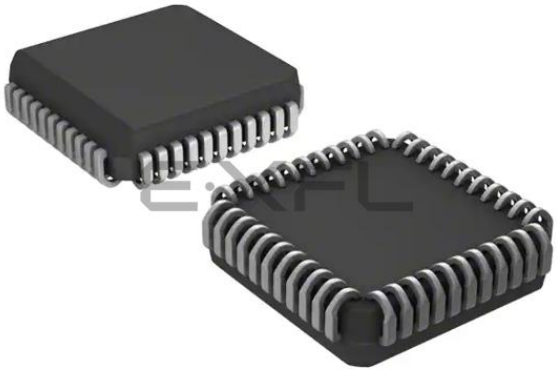


Welcome to E-XFL.COM

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	HC05
Core Size	8-Bit
Speed	2.1MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	304 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc705c8avfne

Section 13. Electrical Specifications

Section 14. Mechanical Specifications

3.3.5 Condition Code Register

The condition code register (CCR) shown in **Figure 3-6** is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four bits that indicate the results of prior instructions.

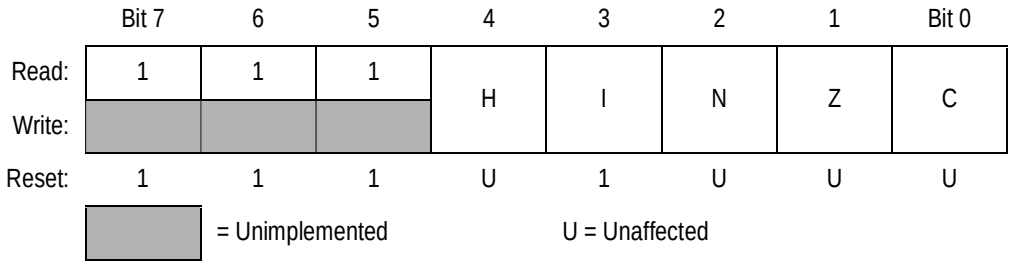


Figure 3-6. Condition Code Register (CCR)

H — Half-Carry Bit

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an add without carry (ADD) or add with carry (ADC) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations. Reset has no affect on the half-carry flag.

I — Interrupt Mask Bit

Setting the interrupt mask (I) disables interrupts. If an interrupt request occurs while the interrupt mask is a logic 0, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. The CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After a reset, the interrupt mask is set and can be cleared only by a CLI, STOP, or WAIT instruction.

