

A Comparison of 8-Bit Microcontrollers

Author: Mark Palmer
Microchip Technology Inc.

INTRODUCTION

The PIC16C5X/XX microcontrollers from Microchip Technology Inc., provide 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 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 are used frequently in microcontroller applications.

PACKING BINARY CODED DECIMAL (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				COP800			
		Byte/Words	Cycles			Byte/Words	Cycles
SWAPF	REGHI,W	1	1	X	A,[B+]	1	2
IORWF	REGLO	$\frac{1}{2}$	$\frac{1}{2}$	SWAP	A	1	1
				OR	A,[B]	1	1
				X	A,[B]	$\frac{1}{4}$	$\frac{1}{5}$
			0.4 μ s				5 μ s
				B is pointing to the higher BCD digit initially. After auto-increment, it points to the lower BCD digit.			
ST62				MC68HC05			
		Byte/Words	Cycles			Byte/Words	Cycles
LD	A,REGHI	2	4	LDA	REGHI	2	3
RLC	A	1	4	ROLA		1	3
RLC	A	1	4	ROLA		1	3
RLC	A	1	4	ROLA		1	3
RLC	A	1	4	ROLA		1	3
ADD	A,REGLO	2	4	ADD	REGLO	2	3
LD	REGLO,A	2	4	STA	REGLO	2	4
		$\frac{2}{10}$	$\frac{4}{28}$			$\frac{2}{10}$	$\frac{4}{22}$
			45.5 μ s				10.5 μ s
				REGHI and REGLO are registers addressable by short direct addressing mode.			
Z86CXX				8051			
		Byte/Words	Cycles			Byte/Words	Cycles
SWAP	REGHI	2	8	MOV	A,Rx	1	1
OR	REGHI,REGLO	2	6	SWAP	A	1	1
		$\frac{2}{4}$	$\frac{6}{14}$	ORL	A,Ry	1	1
			5.33 μ s	MOV	Ry,A	$\frac{1}{4}$	$\frac{1}{4}$
							2.4 μ s
				Register Rx contains higher BCD digit, Ry holds lower BCD digit.			

