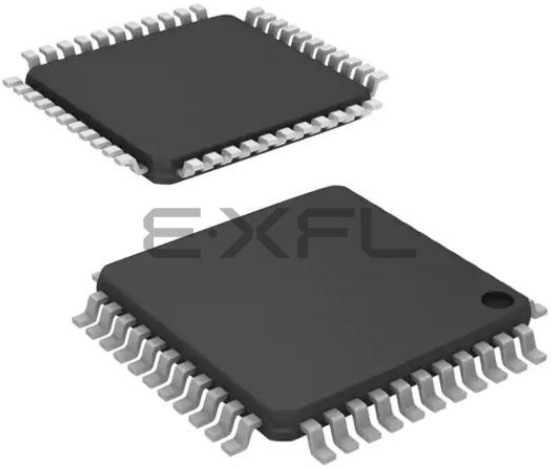


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	33MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	4KB (2K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	232 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c42a-33e-pt

NOTES:

6.2.2.3 TMR0 STATUS/CONTROL REGISTER (T0STA)

This register contains various control bits. Bit7 (INTEDG) is used to control the edge upon which a signal on the RA0/INT pin will set the RB0/INT interrupt flag. The other bits configure the Timer0 prescaler and clock source. (Figure 11-1).

FIGURE 6-9: T0STA REGISTER (ADDRESS: 05h, UNBANKED)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0
INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented, reads as ‘0’
-n = Value at POR reset

bit 7: **INTEDG**: RA0/INT Pin Interrupt Edge Select bit
This bit selects the edge upon which the interrupt is detected.
1 = Rising edge of RA0/INT pin generates interrupt
0 = Falling edge of RA0/INT pin generates interrupt

bit 6: **T0SE**: Timer0 Clock Input Edge Select bit
This bit selects the edge upon which TMR0 will increment.
<u>When T0CS = 0</u>
1 = Rising edge of RA1/T0CKI pin increments TMR0 and/or generates a T0CKIF interrupt
0 = Falling edge of RA1/T0CKI pin increments TMR0 and/or generates a T0CKIF interrupt
<u>When T0CS = 1</u>
Don’t care

bit 5: **T0CS**: Timer0 Clock Source Select bit
This bit selects the clock source for Timer0.
1 = Internal instruction clock cycle (TCY)
0 = T0CKI pin

bit 4-1: **PS3:PS0**: Timer0 Prescale Selection bits
These bits select the prescale value for Timer0.

PS3:PS0	Prescale Value
0000	1:1
0001	1:2
0010	1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

bit 0: **Unimplemented**: Read as '0'

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in 4 registers RES3:RES0.

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
\text{RES3:RES0} &= \text{ARG1H:ARG1L} * \text{ARG2H:ARG2L} \\
&= (\text{ARG1H} * \text{ARG2H} * 2^{16}) + \\
&\quad (\text{ARG1H} * \text{ARG2L} * 2^{8}) + \\
&\quad (\text{ARG1L} * \text{ARG2H} * 2^{8}) + \\
&\quad (\text{ARG1L} * \text{ARG2L})
\end{aligned}
$$

EXAMPLE 8-3: 16 x 16 MULTIPLY ROUTINE

```
   MOVFP    ARG1L, WREG
   MULWF    ARG2L       ; ARG1L * ARG2L ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES1 ;
   MOVPF    PRODL, RES0 ;
;
   MOVFP    ARG1H, WREG
   MULWF    ARG2H       ; ARG1H * ARG2H ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES3 ;
   MOVPF    PRODL, RES2 ;
;
   MOVFP    ARG1L, WREG
   MULWF    ARG2H       ; ARG1L * ARG2H ->
                        ;   PRODH:PRODL
   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
;
   MOVFP    ARG1H, WREG ;
   MULWF    ARG2L       ; ARG1H * ARG2L ->
                        ;   PRODH:PRODL

   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
```

10.0 OVERVIEW OF TIMER RESOURCES

The PIC17C4X has four timer modules. Each module can generate an interrupt to indicate that an event has occurred. These timers are called:

- Timer0 - 16-bit timer with programmable 8-bit prescaler
- Timer1 - 8-bit timer
- Timer2 - 8-bit timer
- Timer3 - 16-bit timer

For enhanced time-base functionality, two input Captures and two Pulse Width Modulation (PWM) outputs are possible. The PWMs use the TMR1 and TMR2 resources and the input Captures use the TMR3 resource.

