}	–2	—	+2	mA
Total dc current injection (sum of all I/O)	I_{INJTOT}	–25	—	+25	mA
I/O ports Hi-Z leakage current[7]	I_{IL}	–10	—	+10	µA
Input current	I_{In}	–1	—	+1	µA
Pullup resistors (as input only) Ports PTA7/$\overline{KBD7}$–PTA0/$\overline{KBD0}$, PTC6–PTC0, PTD7/T2CH1–PTD0/$\overline{SS}$	R_{PU}	20	45	65	kΩ
Capacitance	C_{Out}	—	—	12	pF
Ports (as input or output)	C_{In}	—	—	8	

Continued on next page

20.7 5.0-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option[2]	f_{OSC}	 32 dc	 100 32.8	 kHz MHz
Internal operating frequency	f_{OP} (f_{Bus})	—	8.2	MHz
Internal clock period ($1/f_{OP}$)	t_{CYC}	122	—	ns
RESET input pulse width low[3]	t_{IRL}	50	—	ns
$\overline{IRQ}$ interrupt pulse width low[4] (edge-triggered)	t_{ILIH}	50	—	ns
$\overline{IRQ}$ interrupt pulse period	t_{ILIL}	Note 5	—	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. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
4. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.

20.8 3.3-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option[2]	f_{OSC}	 32 dc	 100 16.4	 kHz MHz
Internal operating frequency	f_{OP} (f_{Bus})	—	4.1	MHz
Internal clock period ($1/f_{OP}$)	t_{CYC}	244	—	ns
RESET input pulse width low[3]	t_{IRL}	125	—	ns
$\overline{IRQ}$ interrupt pulse width low[4] (edge-triggered)	t_{ILIH}	125	—	ns
$\overline{IRQ}$ interrupt pulse period	t_{ILIL}	Note 5	—	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. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
4. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.