AN520

LOOP CONTROL

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

<p>PIC16C5X/XX</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DECFSZ</td> <td>COUNT</td> <td>1</td> <td>1/2</td> </tr> <tr> <td>GOTO</td> <td>BEG_LOOP</td> <td>$\frac{1}{2}$</td> <td>$\frac{2/-}{3/2}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">0.6 μs/0.4 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DECFSZ	COUNT	1	1/2	GOTO	BEG_LOOP	$\frac{1}{2}$	$\frac{2/-}{3/2}$	0.6 μ s/0.4 μ s				<p>COP800</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DRSZ</td> <td>COUNT</td> <td>1</td> <td>3</td> </tr> <tr> <td>JP</td> <td>BEG_LOOP</td> <td>$\frac{1}{2}$</td> <td>$\frac{3}{6}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">COUNT is Register (RAM F0h-FFh). 6 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DRSZ	COUNT	1	3	JP	BEG_LOOP	$\frac{1}{2}$	$\frac{3}{6}$	COUNT is Register (RAM F0h-FFh). 6 μ s			
		Byte/Words	Cycles																														
DECFSZ	COUNT	1	1/2																														
GOTO	BEG_LOOP	$\frac{1}{2}$	$\frac{2/-}{3/2}$																														
0.6 μ s/0.4 μ s																																	
		Byte/Words	Cycles																														
DRSZ	COUNT	1	3																														
JP	BEG_LOOP	$\frac{1}{2}$	$\frac{3}{6}$																														
COUNT is Register (RAM F0h-FFh). 6 μ s																																	
<p>ST62</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DEC</td> <td>X</td> <td>1</td> <td>4</td> </tr> <tr> <td>JRZ</td> <td>BEG_LOOP</td> <td>$\frac{1}{2}$</td> <td>$\frac{2}{6}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">9.75 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DEC	X	1	4	JRZ	BEG_LOOP	$\frac{1}{2}$	$\frac{2}{6}$	9.75 μ s				<p>MC68HC05</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DECX</td> <td></td> <td>1</td> <td>3</td> </tr> <tr> <td>BEQ</td> <td>BEG_LOOP</td> <td>$\frac{2}{3}$</td> <td>$\frac{3}{6}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">2.86 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DECX		1	3	BEQ	BEG_LOOP	$\frac{2}{3}$	$\frac{3}{6}$	2.86 μ s			
		Byte/Words	Cycles																														
DEC	X	1	4																														
JRZ	BEG_LOOP	$\frac{1}{2}$	$\frac{2}{6}$																														
9.75 μ s																																	
		Byte/Words	Cycles																														
DECX		1	3																														
BEQ	BEG_LOOP	$\frac{2}{3}$	$\frac{3}{6}$																														
2.86 μ s																																	
<p>Z86CXX</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DJNZ</td> <td>COUNT, BEG_LOOP</td> <td>2</td> <td>10/12</td> </tr> <tr> <td colspan="4" style="text-align: right;">1.67 μs/2.0 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DJNZ	COUNT, BEG_LOOP	2	10/12	1.67 μ s/2.0 μ s				<p>8051</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>DJNZ</td> <td>Rx, BEG_LOOP</td> <td>2</td> <td>2</td> </tr> <tr> <td colspan="4" style="text-align: right;">1.2 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	DJNZ	Rx, BEG_LOOP	2	2	1.2 μ s											
		Byte/Words	Cycles																														
DJNZ	COUNT, BEG_LOOP	2	10/12																														
1.67 μ s/2.0 μ s																																	
		Byte/Words	Cycles																														
DJNZ	Rx, BEG_LOOP	2	2																														
1.2 μ s																																	

BIT TEST & BRANCH

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

<p>PIC16C5X/XX</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>BTFSC</td> <td>REG, 7</td> <td>1</td> <td>1/2</td> </tr> <tr> <td>GOTO</td> <td>NEWADD</td> <td>$\frac{1}{2}$</td> <td>$\frac{2/-}{3/2}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">0.6 μs/0.4 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	BTFSC	REG, 7	1	1/2	GOTO	NEWADD	$\frac{1}{2}$	$\frac{2/-}{3/2}$	0.6 μ s/0.4 μ s				<p>COP800</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>IFBIT</td> <td>7, [B]</td> <td>1</td> <td>1</td> </tr> <tr> <td>JP</td> <td>NEWADD</td> <td>$\frac{1}{2}$</td> <td>$\frac{3}{4}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">B points to the memory location under test. 4 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	IFBIT	7, [B]	1	1	JP	NEWADD	$\frac{1}{2}$	$\frac{3}{4}$	B points to the memory location under test. 4 μ s			
		Byte/Words	Cycles																														
BTFSC	REG, 7	1	1/2																														
GOTO	NEWADD	$\frac{1}{2}$	$\frac{2/-}{3/2}$																														
0.6 μ s/0.4 μ s																																	
		Byte/Words	Cycles																														
IFBIT	7, [B]	1	1																														
JP	NEWADD	$\frac{1}{2}$	$\frac{3}{4}$																														
B points to the memory location under test. 4 μ s																																	
<p>ST62</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>JRR</td> <td>7, NEWADD</td> <td>3</td> <td>5</td> </tr> <tr> <td colspan="4" style="text-align: right;">8.125 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	JRR	7, NEWADD	3	5	8.125 μ s				<p>MC68HC05</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>BRCLR</td> <td>7, NEWADD</td> <td>3</td> <td>5</td> </tr> <tr> <td colspan="4" style="text-align: right;">2.38 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	BRCLR	7, NEWADD	3	5	2.38 μ s											
		Byte/Words	Cycles																														
JRR	7, NEWADD	3	5																														
8.125 μ s																																	
		Byte/Words	Cycles																														
BRCLR	7, NEWADD	3	5																														
2.38 μ s																																	
<p>Z86CXX</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>BTJRT</td> <td>NEWADD, REG, 7</td> <td>3</td> <td>16/18</td> </tr> <tr> <td colspan="4" style="text-align: right;">2.67 μs/3.0 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	BTJRT	NEWADD, REG, 7	3	16/18	2.67 μ s/3.0 μ s				<p>8051</p> <table> <thead> <tr> <th></th> <th></th> <th>Byte/Words</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>MOV</td> <td>A, @Rx</td> <td>1</td> <td>1</td> </tr> <tr> <td>JB</td> <td>A.7, NEWADD</td> <td>$\frac{3}{4}$</td> <td>$\frac{2}{3}$</td> </tr> <tr> <td colspan="4" style="text-align: right;">Register Rx is assumed to be pointing to the memory location under test. 1.8 μs</td> </tr> </tbody> </table>			Byte/Words	Cycles	MOV	A, @Rx	1	1	JB	A.7, NEWADD	$\frac{3}{4}$	$\frac{2}{3}$	Register Rx is assumed to be pointing to the memory location under test. 1.8 μ s							
		Byte/Words	Cycles																														
BTJRT	NEWADD, REG, 7	3	16/18																														
2.67 μ s/3.0 μ s																																	
		Byte/Words	Cycles																														
MOV	A, @Rx	1	1																														
JB	A.7, NEWADD	$\frac{3}{4}$	$\frac{2}{3}$																														
Register Rx is assumed to be pointing to the memory location under test. 1.8 μ s																																	