10.1 Timer0 Overview

The Timer0 module is a simple 16-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock.

The Timer0 module also has a programmable prescaler option. The PS3:PS0 bits (T0STA<4:1>) determine the prescaler value. TMR0 can increment at the following rates: 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

When TImer0's clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

10.2 Timer1 Overview

The TImer0 module is an 8-bit timer/counter with an 8-bit period register (PR1). When the TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the Timer2 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR1 register is the LSB and TMR2 is the MSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

10.3 Timer2 Overview

The TMR2 module is an 8-bit timer/counter with an 8-bit period register (PR2). When the TMR2 value rolls over from the period match value to 0h, the TMR2IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the TMR1 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR2 register is the MSB and TMR1 is the LSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

10.4 Timer3 Overview

The TImer3 module is a 16-bit timer/counter with a 16-bit period register. When the TMR3H:TMR3L value rolls over to 0h, the TMR3IF bit is set and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB5/TCLK3 pin.

When operating in the dual capture mode, the period registers become the second 16-bit capture register.

10.5 Role of the Timer/Counters

The timer modules are general purpose, but have dedicated resources associated with them. TImer1 and Timer2 are the time-bases for the two Pulse Width Modulation (PWM) outputs, while Timer3 is the time-base for the two input captures.

12.1.2 TIMER1 & TIMER2 IN 16-BIT MODE

To select 16-bit mode, the T16 bit must be set. In this mode TMR1 and TMR2 are concatenated to form a 16-bit timer (TMR2:TMR1). The 16-bit timer increments until it matches the 16-bit period register (PR2:PR1). On the following timer clock, the timer value is reset to 0h, and the TMR1IF bit is set.

When selecting the clock source for the16-bit timer, the TMR1CS bit controls the entire 16-bit timer and TMR2CS is a “don’t care.” When TMR1CS is clear, the timer increments once every instruction cycle (Fosc/4). When TMR1CS is set, the timer increments on every falling edge of the RB4/TCLK12 pin. For the 16-bit timer to increment, both TMR1ON and TMR2ON bits must be set (Table 12-1).

12.1.2.1 EXTERNAL CLOCK INPUT FOR TMR1:TMR2

When TMR1CS is set, the 16-bit TMR2:TMR1 increments on the falling edge of clock input TCLK12. The input on the RB4/TCLK12 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on RB4/TCLK12 to the time TMR2:TMR1 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

TABLE 12-1: TURNING ON 16-BIT TIMER

TMR2ON	TMR1ON	Result
1	1	16-bit timer (TMR2:TMR1) ON
0	1	Only TMR1 increments
x	0	16-bit timer OFF

FIGURE 12-4: TMR1 AND TMR2 IN 16-BIT TIMER/COUNTER MODE

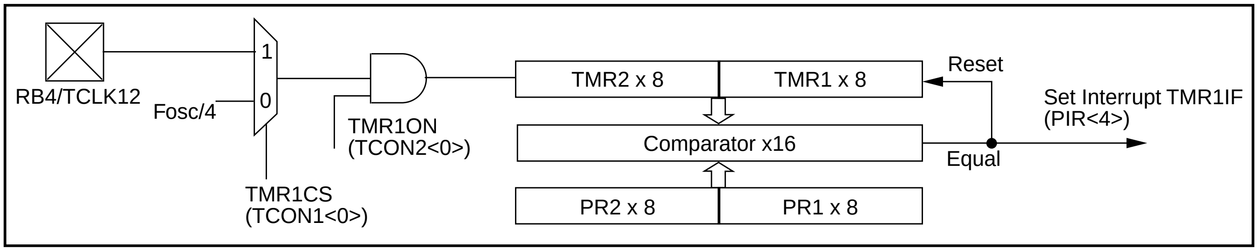


TABLE 12-2: SUMMARY OF TIMER1 AND TIMER2 REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 register								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 register								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	—	—	--11 11--	--11 qq--
14h, Bank 2	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', q - value depends on condition, shaded cells are not used by Timer1 or Timer2.

Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and WDT Timer Reset.

TABLE 13-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	`0000 0010`	`0000 0010`
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	`0000 -00x`	`0000 -00u`
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	`xxxx xxxx`	`uuuu uuuu`
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	`0000 0000`	`0000 0000`
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	`0000 --1x`	`0000 --1u`
17h, Bank 0	SPBRG	Baud rate generator register								`xxxx xxxx`	`uuuu uuuu`

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous master reception.

Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

TABLE 15-2: PIC17CXX INSTRUCTION SET (Cont.'d)

Mnemonic, Operands		Description	Cycles	16-bit Opcode		Status Affected	Notes
				MSb	LSb		
TABLWT	**t,i,f**	Table Write	2	`1010 11ti ffff ffff`		None	5
TLRD	**t,f**	Table Latch Read	1	`1010 00tx ffff ffff`		None	
TLWT	**t,f**	Table Latch Write	1	`1010 01tx ffff ffff`		None	
TSTFSZ	**f**	Test f, skip if 0	1 (2)	`0011 0011 ffff ffff`		None	6,8
XORWF	**f,d**	Exclusive OR WREG with f	1	`0000 110d ffff ffff`		Z	
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	**f,b**	Bit Clear f	1	`1000 1bbb ffff ffff`		None	
BSF	**f,b**	Bit Set f	1	`1000 0bbb ffff ffff`		None	
BTFSC	**f,b**	Bit test, skip if clear	1 (2)	`1001 1bbb ffff ffff`		None	6,8
BTFSS	**f,b**	Bit test, skip if set	1 (2)	`1001 0bbb ffff ffff`		None	6,8
BTG	**f,b**	Bit Toggle f	1	`0011 1bbb ffff ffff`		None	
LITERAL AND CONTROL OPERATIONS							
ADDLW	**k**	ADD literal to WREG	1	`1011 0001 kkkk kkkk`		OV,C,DC,Z	
ANDLW	**k**	AND literal with WREG	1	`1011 0101 kkkk kkkk`		Z	
CALL	**k**	Subroutine Call	2	`111k kkkk kkkk kkkk`		None	7
CLRWDT	**—**	Clear Watchdog Timer	1	`0000 0000 0000 0100`		$\overline{\text{TO}}$,$\overline{\text{PD}}$	
GOTO	**k**	Unconditional Branch	2	`110k kkkk kkkk kkkk`		None	7
IORLW	**k**	Inclusive OR literal with WREG	1	`1011 0011 kkkk kkkk`		Z	
LCALL	**k**	Long Call	2	`1011 0111 kkkk kkkk`		None	4,7
MOVLB	**k**	Move literal to low nibble in BSR	1	`1011 1000 uuuu kkkk`		None	
MOVLR	**k**	Move literal to high nibble in BSR	1	`1011 101x kkkk uuuu`		None	9
MOVLW	**k**	Move literal to WREG	1	`1011 0000 kkkk kkkk`		None	
MULLW	**k**	Multiply literal with WREG	1	`1011 1100 kkkk kkkk`		None	9
RETFIE	**—**	Return from interrupt (and enable interrupts)	2	`0000 0000 0000 0101`		GLINTD	7
RETLW	**k**	Return literal to WREG	2	`1011 0110 kkkk kkkk`		None	7
RETURN	**—**	Return from subroutine	2	`0000 0000 0000 0010`		None	7
SLEEP	**—**	Enter SLEEP Mode	1	`0000 0000 0000 0011`		$\overline{\text{TO}}$, $\overline{\text{PD}}$	
SUBLW	**k**	Subtract WREG from literal	1	`1011 0010 kkkk kkkk`		OV,C,DC,Z	
XORLW	**k**	Exclusive OR literal with WREG	1	`1011 0100 kkkk kkkk`		Z	

Legend: Refer to Table 15-1 for opcode field descriptions.

