



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

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

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LED, LVD, POR, PWM
Number of I/O	15
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.3V
Data Converters	A/D 12x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908jk3cdwe

15.4	I/O Signals	175
15.4.1	2OSCOU	175
15.4.2	COPCTL Write	175
15.4.3	Power-On Reset	175
15.4.4	Internal Reset	175
15.4.5	Reset Vector Fetch	176
15.4.6	COPD (COP Disable)	176
15.4.7	COPRS (COP Rate Select)	176
15.5	COP Control Register	177
15.6	Interrupts	177
15.7	Monitor Mode	177
15.8	Low-Power Modes	177
15.8.1	Wait Mode	177
15.8.2	Stop Mode	178
15.9	COP Module During Break Mode	178

Section 16. Low Voltage Inhibit (LVI)

16.1	Contents	179
16.2	Introduction	179
16.3	Features	179
16.4	Functional Description	180
16.5	LVI Control Register (CONFIG2/CONFIG1)	180
16.6	Low-Power Modes	181
16.6.1	Wait Mode	181
16.6.2	Stop Mode	181

Section 17. Break Module (BREAK)

17.1	Contents	183
17.2	Introduction	183
17.3	Features	184

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$002B	Unimplemented	Read:								
\$003B		Write:								
\$003C	ADC Status and Control Register (ADSCR)	Read:	COCO	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
\$003D	ADC Data Register (ADR)	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:								
		Reset:	Indeterminate after reset							
\$003E	ADC Input Clock Register (ADICLK)	Read:	ADIV2	ADIV1	ADIV0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$003F	Unimplemented	Read:								
		Write:								
\$FE00	Break Status Register (BSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							See note	
		Reset:	0							
Note: Writing a logic 0 clears SBSW.										
\$FE01	Reset Status Register (RSR)	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
\$FE02	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
\$FE03	Break Flag Control Register (BFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1)	Read:	0	IF5	IF4	IF3	0	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
				= Unimplemented			R	= Reserved		

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	0	IF15
		Write:	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0
\$FE07	Reserved	Read:							
		Write:	R	R	R	R	R	R	R
\$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE09	FLASH Block Protect Register (FLBPR)	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0A ↓ \$FE0B	Reserved	Read:							
		Write:	R	R	R	R	R	R	R
\$FE0C	Break Address High Register (BRKH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0D	Break Address low Register (BRKL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector						
		Write:	Writing clears COP counter (any value)						
		Reset:	Unaffected by reset						

= Unimplemented
 R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 5)

Section 4. FLASH Memory (FLASH)

4.1 Contents

- 4.2 Introduction39
- 4.3 Functional Description40
- 4.4 FLASH Control Register40
- 4.5 FLASH Block Erase Operation41
- 4.6 FLASH Mass Erase Operation42
- 4.7 FLASH Program Operation.43
- 4.8 FLASH Protection44
- 4.9 FLASH Block Protect Register46

4.2 Introduction

This section describes the operation of the embedded FLASH memory. The FLASH memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

MC68H(R)C908JL3/JK3: 4096 bytes user FLASH from \$EC00 – \$FBFF.
MC68H(R)C908JK1: 1536 bytes user FLASH from \$F600 – \$FBFF.

4.7 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses \$XX00, \$XX20, \$XX40, \$XX60, \$XX80, \$XXA0, \$XXC0 or \$XXE0. Use this step-by-step procedure to program a row of FLASH memory: (Figure 4-2 shows a flowchart of the programming algorithm.)

NOTE: *In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.*

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH location within the address range of the row to be programmed.
3. Wait for a time, t_{nvs} (10 μ s).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (5 μ s).
6. Write data to the byte being programmed.
7. Wait for time, t_{prog} (30 μ s).
8. Repeat step 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, t_{nvh} (5 μ s).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1 μ s), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

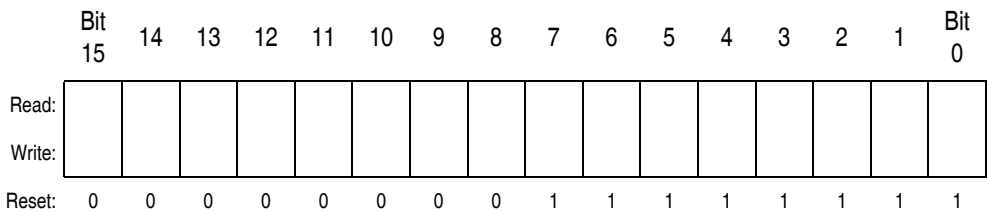


Figure 6-4. Stack Pointer (SP)

NOTE: The location of the stack is arbitrary and may be relocated anywhere in RAM. Moving the SP out of page 0 (\$0000 to \$00FF) frees direct address (page 0) space. For correct operation, the stack pointer must point only to RAM locations.

