AN520

A Comparison of 8-Bit Microcontrollers

Author: *Mark Palmer*
Logic Products Division

INTRODUCTION

The PIC16C5X/XX microcontrollers from Microchip Technology Inc., provides significant execution speed and code-compaction improvement over any other 8-bit microcontroller in its price range.

The superior performance of the PIC16C5X/XX microcontrollers can be attributed primarily to its RISC-like architecture. The PIC16C5X/XX devices employ a Harvard architecture (i.e., has separate program memory space and data memory space [8-bit wide data]). It also uses a two stage pipelining instruction fetch and execution. All instructions are executed in a single cycle (200 ns @ 20 MHz clock) except for program branches which take two cycles, and there are only 33 instructions to remember.

Separation of program and data space allows the instruction word to be optimized to any size (12-bit wide for PIC16C5X devices and 14-bit wide for PIC16CXX devices). This makes it possible, for example, to load an 8-bit immediate value in one cycle. First, because there is no conflict between instruction fetch and data fetch (as opposed to von Neumann architecture) and secondly because the instruction word is wide enough to hold the 8-bit data.

In the following sections we will compare the PIC16C5X/XX devices @ 20 MHz with:

- SGS-Thomson ST62 @ 8 MHz
- Motorola MC68HC05 @ 4.2 MHz
- Intel 8051 @ 20 MHz
- Zilog Z86CXX @ 12 MHz
- National COP800 @ 20 MHz

Several coding examples will be considered. While the comparisons are not entirely scientific, they will demonstrate to the reader the relative superior performance of the PIC16C5X/XX devices. The examples chosen here are used frequently in microcontroller applications.

PACKING BCD

This example will take two bytes in RAM or registers, each containing a BCD digit in the lower nibble and create a packed BCD data byte, which is stored back in the register or RAM location holding the low BCD digit.

PIC16C5X/XX

		Byte/Words	Cycles
SWAPF	REGHI,W	1	1
IORWF	REGLO	1	1
		2	2

0.4 µs

COP800

		Byte/Words	Cycles
X	A,[B+]	1	2
SWAP	A	1	1
OR	A,[B]	1	1
X	A,[B]	1	1
		4	5

5 µs

B is pointing to the higher BCD digit initially.
After auto-increment, it points to the lower BCD digit.

ST62

		Byte/Words	Cycles
LD	A,REGHI	2	4
RLC	A	1	4
RLC	A	1	4
RLC	A	1	4
RLC	A	1	4
ADD	A,REGLO	2	4
LD	REGLO,A	2	4
		10	28

45.5 µs

REGHI and REGLO are registers addressable by short direct addressing mode.

MC68HC05

		Byte/Words	Cycles
LDA	REGHI	2	3
ROLA		1	3
ROLA		1	3
ROLA		1	3
ROLA		1	3
ADD	REGLO	2	3
STA	REGLO	2	4
		10	22

10.5 µs

Z86CXX

		Byte/Words	Cycles
SWAP	REGHI	2	8
OR	REGHI,REGLO	2	6
		4	14

5.33 µs

REGHI and REGLO are addressable via the working register addressing mode.

8051

		Byte/Words	Cycles
MOV	A,Rx	1	1
SWAP	A	1	1
ORL	A,Ry	1	1
MOV	Ry,A	1	1
		4	4

2.4 µs

Register Rx contains higher BCD digit, Ry holds lower BCD digit.

LOOP CONTROL

This example is one of simple loop control where a register containing loop count is decremented, tested for zero and if not branched back to the beginning of the loop.

PIC16C5X/XX

		Byte/Words	Cycles
DECFSZ	COUNT	1	1/2
GOTO	BEG_LOOP	1	2/-
		2	3/2

0.6 µs/0.4 µs

COP800

		Byte/Words	Cycles
DRSZ	COUNT	1	3
JP	BEG_LOOP	1	3
		2	6

COUNT is Register (RAM F0h-FFh).