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
	DECFSZ	BITCNT	; Decrement Count ; Skip if Zero	1	1	1
	GOTO	XM1	;	1	2	2
	BCF	PORTC, 1	; 0 → Clock	1	1	1
				<u>11</u>	<u>74</u>	<u>74</u>

Transmit time is the same for 00h or FFh: 74 Tcyc = 14.8 μs. Note that there was no need to load the data into 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
	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
				<u>16</u>	<u>100</u>	<u>108</u>

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
	SET	0, DRB	;	2	4	4
	DEC	X	;	1	4	4
	JRNZ	XM1	;	1	2	2
	RES	0, DRB	;	2	4	4
			; 0 → Data	<u>19</u>	<u>208</u>	<u>240</u>

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

AN520

SHIFTING OUT 8-BIT DATA & CLOCK (Cont.'d)

				Byte	Cycles	Cycles
				/Words	Xmit 00h	Xmit FFh
MC68HC05						
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
XM2	BSET	0, PORTB	; 1 → Clock	2	5	5
	DECX	:	:	1	3	3
	BNE	XM1	:	2	3	3
	BCLR	0, PORTB	; 0 → Data	2	5	5
				<u>20</u>	<u>226</u>	<u>266</u>
Transmit time is maximum for transmitting FFh = 266 cycles = 126.7 μs.						
Z86CXX						
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
	DJNZ	COUNT, XM1	:	2	12	12
	AND	P2, #%FC	; 0 → Clock, Data	3	10	10
				<u>21</u>	<u>348</u>	<u>412</u>
Transmit time is maximum for transmitting FFh = 412 cycles = 68.67 μs.						
8051						
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
XM2	SETB	PORT1, 1	; 1 → Clock	2	1	1
	DJNZ	R1, XM1	; Decrement Count	2	2	2
				<u>15</u>	<u>66</u>	<u>74</u>
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
				<u>8</u>	
Execution time for the routine = $5 + (255 \times 3 + 5) \times 65 = 20025 \text{ Tcyc} = 10.011 \text{ ms}$. 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
				<u>8</u>	
Execution time for the routine = $(6N2 + 6) N1 + 9$ cycles. Here $N1 = 0Bh$ and $N2 = 0Eh$, which gives us: $999 \text{ Tcyc} = 9.99 \text{ ms}$.					
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
				<u>10</u>	
Execution time for the subroutine = $(6N1 + 6) N2 + 16$ cycles, where $N1 = FFh$, $N2 = 04h$ gives us 10.01 ms.					