Note 1: 2's Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an `LCALL`, the contents of PCLATH are loaded into the MSB of the PC and `kkkk kkkk` is loaded into the LSB of the PC (PCL)

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.

6: Two-cycle instruction when condition is true, else single cycle instruction.

7: Two-cycle instruction except for `TABLRD` to PCL (program counter low byte) in which case it takes 3 cycles.

8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.

9: These instructions are not available on the PIC17C42.

DECF — Decrement f

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]

Operation: (f) – 1 → (dest)

Status Affected: OV, C, DC, Z

Encoding:

`0000`	`011d`	`ffff`	`ffff`

Description: Decrement register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write to destination

<u>Example</u>: `DECF    CNT,   1`

Before Instruction

CNT	=	0x01
Z	=	0

After Instruction

CNT	=	0x00
Z	=	1

DECFSZ — Decrement f, skip if 0

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]

Operation: (f) – 1 → (dest);
skip if result = 0

Status Affected: None

Encoding:

`0001`	`011d`	`ffff`	`ffff`

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write to destination

<u>Example</u>:

```
HERE      DECFSZ   CNT, 1
          GOTO     LOOP
CONTINUE
```

Before Instruction

PC	=	Address (`HERE`)

After Instruction

CNT	=	CNT - 1
If CNT	=	0;
PC	=	Address (`CONTINUE`)
If CNT	≠	0;
PC	=	Address (`HERE+1`)

SUBWF — Subtract WREG from f

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – (W) → (dest)

Status Affected: OV, C, DC, Z

Encoding:

`0000`	`010d`	`ffff`	`ffff`

Description: Subtract WREG from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write to destination

Example 1: `SUBWF REG1, 1`

Before Instruction

REG1	=	3	
WREG	=	2	
C	=	?	

After Instruction

REG1	=	1	
WREG	=	2	
C	=	1	; result is positive
Z	=	0	

Example 2:

Before Instruction

REG1	=	2	
WREG	=	2	
C	=	?	

After Instruction

REG1	=	0	
WREG	=	2	
C	=	1	; result is zero
Z	=	1	

Example 3:

Before Instruction

REG1	=	1	
WREG	=	2	
C	=	?	

After Instruction

REG1	=	FF	
WREG	=	2	
C	=	0	; result is negative
Z	=	0	

SUBWFB — Subtract WREG from f with Borrow

Syntax: [*label*] SUBWFB f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – (W) – $\overline{C}$ → (dest)

Status Affected: OV, C, DC, Z

Encoding:

`0000`	`001d`	`ffff`	`ffff`

Description: Subtract WREG and the carry flag (borrow) from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write to destination

Example 1: `SUBWFB REG1, 1`

Before Instruction

REG1	=	0x19	(`0001 1001`)
WREG	=	0x0D	(`0000 1101`)
C	=	1	

After Instruction

REG1	=	0x0C	(`0000 1011`)
WREG	=	0x0D	(`0000 1101`)
C	=	1	; result is positive
Z	=	0	