6 µs

ST62

		Byte/Words	Cycles
DEC	X	1	4
JRZ	BEG_LOOP	1	2
		2	6

9.75 µs

MC68HC05

		Byte/Words	Cycles
DECX		1	3
BEQ	BEG_LOOP	2	3
		3	6

2.86 µs

Z86CXX

		Byte/Words	Cycles
DJNZ	COUNT, BEG_LOOP	2	10/12

1.67 µs/2.0 µs

8051

		Byte/Words	Cycles
DJNZ	Rx, BEG_LOOP	2	2

1.2 µs

BIT TEST & BRANCH

This example tests a single bit in a register or a RAM location and makes a conditional branch. We assume that MSB is tested and a branch is to be taken if the bit is set.

PIC16C5X/XX

		Byte/Words	Cycles
BTFSC	REG, 7	1	1/2
GOTO	NEWADD	1	2/-
		2	3/2

0.6 µs/0.4 µs

COP800

		Byte/Words	Cycles
IFBIT	7, [B]	1	1
JP	NEWADD	1	3
		2	4

B points to the memory location under test.

4 µs

ST62

		Byte/Words	Cycles
JRR	7, NEWADD	3	5

8.125 µs

MC68HC05

		Byte/Words	Cycles
BRCLR	7, NEWADD	3	5

2.38 µs

Z86CXX

		Byte/Words	Cycles
BTJRT	NEWADD, REG, 7	3	16/18

2.67 µs/3.0 µs

8051

		Byte/Words	Cycles
MOV	A, @Rx	1	1
JB	A.7, NEWADD	3	2
		4	3

Registers Rx is assumed to be pointing to the memory location under test.

1.8 µs

SHIFTING OUT 8-BIT DATA & CLOCK

We will now consider the task of serially shifting out an 8-bit data and clock. Data and clock outputs are generated under program control by toggling two output pins.

Data is transmitted on the rising edge of the clock. No attempt is made to make the clock output symmetrical in order to make the code efficient. Data out is guaranteed on the falling edge of the clock. These conditions are satisfactory for most applications.

PIC16C5X/XX

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
XMIT	MOVLW	08H	; Bit Count	1	1	1	
	MOVWF	BITCNT	;	1	1	1	
XM1	BCF	PORTB, 0	; 0 → Data Out Pin	1	1	1	
	BCF	PORTB, 1	; 0 → Clock Out Pin	1	1	1	
	RRF	XDATA	; Rotate Right thru Carry	1	1	1	
	BTFSC	STATUS, CARRY	; Test Carry Bit	1	2	1	
	BSF	PORTB, 0	; 1 → Data Out Pin	1	—	1	
	BSF	PORTB, 1	; 1 → Clock Out Pin	1	1	1	Loop
	DECFSZ	BITCNT	; Decrement Count ; Skip if Zero	1	1	1	
	GOTO	XM1	;	1	2	2	
	BCF	PORTC, 1	; 0 → Clock	1	1	1	
				11	74	74	

Transmit time is the same for 00h or FFh: 74 Tcyc = 14.8 µs. Note that there was no need to load the data in the Accumulator (W) since the PIC16C5X/XX can operate directly on file registers.

COP800

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
XMIT	LD	A, XDATA	; Load Data in Acc.	2	3	3	
	LD	BITCNT #08H	; Load Bit Count	2	3	3	
	LD	B, #D0H	; B Points to PORTL	2	3	3	
XM1	RBIT	0,[B]	; 0 → Clock	1	1	1	
	RBIT	1,[B]	; 0 → Data	1	1	1	
	RRCA		; Rotate A Right thru Carry	1	1	1	
	IFC		;	1	1	1	
	SBIT	1,[B]	; 1 → Data	1	—	1	Loop
	SBIT	0,[B]	; 0 → Clock	1	1	1	
	DRSZ	BITCNT	; Decrement Bit Count	1	3	3	
	JP	XM1	; and Go Back if ≠ 0	2	3	3	
	RBIT	0,[B]	;	1	3	3	
				16	100	108	

Accumulator A is first loaded with the data word. Transmit time is maximum for data = FFh; 105 Tcyc = 105 µs.