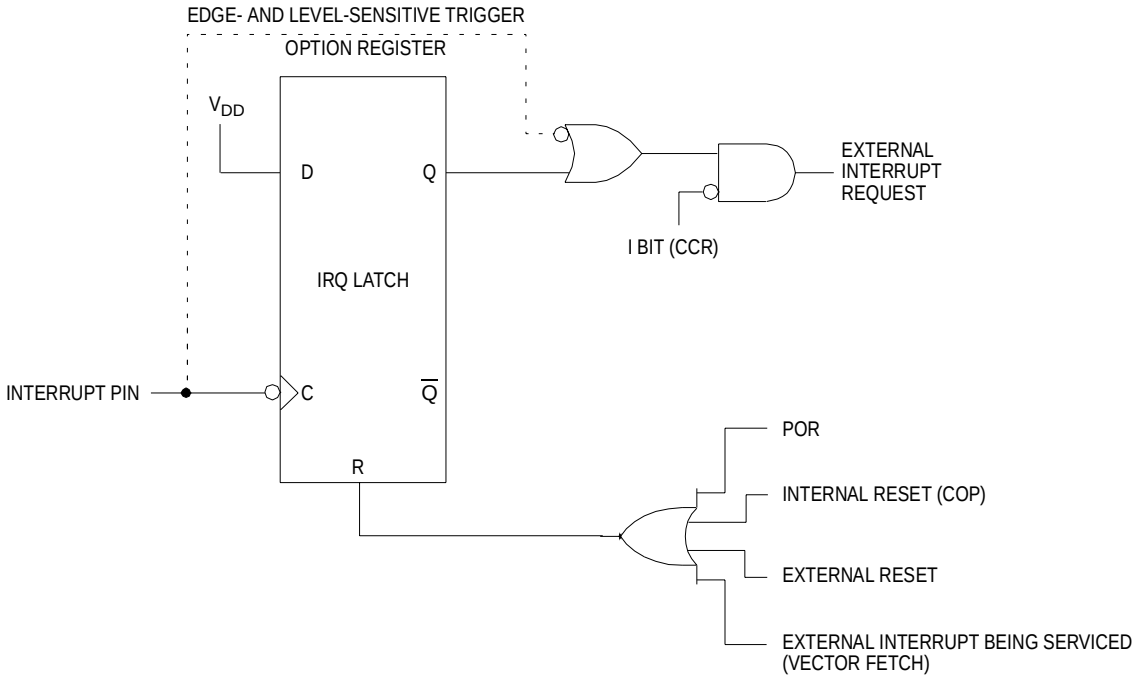
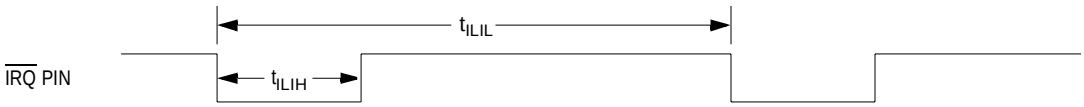
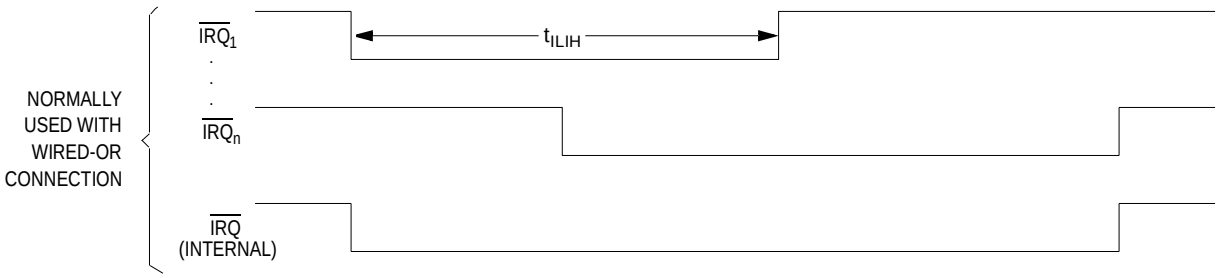


Figure 4-1. External Interrupt Internal Function Diagram



a. Edge-Sensitive Trigger Condition. The minimum pulse width (t_{ILIH}) is either 125 ns (f_{OP} = 2.1 MHz) or 250 ns (f_{OP} = 1 MHz). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 19 t_{CYC} cycles.



b. Level-Sensitive Trigger Condition. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

Figure 4-2. External Interrupt Timing

6.3.4 Non-Programmable COP Watchdog in Stop Mode

The STOP instruction has these effects on the non-programmable COP watchdog:

- Turns off the oscillator and the COP watchdog counter
- Clears the COP watchdog counter

If the $\overline{\text{RESET}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The reset function clears the COP counter again after the 4064-t_{CYC} clock stabilization delay.

If the $\overline{\text{IRQ}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The IRQ function does not clear the COP counter again after the 4064-t_{CYC} clock stabilization delay. See **Figure 6-3**.

NOTE: *If the clock monitor is enabled (CME = 1), the STOP instruction causes it to time out and reset the MCU.*

6.4 Wait Mode

The WAIT instruction places the MCU in an intermediate power consumption mode. All central processor unit (CPU) activity is suspended, but the oscillator, capture/compare timer, SCI, and SPI remain active. Any interrupt or reset brings the MCU out of wait mode. See **Figure 6-1**.

The WAIT instruction has these effects on the CPU:

- Clears the I bit in the condition code register, enabling interrupts
- Stops the CPU clock, but allows the internal clock to drive the capture/compare timer, SCI, and SPI

The WAIT instruction does not affect any other registers or I/O lines. The capture/compare timer, SCI, and SPI can be enabled to allow a periodic exit from wait mode.