Example2: `SUBWFB REG1,0`

Before Instruction

REG1	=	0x1B	(`0001 1011`)
WREG	=	0x1A	(`0001 1010`)
C	=	0	

After Instruction

REG1	=	0x1B	(`0001 1011`)
WREG	=	0x00	
C	=	1	; result is zero
Z	=	1	

Example3: `SUBWFB REG1,1`

Before Instruction

REG1	=	0x03	(`0000 0011`)
WREG	=	0x0E	(`0000 1101`)
C	=	1	

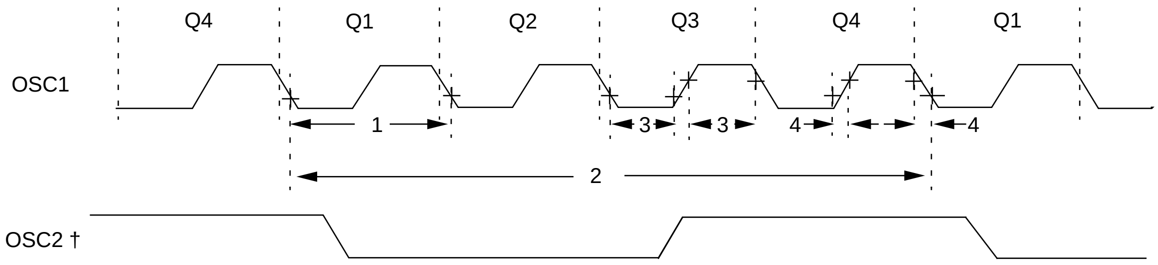
After Instruction

REG1	=	0xF5	(`1111 0100`) [2's comp]
WREG	=	0x0E	(`0000 1101`)
C	=	0	; result is negative
Z	=	0	

Applicable Devices	42	R42	42A	43	R43	44

17.4 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING



† In EC and RC modes only.

TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency (Note 1)**	DC DC	— —	16 25	MHz MHz	EC osc mode - PIC17C42-16 - PIC17C42-25
		Oscillator Frequency (Note 1)	DC 1 1 DC	— — — —	4 16 25 2	MHz MHz MHz MHz	RC osc mode XT osc mode - PIC17C42-16 - PIC17C42-25 LF osc mode
1	Tosc	**External CLKIN Period (Note 1)**	62.5 40	— —	— —	ns ns	EC osc mode - PIC17C42-16 - PIC17C42-25
		Oscillator Period (Note 1)	250 62.5 40 500	— — — —	— 1,000 1,000 —	ns ns ns ns	RC osc mode XT osc mode - PIC17C42-16 - PIC17C42-25 LF osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	160	4/Fosc	DC	ns	
3	TosL, TosH	**Clock in (OSC1) High or Low Time**	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	**Clock in (OSC1) Rise or Fall Time**	—	—	5 ‡	ns	EC oscillator

† Data in “Typ” column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1 pin.
When an external clock input is used, the “Max.” cycle time limit is “DC” (no clock) for all devices.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 17-9: USART MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

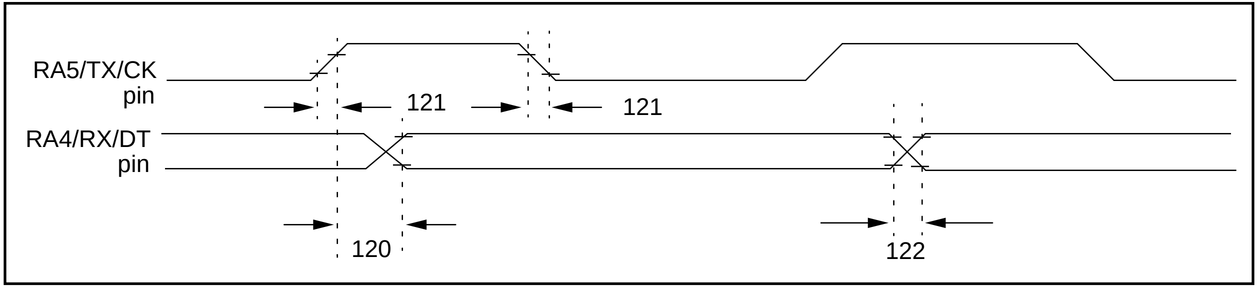


TABLE 17-9: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	—	—	65	ns	
121	TckRF	Clock out rise time and fall time (Master Mode)	—	10	35	ns	
122	TdtRF	Data out rise time and fall time	—	10	35	ns	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-10: USART MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