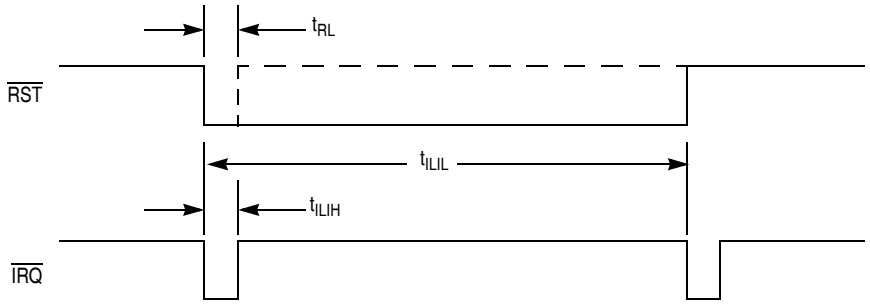
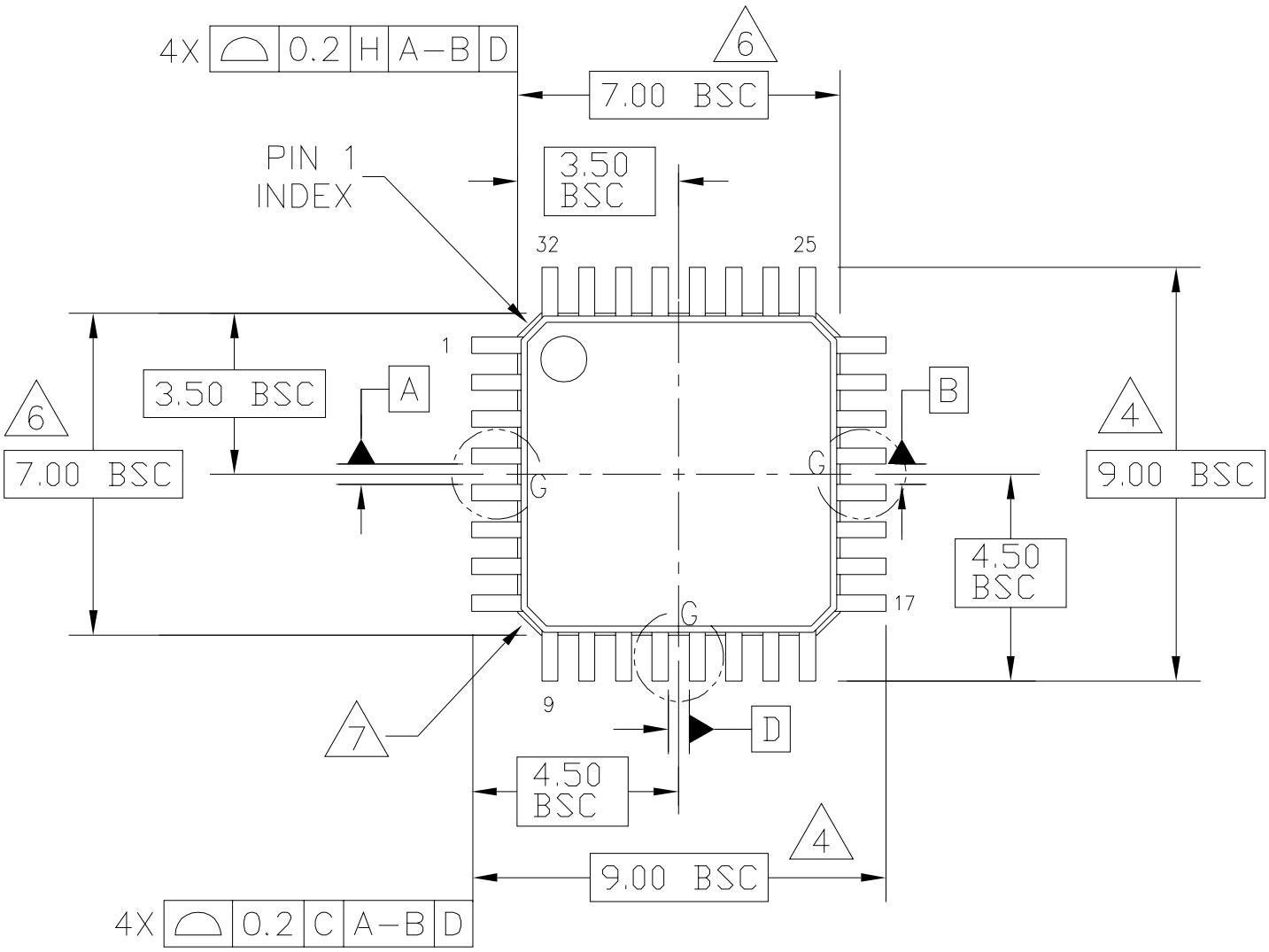
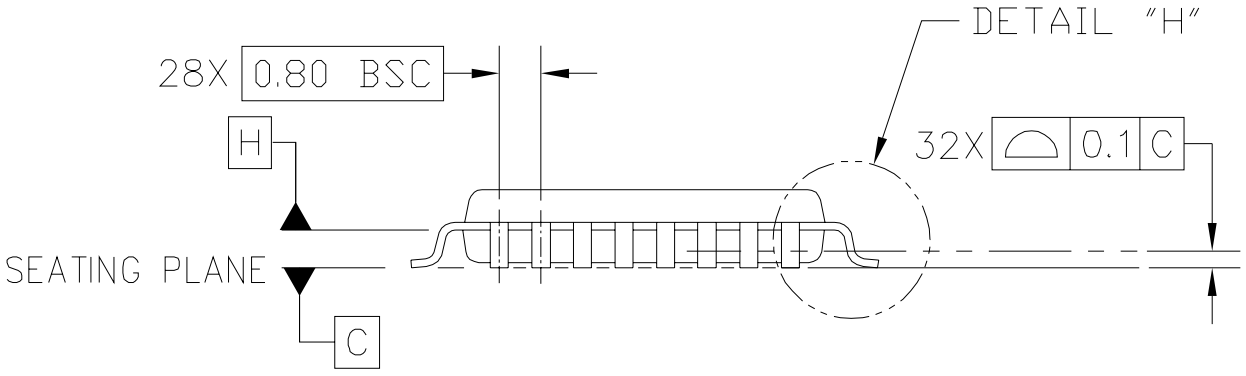


Figure 20-1. $\overline{RST}$ and $\overline{IRQ}$ Timing





MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASH70029A	REV: C
LOW PROFILE QUAD FLAT PACK (LQFP) 32 LEAD, 0.8 PITCH (7 X 7 X 1.4)	CASE NUMBER: 873A-04	01 APR 2005
	STANDARD: JEDEC MS-026 BBA	