7.3.2 Data Direction Register A

The contents of data direction register A (DDRA) shown in **Figure 7-2** determine whether each port A pin is an input or an output. Writing a logic 1 to a DDRA bit enables the output buffer for the associated port A pin; a logic 0 disables the output buffer. A reset clears all DDRA bits, configuring all port A pins as inputs.

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 7-2. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Port A Data Direction Bits

These read/write bits control port A data direction. Reset clears bits DDRA7–DDRA0.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE: *Avoid glitches on port A pins by writing to the port A data register before changing DDRA bits from logic 0 to logic 1.*

7.4.2 Data Direction Register B

The contents of data direction register B (DDRB) shown in **Figure 7-5** determine whether each port B pin is an input or an output. Writing a logic 1 to a DDRB bit enables the output buffer for the associated port B pin; a logic 0 disables the output buffer. A reset clears all DDRB bits, configuring all port B pins as inputs. If the pullup devices are enabled by mask option, setting a DDRB bit to a logic 1 turns off the pullup device for that pin.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 7-5. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Port B Data Direction Bits

These read/write bits control port B data direction. Reset clears bits DDRB7–DDRB0.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE: *Avoid glitches on port B pins by writing to the port B data register before changing DDRB bits from logic 0 to logic 1.*

Section 8. Capture/Compare Timer

8.1 Contents

8.2 Introduction

This section describes the operation of the 16-bit capture/compare timer. **Figure 8-1** shows the structure of the timer module. **Figure 8-2** is a summary of the timer input/output (I/O) registers.

8.3 Timer Operation

The core of the capture/compare timer is a 16-bit free-running counter. The counter is the timing reference for the input capture and output compare functions. The input capture and output compare functions can latch the times at which external events occur, measure input waveforms, and generate output waveforms and timing delays. Software can read the value in the counter at any time without affecting the counter sequence.

No LED is illuminated during this routine. Further, the end of the routine does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the green LED does not light, the PROM has been secured properly.

9.4.6 Load Program into RAM and Execute

In the load program in RAM and execute routine, user programs are loaded via the SCI port and then executed. Data is loaded sequentially starting at address $0050. After the last byte is loaded, control is transferred to the RAM program starting at $0051. The first byte loaded is the count of the total number of bytes in the program plus the count byte. The program starts at location $0051 in RAM. During initialization, the SCI is configured for eight data bits and one stop bit. The baud rate is 4800 with a 2-MHz crystal or 9600 with a 4-MHz crystal.

To load a program into RAM and execute it, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3, S5, and S6 in the OFF position.
4. Set switch S4 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).

The downloaded program starts executing as soon as the last byte is received by the SCI.

Execution of the routine can be held off by setting the byte count in the count byte (the first byte loaded) to a value greater than the number of bytes to be loaded. After loading the last byte, the firmware waits for more data. Program execution does not begin. At this point, placing switch 2 in the RESET position resets the MCU with the RAM data intact. Any other routine can be entered, including the one to execute the program in RAM, simply by setting switches S3–S6 as necessary to select the desired routine, then setting switch 2 in the OUT position.

9.4.7 Execute Program in RAM

This routine allows the MCU to transfer control to a program previously loaded in RAM. This program is executed once bootstrap mode is entered, if switch S6 is in the ON position and switch 2 is in the OUT position, without any firmware initialization. The program must start at location $0051 to be compatible with the load program in RAM routine.

To run the execute program in RAM routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switch S6 in the OFF position.
4. Switches S3, S4, and S5 can be in either position.
5. Set switch 2 in the OUT position (routine is activated).

NOTE: *The non-programmable watchdog COP is disabled in bootloader mode, even if the NCOPE bit is programmed.*

9.4.8 Dump PROM Contents

In the dump PROM contents routine, the PROM contents are dumped sequentially to the SCI output, provided the PROM has not been secured. The first location sent is $0020 and the last location sent is $1FFF. Unused locations are skipped so that no gaps exist in the data stream. The external memory address lines indicate the current location being sent. Data is sent with eight data bits and one stop bit at 4800 baud with a 2-MHz crystal or 9600 baud with a 4-MHz crystal.

To run the dump PROM contents routine, take these steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3 and S6 in the OFF position.
4. Set switches S4 and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).
6. Once PROM dumping is complete, set switch 2 in the RESET position.