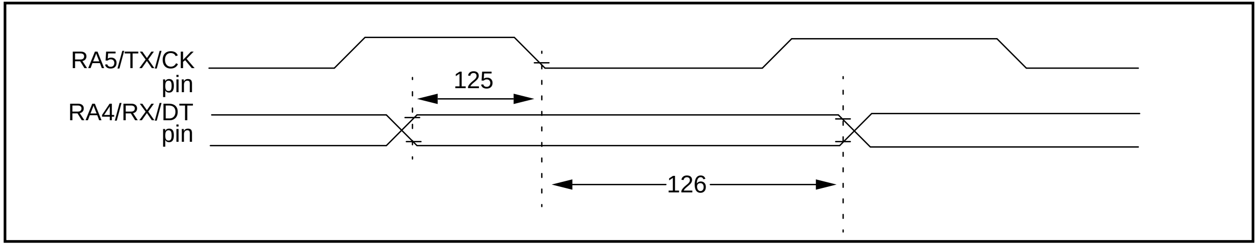


TABLE 17-10: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data hold before CK↓ (DT hold time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 17-12: MEMORY INTERFACE READ TIMING

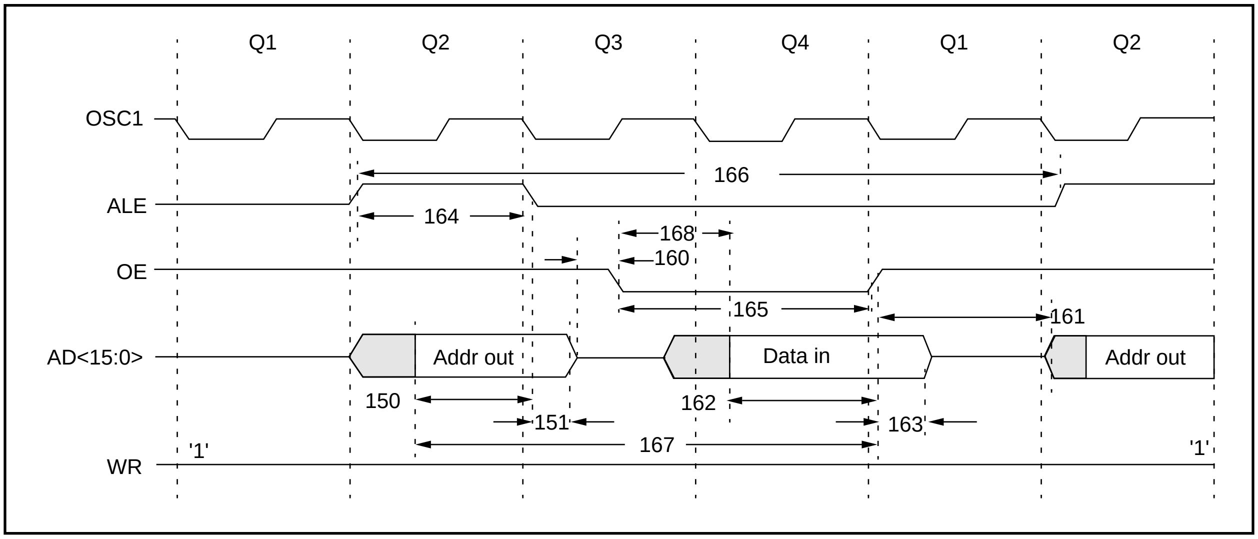


TABLE 17-12: MEMORY INTERFACE READ REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	5*	—	—	ns	
160	TadZ2oeL	AD<15:0> high impedance to $\overline{\text{OE}}$↓	0*	—	—	ns	
161	ToeH2adD	$\overline{\text{OE}}$↑ to AD<15:0> driven	0.25Tcy - 15	—	—	ns	
162	TadV2oeH	Data in valid before $\overline{\text{OE}}$↑ (data setup time)	35	—	—	ns	
163	ToeH2adI	$\overline{\text{OE}}$↑to data in invalid (data hold time)	0	—	—	ns	
164	TalH	ALE pulse width	—	0.25TCY §	—	ns	
165	ToeL	$\overline{\text{OE}}$ pulse width	0.5Tcy - 35 §	—	—	ns	
166	TalH2alH	ALE↑ to ALE↑ (cycle time)	—	TCY §	—	ns	
167	Tacc	Address access time	—	—	0.75 TCY-40	ns	
168	Toe	Output enable access time ($\overline{\text{OE}}$ low to Data Valid)	—	—	0.5 TCY - 60	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