6.4.4 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

During reset, the program counter is loaded with the reset vector address located at \$FFFE and \$FFFF. The vector address is the address of the first instruction to be executed after exiting the reset state.

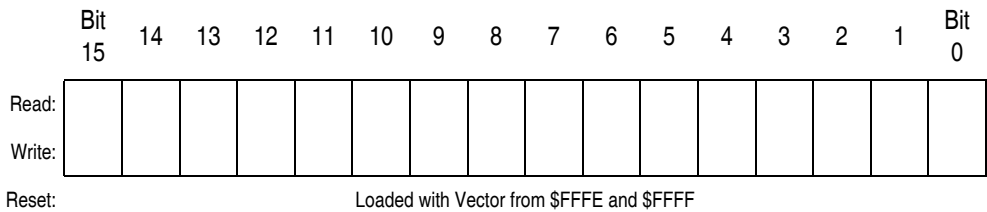


Figure 6-5. Program Counter (PC)

6.4.5 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

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

7.4.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 2OSCOUT cycles. Sixty-four 2OSCOUT cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, the following events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables the oscillator to drive 2OSCOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 2OSCOUT cycles to allow stabilization of the oscillator.
- The $\overline{\text{RST}}$ pin is driven low during the oscillator stabilization time.
- The POR bit of the reset status register (RSR) is set and all other bits in the register are cleared.

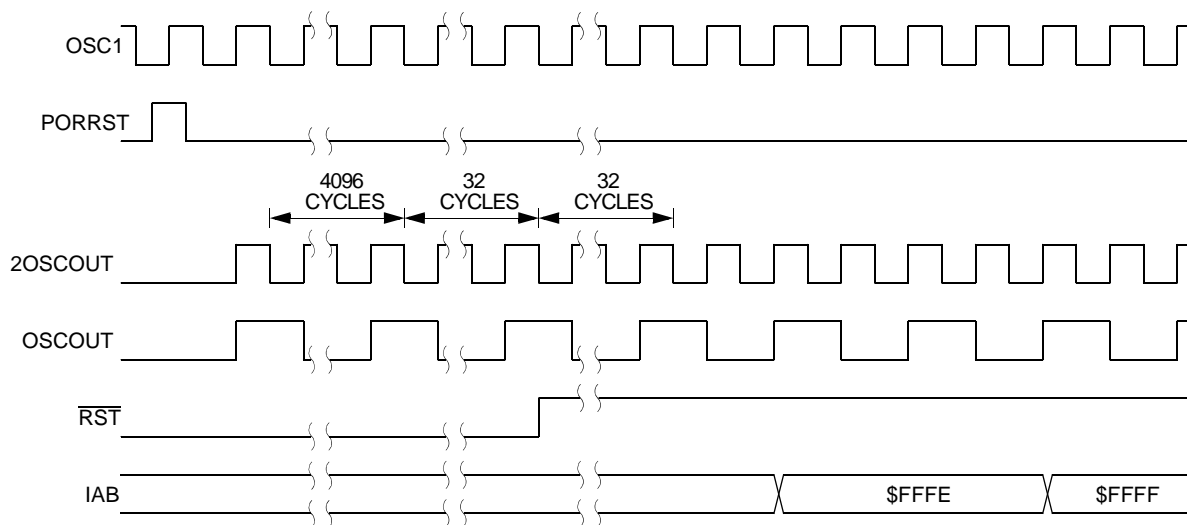


Figure 7-7. POR Recovery

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. **Figure 7-9** shows interrupt entry timing. **Figure 7-10** shows interrupt recovery timing.