SPI — SPI Enable Bit

This read/write bit enables the SPI. Reset clears the SPE bit.

1 = SPI enabled
0 = SPI disabled

MSTR — Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset clears the MSTR bit.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the PD4/SCK pin between transmissions. To transmit data between SPIs, the SPIs must have identical CPOL bits. Reset has no effect on the CPOL bit.

1 = PD4/SCK pin at logic 1 between transmissions
0 = PD4/SCK pin at logic 0 between transmissions

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPIs, the SPIs must have identical CPHA bits. When CPHA = 0, the PD5/$\overline{SS}$ pin of the slave SPI must be set to logic 1 between bytes. Reset has no effect on the CPHA bit.

1 = Edge following first active edge on PD4/SCK latches data
0 = First active edge on PD4/SCK latches data

SPR1 and SPR0 — SPI Clock Rate Bits

These read/write bits select the master mode serial clock rate, as shown in **Table 11-1**. The SPR1 and SPR0 bits of a slave SPI have no effect on the serial clock. Reset has no effect on SPR1 and SPR0.

Table 11-1. SPI Clock Rate Selection

SPR[1:0]	SPI Clock Rate
00	Internal Clock ÷ 2
01	Internal Clock ÷ 4
10	Internal Clock ÷ 16
11	Internal Clock ÷ 32

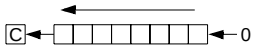
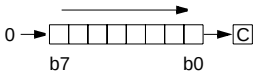
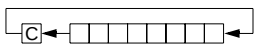
12.4.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear carry bit	CLC
Clear interrupt mask	CLI
No operation	NOP
Reset stack pointer	RSP
Return from interrupt	RTI
Return from subroutine	RTS
Set carry bit	SEC
Set interrupt mask	SEI
Stop oscillator and enable $\overline{\text{IRQ}}$ pin	STOP
Software interrupt	SWI
Transfer accumulator to index register	TAX
Transfer index register to accumulator	TXA
Stop CPU clock and enable interrupts	WAIT

Table 12-6. Instruction Set Summary (Sheet 4 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		1 1
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5

13.6 Power Considerations

The average chip junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad (1)$$

Where:

T_A = ambient temperature in °C
θ_{JA} = package thermal resistance, junction to ambient in °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$ = chip internal power dissipation
$P_{I/O}$ = power dissipation on input and output pins (user-determined)

For most applications, $P_{I/O} < P_{INT}$ and can be neglected.

Ignoring $P_{I/O}$, the relationship between P_D and T_J is approximately:

$$P_D = \frac{K}{T_J + 273°C} \quad (2)$$

Solving equations (1) and (2) for K gives:

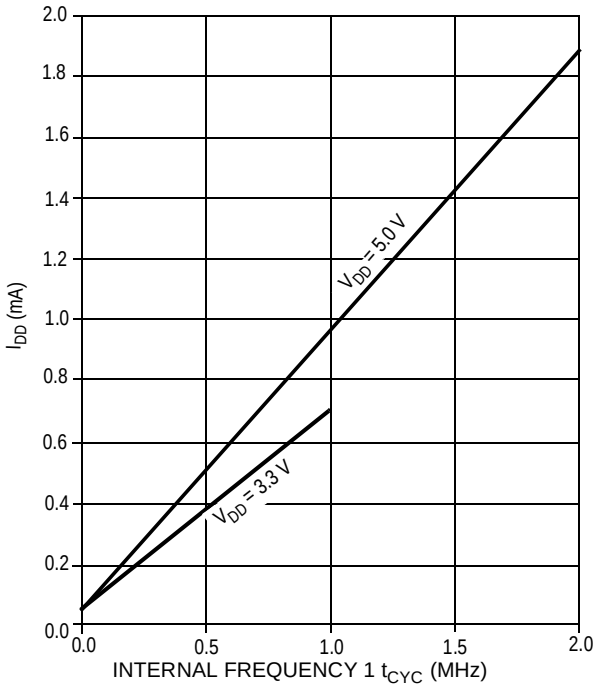
$$= P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