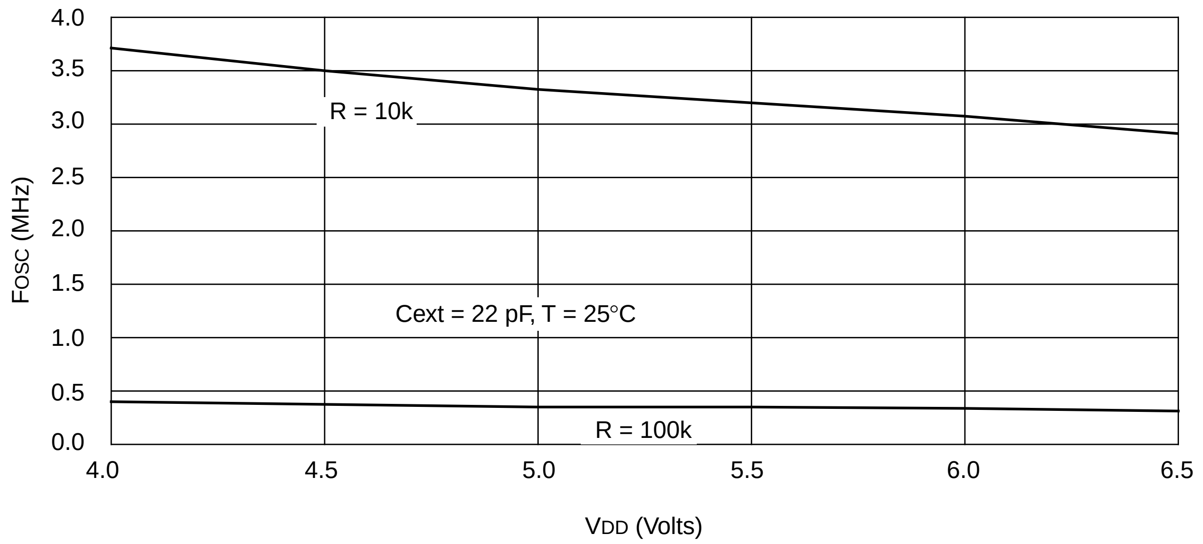
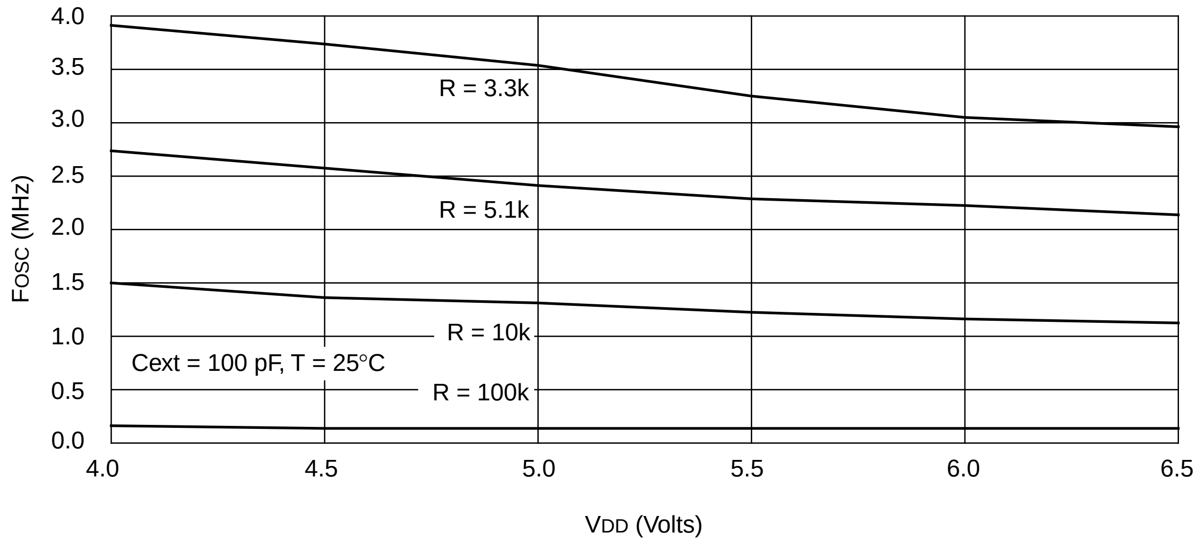