AN520

SOFTWARE TIMER (Cont.'d)

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
				<u>11</u>	
Execution time for the subroutine = $(5 \times N1 + 5)N2 + 10$, 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		;	<u>1</u>	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		;	<u>1</u>	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, a 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 locations 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

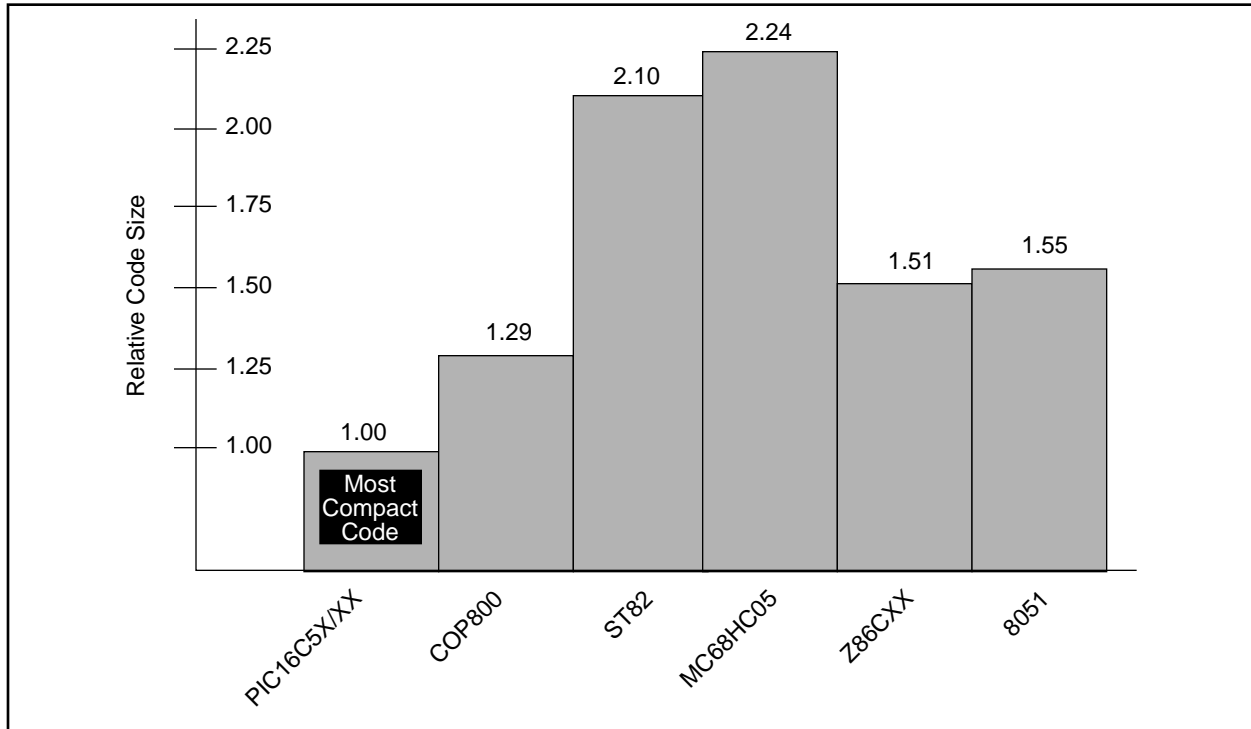


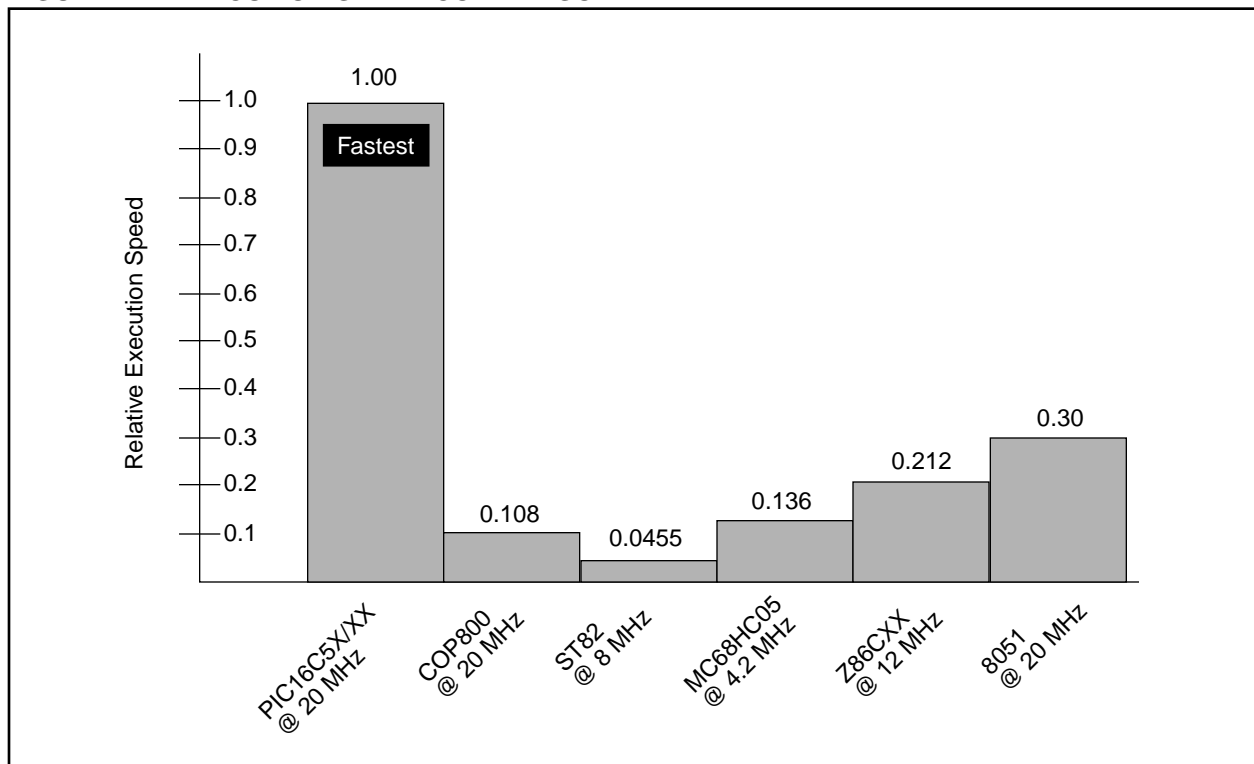
TABLE 2: COMPARISON OF EXECUTION SPEED

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



Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks


The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

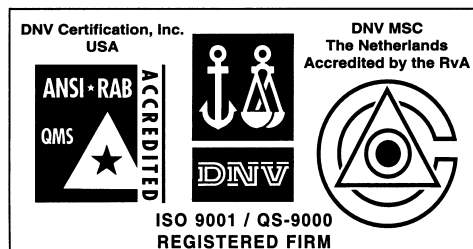
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: <http://www.microchip.com>

Rocky Mountain

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta

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

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York

150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose

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

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia

Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing

Microchip Technology Consulting (Shanghai)
Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu

Microchip Technology Consulting (Shanghai)
Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou

Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai

Microchip Technology Consulting (Shanghai)
Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen

Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong

Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India

Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan

Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471-6166 Fax: 81-45-471-6122

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore

Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan

Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark

Microchip Technology Nordic ApS
Regus Business Centre
Lautrup høj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France

Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - 1er Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

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

Italy

Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom

Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

03/01/02