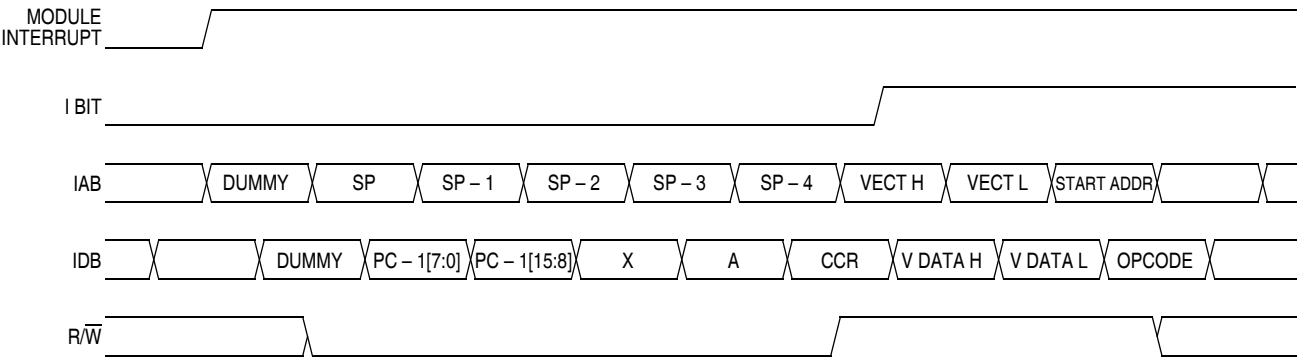


Figure 7-9. Interrupt Entry

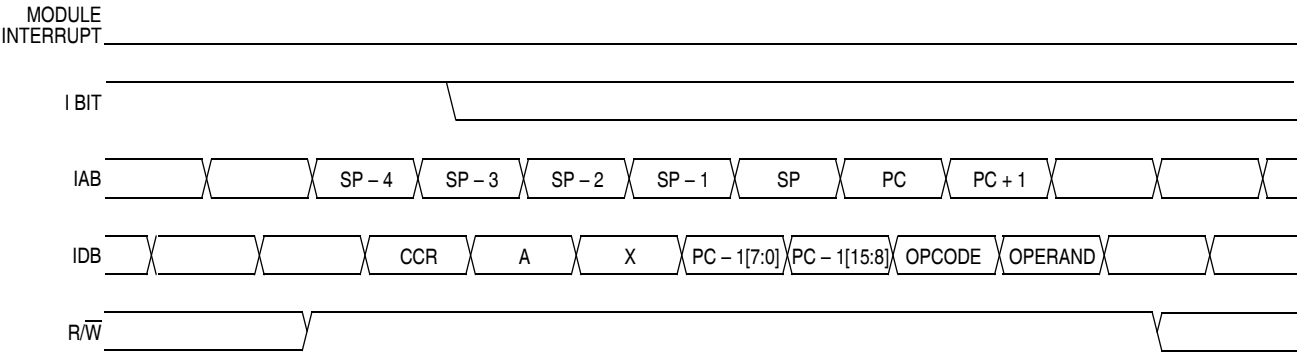


Figure 7-10. Interrupt Recovery

7.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register), and if the corresponding interrupt enable bit is

(TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE: *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. Writing to the active channel registers is the same as generating unbuffered PWM signals.*

10.5.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
 - a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
 - b. Reset the TIM counter by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
 - a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. (See [Table 10-3](#).)
 - b. Write 1 to the toggle-on-overflow bit, TOVx.
 - c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. (See [Table 10-3](#).)

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 can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Table 10-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal Bus Clock ÷ 1
0	0	1	Internal Bus Clock ÷ 2
0	1	0	Internal Bus Clock ÷ 4
0	1	1	Internal Bus Clock ÷ 8
1	0	0	Internal Bus Clock ÷ 16
1	0	1	Internal Bus Clock ÷ 32
1	1	0	Internal Bus Clock ÷ 64
1	1	1	Not available

10.10.2 TIM Counter Registers (TCNTH:TCNTL)

The two read-only TIM counter registers contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