ST62

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
	LDI	A, #08	; Bit Count	2	4	4	
	LD	X, A	; Xmit Data	1	4	4	
	LD	A, W	; 0 → Clock	1	4	4	
XM1	RES	0, DRB	; 0 → Data	2	4	4	
	RES	1, DRB	;	2	4	4	
	SLA	A	;	2	4	4	
	JRNC	XM2	; 1 → Data	1	2	2	
XM2	SET	1, DRB	; 1 → CLK	2	—	4	Loop
	SET	0, DRB	;	2	4	4	
	DEC	X	;	1	4	4	
	JRNZ	XM1	;	1	2	2	
	RES	0, DRB	; ; 0 → Data	2	4	4	
				19	208	240	

Register W contains the Data Word.
Transmit time for FFh = 240 cycles = 390 µs.

SHIFTING OUT 8-BIT DATA & CLOCK (CONT.)

MC68HC05

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
XMIT	LDA	XDATA	; Load Xmit Data	2	3	3	
	LDX	#$08	; Load Bit Count	2	2	2	
XM1	BCLR	0, PORTB	; 0 → Clock	2	5	5	
	BCLR	1, PORTB	; 0 → Data	2	5	5	
	ROLA		;	1	3	3	
	BCC	XM2	;	2	3	3	
	BSET	1, PORTB	; 1 → Data	2	—	5	Loop
	BSET	0, PORTB	; 1 → Clock	2	5	5	
XM2	DECX		;	1	3	3	
	BNE	XM1	;	2	3	3	
	BCLR	0, PORTB	; 0 → Data	2	5	5	
				20	226	266	

Transmit time is maximum for transmitting FFh = 266 cycles = 126.7 µs.

Z86CXX

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
XMIT	LD	COUNT, #8	; Load Bit Count	3	10	10	
	AND	P2, #%FC	; 0 → Data, Clock	3	6	6	
XM1	RRC	XDATA	;	2	6	6	
	JR	NC, XM2	;	2	12	10	
	OR	P2, #01	; 1 → Data	3	—	10	
XM2	OR	P2, #02	; 1 → Clock	3	10	10	Loop
	DJNZ	COUNT, XM1	;	2	12	12	
	AND	P2, #%FC	; 0 → Clock, Data	3	10	10	
				21	348	412	

Transmit time is maximum for transmitting FFh = 412 cycles = 68.67 µs.

8051

				Byte /Words	Cycles Xmit 00h	Cycles Xmit FFh	
MXIT	MOV	A, @R0	; R0 Points to Data Word	1	1	1	
	MOV	R1, #08H	; Load Bit Count	2	1	1	
XM1	ANL	PORT1, #0FCH	; 0 → Data, Clock	3	2	2	
	RRC	A	; Rotate Right A thru Carry	1	1	1	
	JNC	XM2	;	2	2	2	
	SETB	PORT1, 0	; 1 → Data	2	—	1	Loop
XM2	SETB	PORT1, 1	; 1 → Clock	2	1	1	
	DJNZ	R1, XM1	; Decrement Count	2	2	2	
				15	66	74	

Transmit time is maximum for transmitting FFh = 74 cycles = 44.4 µs.

SOFTWARE TIMER

Microcontrollers quite often need to implement time delays. Debouncing key input, pulse width modulation, and phase angle control are just a few examples. Implementing a 10 ms time delay loop subroutine will be considered in this section.

PIC16C5X/XX

```
                                                        Byte/Words   Cycles
DELAY   MOVLW   41H         ; 10 ms Delay Loop              1          1
        MOVWF   COUNT2      ;                               1          1
        CLRF    COUNT1      ;                               1          1

LOOP    INCFSZ  COUNT1      ; This inner Loop will be       1          2/1
        GOTO    LOOP        ; Executed 256 Times            1          2
        DECFSZ  COUNT2      ;                               1          2/1
        GOTO    LOOP        ;                               1          2
        RET                 ;                               1          2
                                                            8
```

Execution time for the routine = 5 + (255 x 3 + 5) • 65 = 20025 Tcyc = 10.011 ms. The PIC16C5X/XX can implement delay times very precisely (when necessary) because of its fine instruction cycle resolution.