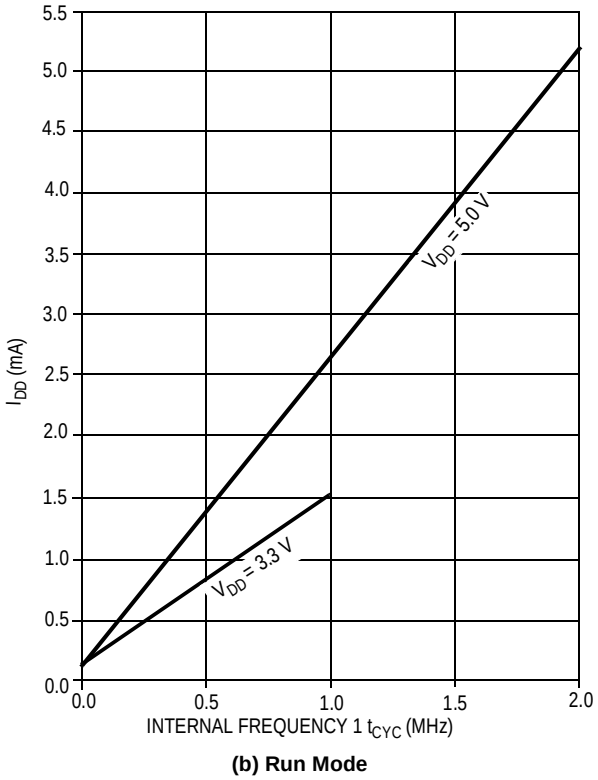
13.7 5.0-Volt DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output high voltage I_{Load} = –0.8 mA, PA7–PA0, PB7–PB0, PC6–PC0, TCMP (see **Figure 13-2**) I_{Load} = –1.6 mA, PD4–PD1 (see **Figure 13-3**) I_{Load} = –5.0 mA, PC7	V_{OH}	$V_{DD} - 0.8$	— — —	— — —	V
Output low voltage (see **Figure 13-4**) I_{Load} = 1.6 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{Load} = 20 mA, PC7	V_{OL}	— —	— —	0.4 0.4	V
Input high voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
EPROM programming voltage	V_{PP}	14.5	14.75	15.0	V
EPROM/OTPROM programming current	I_{PP}	—	5	10	mA
User mode current	I_{PP}	—	—	± 10	mA
Data-retention mode (0°C to 70°C)	V_{RM}	2.0	—	—	V
Supply current [3] Run[4] Wait[5] Stop[6] 25°C –40°C to +85°C	I_{DD}	 — — — —	 5.0 1.95 5.0 5.0	 7.0 3.0 50 50	 mA mA µµA µµA
I/O ports hi-z leakage current PA7–PA0, PB7–PB0, PC7–PC0, PD4–PD1, PD7, $\overline{RESET}$	I_{IL}	—	—	± 10	µA
Input current, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5	I_{In}	—	—	± 1	µA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0–PD5, PD7	C_{Out} C_{In}	— —	— —	12 8	pF

1. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range at 25°C.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects run I_{DD}.
5. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects wait I_{DD}.
6. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.



(a) Wait Mode



(b) Run Mode

Figure 13-3. Typical Current versus Internal Frequency for Run and Wait Modes

13.9 5.0-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option	f_{OSC}	— dc	4.2 4.2	MHz
Internal operating frequency Crystal ($f_{OSC} \div 2$) External clock ($f_{OSC} \div 2$)	f_{OP}	— dc	2.1 2.1	MHz
Cycle time (see **Figure 13-7**)	t_{CYC}	480	—	ns
Crystal oscillator startup time (see **Figure 13-7**)	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator) (see **Figure 13-6**)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width (see **Figure 13-7**)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input capture pulse width (see **Figure 13-5**) Input capture pulse period (see **Figure 13-5**)	t_{RESL} t_{TH}, t_{TL} t_{TLTL}	4.0 125 [3]	— — —	t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered) (see **Figure 4-2. External Interrupt Timing**)	t_{ILIH}	125	—	ns
Interrupt pulse period (see **Figure 4-2. External Interrupt Timing**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period, t_{TLTL}, should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

V

W