NOTE: *If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.*

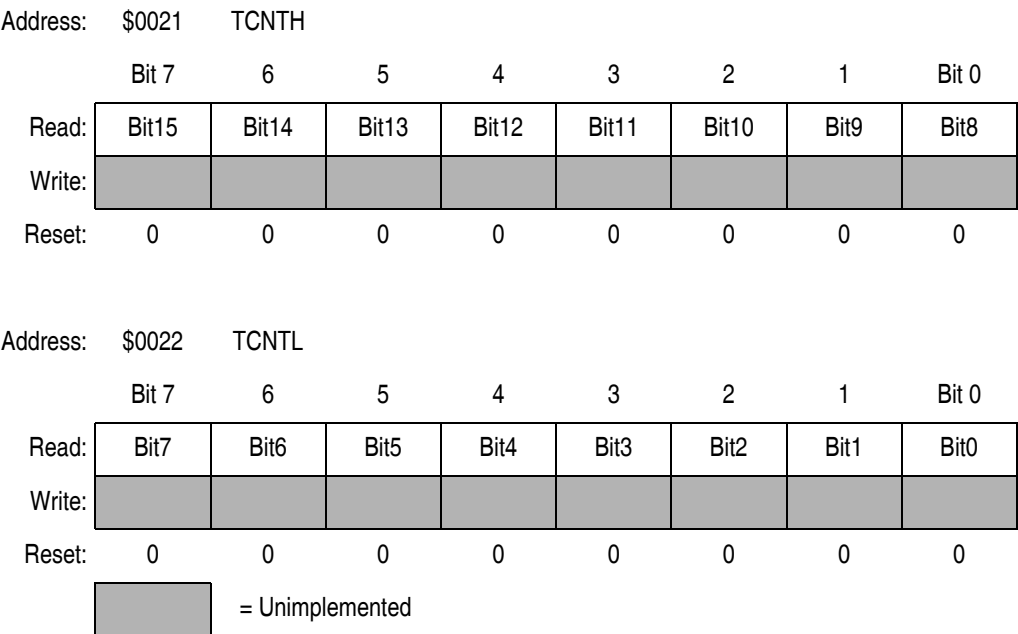


Figure 10-5. TIM Counter Registers (TCNTH:TCNTL)

10.10.3 TIM Counter Modulo Registers (TMODH:TMODL)

The read/write TIM modulo registers contain the modulo value for the TIM counter. When the TIM counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM counter resumes counting from \$0000 at the next clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.