COP800

```
                                                        Byte/Words   Cycles
DELAY   LD      COUNT1, #0BH ; 10 ms Delay Loop             2          3
        LD      B, #0EH      ;                              1          1

LOOP    DRSZ    B            ;                              1          1
        JP      LOOP         ;                              1          1
        DRSZ    COUNT1       ;                              1          1
        JP      LOOP         ;                              1          1
        RET                  ;                              1          5
                                                            8
```

Execution time for the routine = (6N2 + 6) N1 + 9 cycles. Here N1 = 0BH and N2 = 0EH, which gives us: 999 Tcyc = 9.99 ms.

ST62

```
                                                        Byte/Words   Cycles
        LDI     A, #FF                                      2          4
        LD      X, A        ; LOOP1 Count                   1          4
        LDI     A, #04                                      2          4
        LD      Y, A        ; LOOP2 Count                   1          4
LOOP    DEC     X           ; 0   CLK                       1          4
        JRNZ    LOOP        ;                               1          2
        DEC     Y           ; 0   CLK                       1          4
        JRNZ    LOOP                                        1          2
                                                           10
```

Execution time for the subroutine = (6N1 + 6) N2 + 16 cycles, where N1 = FFh, N2 = 04 gives us 10.01 ms.

SOFTWARE TIMER (CONT.)

MC68HC05

```
                                                          Byte/Words   Cycles
DELAY   LDX     $2D         ; 10 ms Delay Loop                2          2
        LDX     $5C         ;                                 2          2

LOOP    DECA                ;                                 1          3
        BNE                 ;                                 2          2
        DECX    LOOP        ;                                 1          3
        BNE                 ;                                 2          2
        RTS     LOOP        ;                                 1          6
                                                              11
```

Execution time for the subroutine = (5 x N1 + 5) N2 + 10, with N1 = 2DH, N2 = 5CH, time delay = 10.081 ms.

Z86CXX

```
                                                          Byte/Words   Cycles
DELAY   LD      COUNT1, #%61    ; 10 ms Delay Loop            2          6
        LD      COUNT2, #%33    ;                             2          6

LOOP    DJNZ    COUNT1, LOOP    ;                             2         10/12
        DJNZ    COUNT2, LOOP    ;                             2         10/12
        RET                     ;                             1         14
                                                              9
```

Total execution time = (12N1 +10)N2, with N1 = 61H, N2 = 33H, time delay = 59976 cycles = 9.979 ms.

8051

```
                                                          Byte/Words   Cycles
DELAY   MOV     COUNT1, #21H    ; 10 ms Delay Loop            2          1

LOOP1   MOV     COUNT2, #FBH    ;                             2          1

LOOP2   DJNZ    COUNT2, LOOP2   ;                             3          2
        DJNZ    COUNT1, LOOP1   ;                             3          2
        RET                     ;                             1          2
                                                              11
```

Execution time for the subroutine = (2N1 + 3)N2 + 3 cycles. Where N1 = FBH, N2 = 21H, which gives us: 16668 cycles = 10.0008 ms.

SUMMARY

Table 1 summarizes code sizes for different microcontrollers. The overall relative code size number is an average of the individual relative code sizes. Given that the program word size of the PIC16C5X/XX is 12- or 14-bit (compared to an 8-bit program memory of all the other microcontrollers), a compaction of 1.5 is expected. Clearly, the PIC16C5X/XX meets this compaction (except for the COP800) and exceeds the compaction ratio in most comparisons.

Table 2 summarizes relative execution speed. The overall speed is an average of relative speed numbers. For example, the COP800 will, on average, exhibit 27% of the code execution speed of a PIC16C5X/XX devices. In other words, the PIC16C5X/XX will be (1/0.27) 3.7 times faster than a COP800 on average.

TABLE 1: COMPARISON OF CODE EFFICIENCY*

Device	Packing BCD	Loop Control	Bit Test & Branch	8-Bit Sync Transmission	10 ms Software Timer	Overall
COP800	4 2.00	2 1.00	2 1.00	16 1.46	8 1.00	1.29
ST62	10 5.00	2 1.00	3 1.50	19 1.73	10 1.25	2.10
MC68HC05	10 5.00	3 1.50	3 1.50	20 1.82	11 1.38	2.24
Z86CXX	4 2.00	2 1.00	3 1.50	21 1.91	9 1.125	1.51
8051	4 2.00	2 1.00	4 2.00	15 1.36	11 1.375	1.547
PIC16C5X/XX	2	2	2	11	8	1.00