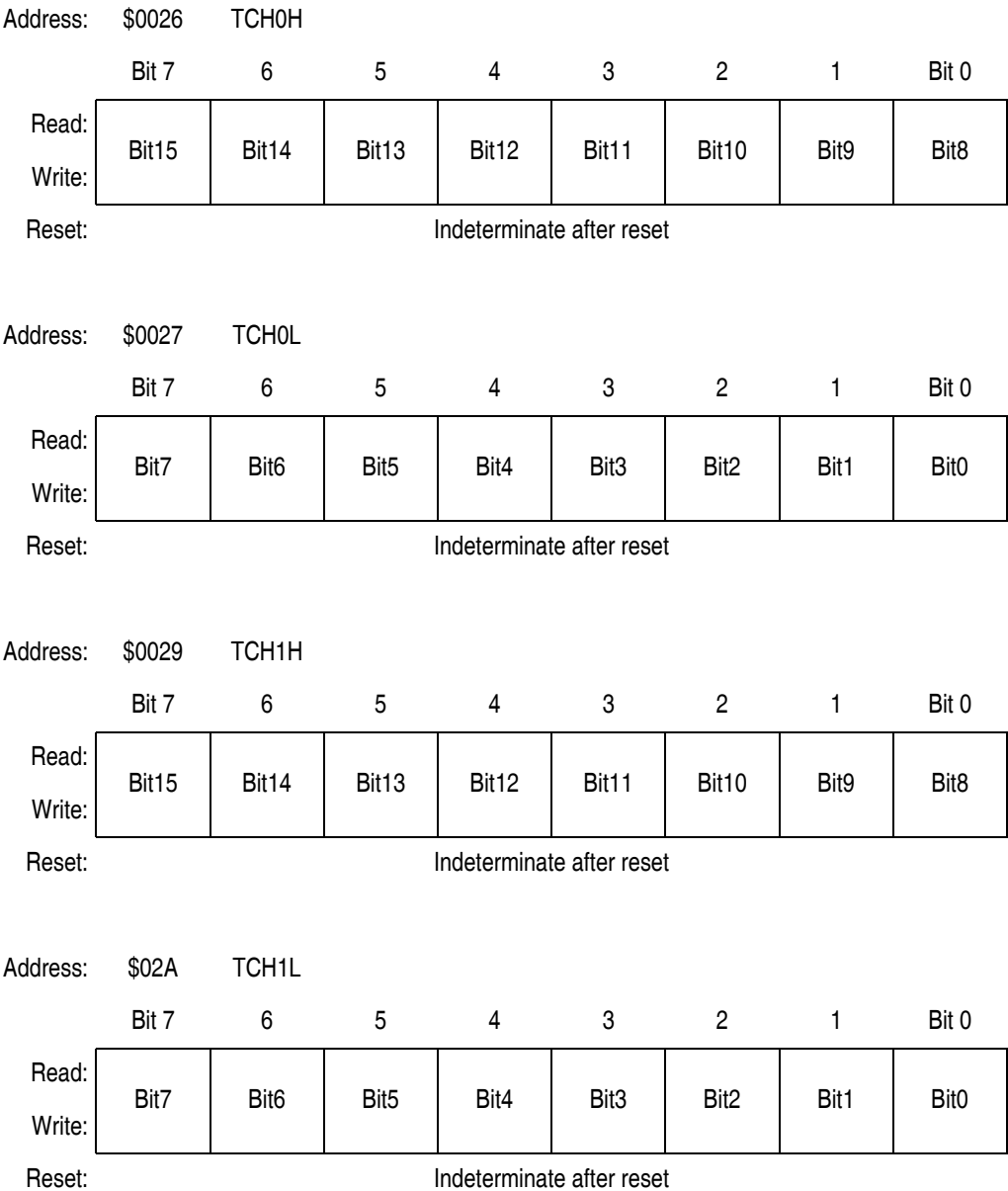


Figure 10-9. TIM Channel Registers (TCH0H/L:TCH1H/L)

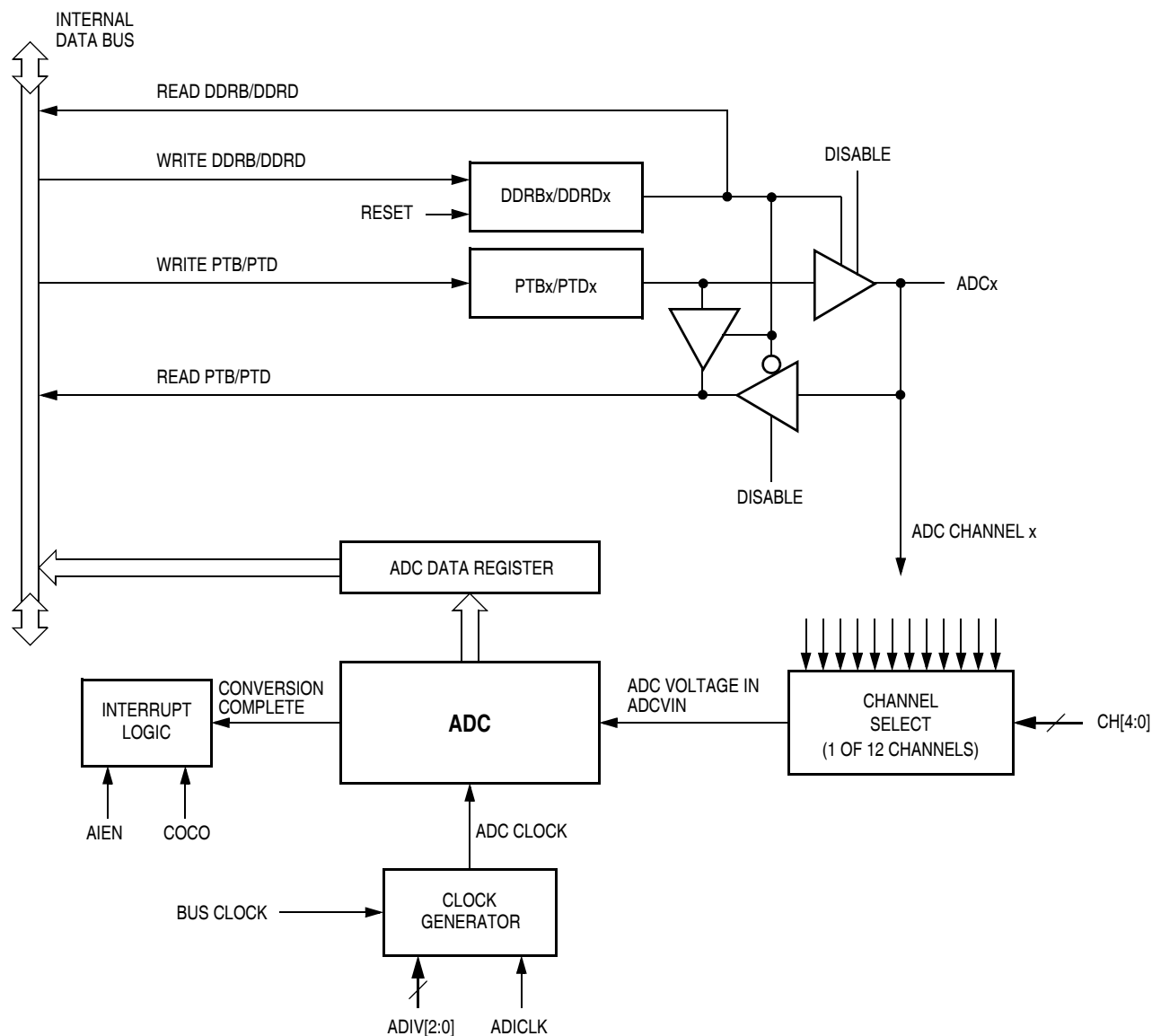


Figure 11-2. ADC Block Diagram

11.4.1 ADC Port I/O Pins

PTB0–PTB7 and PTD0–PTD3 are general-purpose I/O pins that are shared with the ADC channels. The channel select bits (ADC Status and Control register, \$003C), 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.

11.7.1 ADC Voltage In (ADCVIN)

ADCVIN is the input voltage signal from one of the 12 ADC channels to the ADC module.

11.8 I/O Registers

These I/O registers control and monitor ADC operation:

- ADC Status and Control register (ADSCR)
- ADC data register (ADR)
- ADC clock register (ADICLK)

11.8.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC Status and Control register.

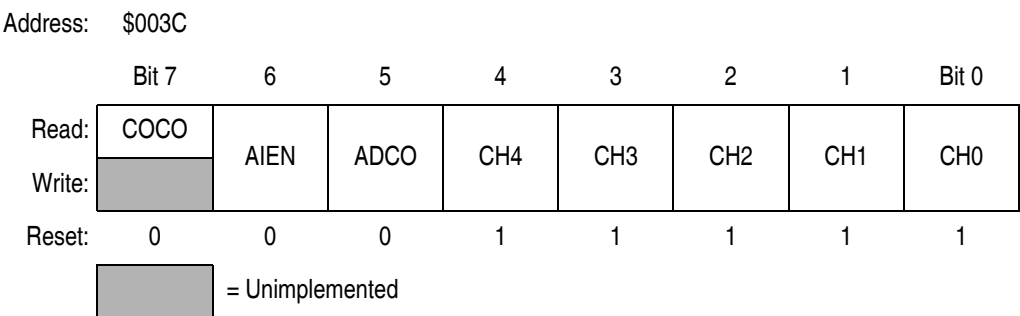


Figure 11-3. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

When the AIEN bit is a logic 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the ADC status and control register is written or whenever the ADC data register is read. Reset clears this bit.

- 1 = conversion completed (AIEN = 0)
- 0 = conversion not completed (AIEN = 0)

When the AIEN bit is a logic 1 (CPU interrupt enabled), the COCO is a read-only bit, and will always be logic 0 when read.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0000	Port A Data Register (PTA)	Read:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
\$0004	Data Direction Register A (DDRA)	Read:	0	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000A	Port D Control Register (PDCR)	Read:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000D	Port A Input Pull-up Enable Register (PTAPUE)	Read:	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Figure 12-1. I/O Port Register Summary

12.3 Port A

Port A is an 7-bit special function port that shares all seven of its pins with the Keyboard Interrupt (KBI) Module, **See Section 14**. Each port A pin also has software configurable pull-up device if the corresponding port pin is configured as input port. PTA0 to PTA5 has direct LED drive capability.