* In each box, the top number is the number of program memory location required to code the application. The bottom number is relative code size compared to the PIC16C5X/XX:

$$\frac{\text{\# program memory locations for other microcontroller}}{\text{\# program memory locations for the PIC16C5X/XX}}$$

FIGURE 1: CODE SIZE COMPARISON

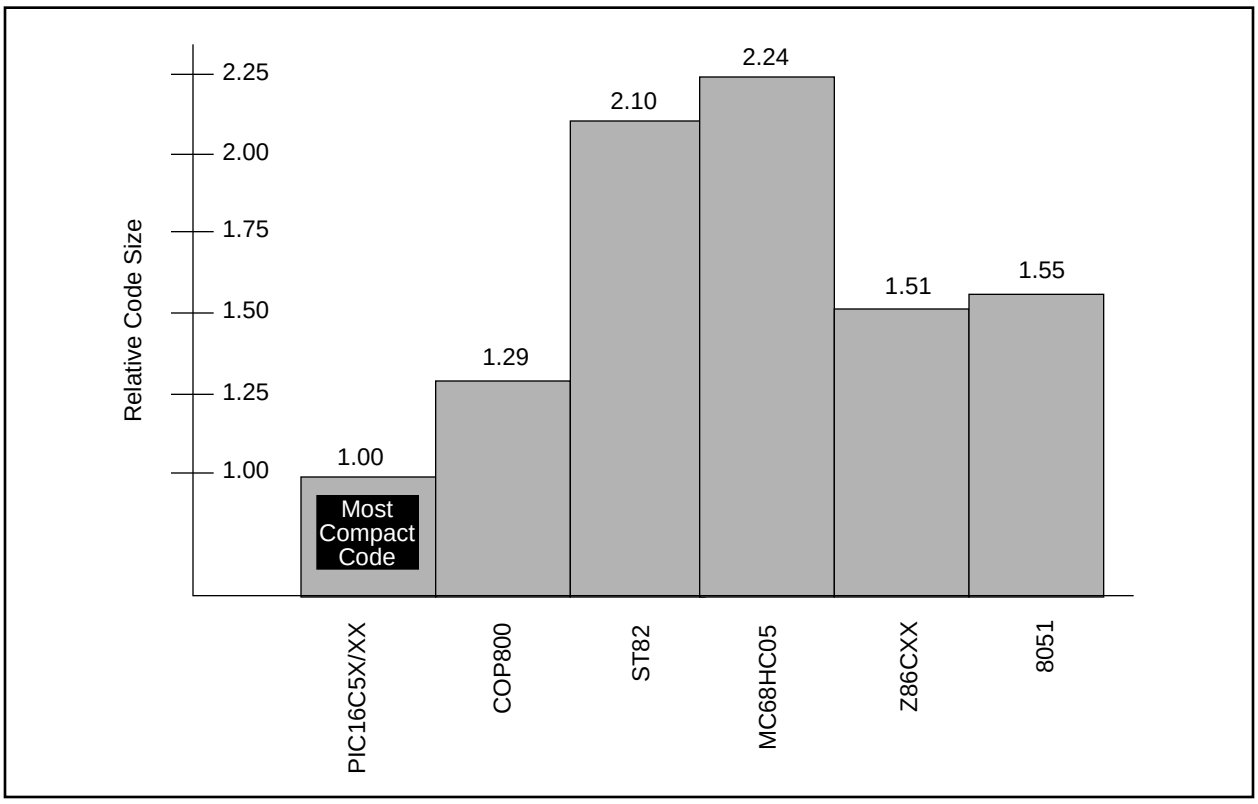


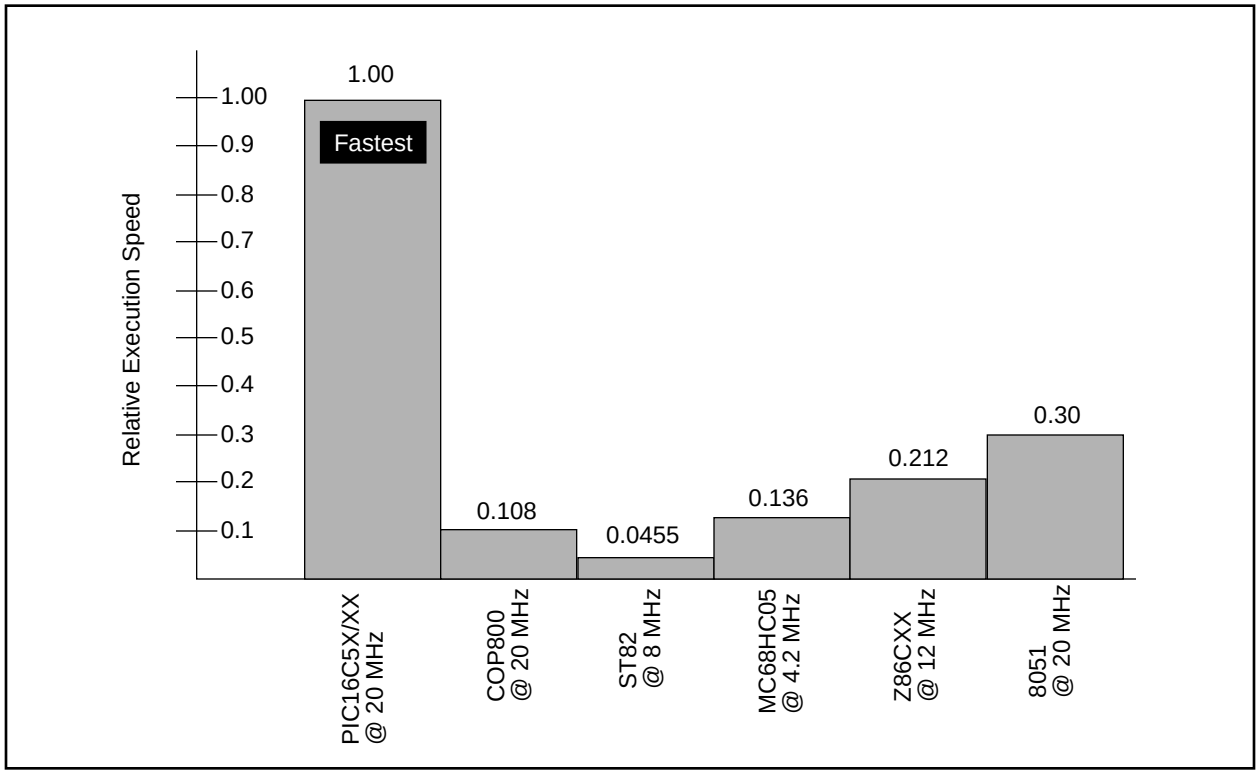
TABLE 2: COMPARISON OF EXECUTION SPEED

Device	Packing BCD	Loop Control	Bit Test & Branch	8-Bit Sync Transmission	10 ms Software Timer	Overall
COP800 @ 20 MHz	5 µs 0.08	6 µs 0.0832	4 µs 0.1252	105 µs 0.1408	—	0.108
ST62 @ 8 MHz	45.5 µs 0.0088	9.75 µs 0.0615	8.125 µs 0.0738	390 µs 0.0329	—	0.0455
MC68HC05 @ 4.2 MHz	10.05 µs 0.038	2.86 µs 0.1748	2.38 µs 0.21	126.7 µs 0.1168	—	0.136
Z86CXX @ 12 MHz	2.33 µs 0.172	1.835 µs 0.272	2.835µs 0.176	68.67 µs 0.224	—	0.212
8051 @ 20 MHz	2.4 µs 0.1666	1.2 µs 0.4166	1.8 µs 0.277	44.4 µs 0.33	—	0.30
PIC16C5X/XX @ 20 MHz	0.4 µs	0.6/0.4 µs	0.6/0.4 µs	14.8 µs	—	1.00

* In each box, the top number is the time required to execute the example code, while the bottom number is a measure of relative performance compared to the PIC16C5X/XX.

$$\frac{\text{time required to execute code by the PIC16C5X/XX}}{\text{time required to execute code by other microcontroller}}$$

FIGURE 2: EXECUTION SPEED COMPARISON



NOTES:

NOTES:

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microchip

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95

MICROCHIP

Printed in the USA, 9/95