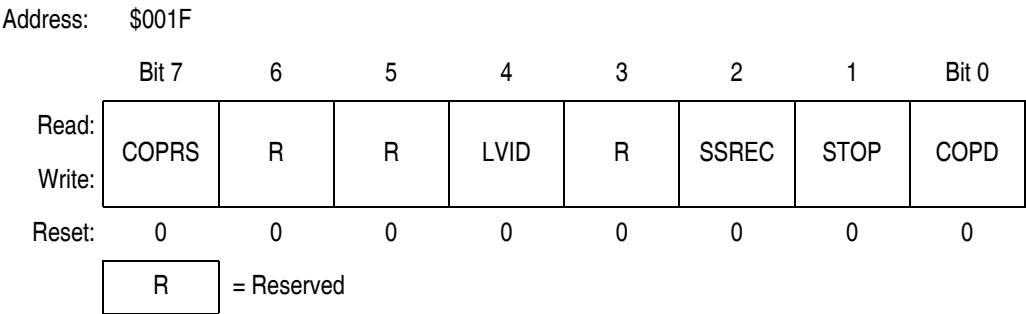


Figure 16-3. Configuration Register 1 (CONFIG1)

LVID —pLow Voltage Inhibit Disable Bit
 1 = Low voltage inhibit disabled
 0 = Low voltage inhibit enabled

LVIT1, LVIT0 — LVI Trip Voltage Selection

These two bits determine at which level of V_{DD} the LVI module will come into action. LVIT1 and LVIT0 are cleared by a Power-On Reset only.

LVIT1	LVIT0	Trip Voltage ⁽¹⁾	Comments
0	0	V_{LVR3} (2.4V)	For V_{DD} =3V operation
0	1	V_{LVR3} (2.4V)	For V_{DD} =3V operation
1	0	V_{LVR5} (4.0V)	For V_{DD} =5V operation
1	1	Reserved	

1. See [Section 18. Electrical Specifications](#) for full parameters.

16.6 Low-Power Modes

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

16.6.1 Wait Mode

The LVI module, when enabled, will continue to operate in WAIT Mode.

16.6.2 Stop Mode

The LVI module, when enabled, will continue to operate in STOP Mode.

17.3 Features

Features of the break module include the following:

- Accessible I/O registers during the break Interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

17.4 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 SIM. The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. 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 logic one 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 begins after the CPU completes its current instruction. A return from interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation. [Figure 17-1](#) shows the structure of the break module.

18.13 ADC Characteristics

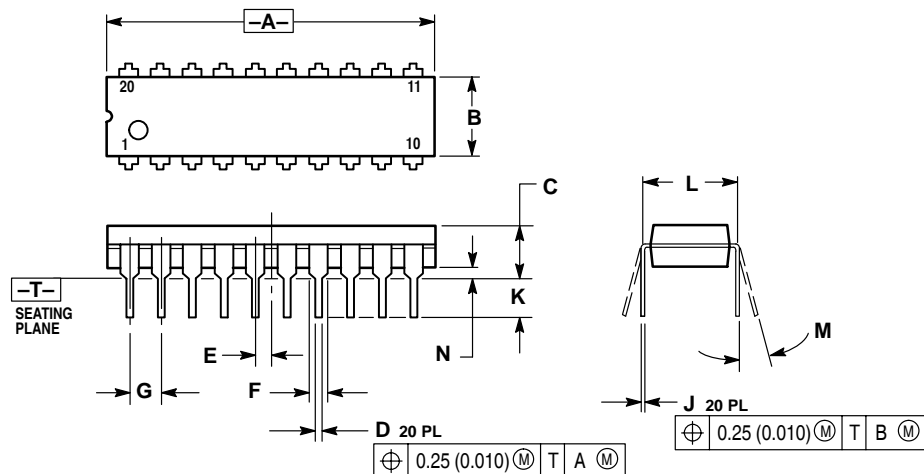
Table 18-10. ADC Characteristics

Characteristic	Symbol	Min	Max	Unit	Comments
Supply voltage	V_{DDAD}	2.7 (V_{DD} min)	5.5 (V_{DD} max)	V	
Input voltages	V_{ADIN}	V_{SS}	V_{DD}	V	
Resolution	B_{AD}	8	8	Bits	
Absolute accuracy	A_{AD}	± 0.5	± 1.5	LSB	Includes quantization
ADC internal clock	f_{ADIC}	0.5	1.048	MHz	$t_{AIC} = 1/f_{ADIC}$, tested only at 1 MHz
Conversion range	R_{AD}	V_{SS}	V_{DD}	V	
Power-up time	t_{ADPU}	16		t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time ⁽¹⁾	t_{ADS}	5	—	t_{AIC} cycles	
Zero input reading ⁽²⁾	Z_{ADI}	00	01	Hex	$V_{IN} = V_{SS}$
Full-scale reading ⁽³⁾	F_{ADI}	FE	FF	Hex	$V_{IN} = V_{DD}$
Input capacitance	C_{ADI}	—	(20) 8	pF	Not tested
Input leakage ⁽³⁾ Port B/port D	—	—	± 1	μA	

NOTES:

1. Source impedances greater than 10 k Ω adversely affect internal RC charging time during input sampling.
2. Zero-input/full-scale reading requires sufficient decoupling measures for accurate conversions.
3. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

19.3 20-Pin PDIP

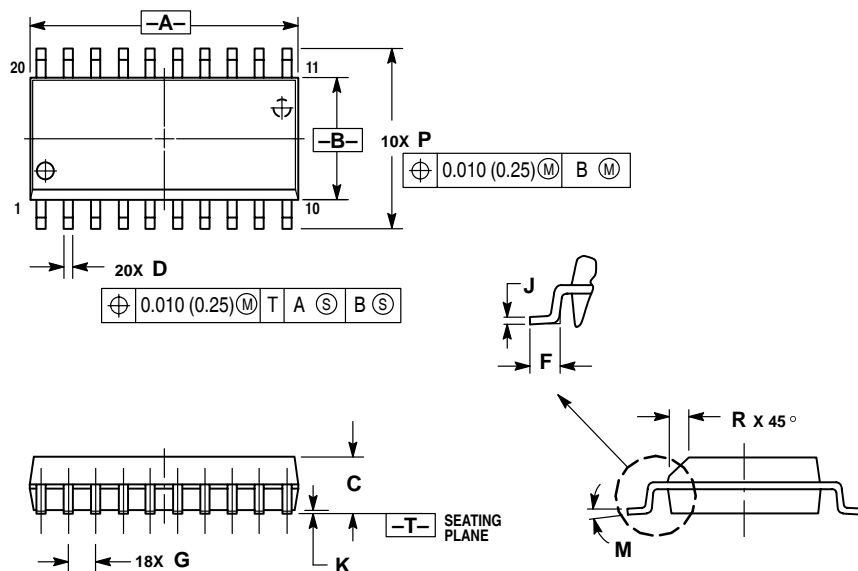


- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
 2. CONTROLLING DIMENSION: INCH.
 3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
 4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.010	1.070	25.66	27.17
B	0.240	0.260	6.10	6.60
C	0.150	0.180	3.81	4.57
D	0.015	0.022	0.39	0.55
E	0.050 BSC		1.27 BSC	
F	0.050	0.070	1.27	1.77
G	0.100 BSC		2.54 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.140	2.80	3.55
L	0.300 BSC		7.62 BSC	
M	0° 15°		0° 15°	
N	0.020	0.040	0.51	1.01

Figure 19-1. 20-Pin PDIP (Case #738)

19.4 20-Pin SOIC



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
 2. CONTROLLING DIMENSION: MILLIMETER.
 3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
 4. MAXIMUM MOLD PROTRUSION 0.150 (0.006) PER SIDE.
 5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	12.65	12.95	0.499	0.510
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.50	0.90	0.020	0.035
G	1.27 BSC		0.050 BSC	
J	0.25	0.32	0.010	0.012
K	0.10	0.25	0.004	0.009
M	0° 7°		0° 7°	
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Figure 19-2. 20-Pin SOIC (Case #751D)