FIGURE 18-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

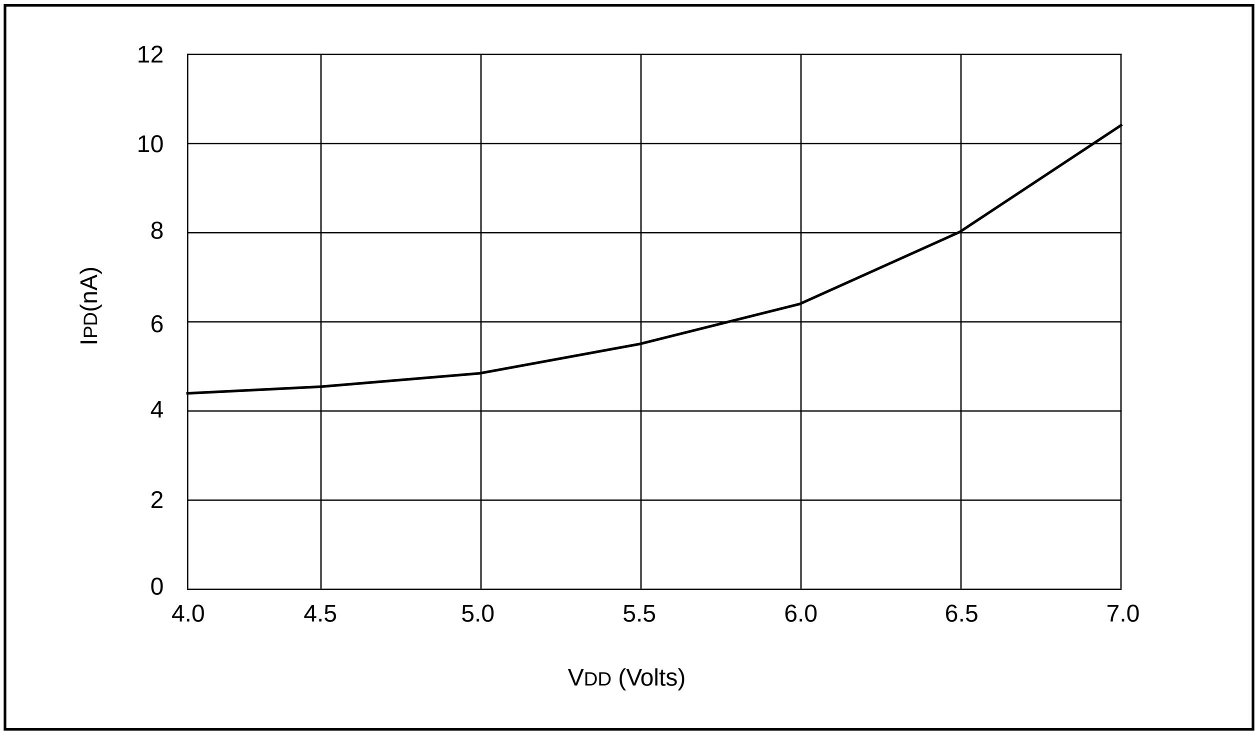


FIGURE 18-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

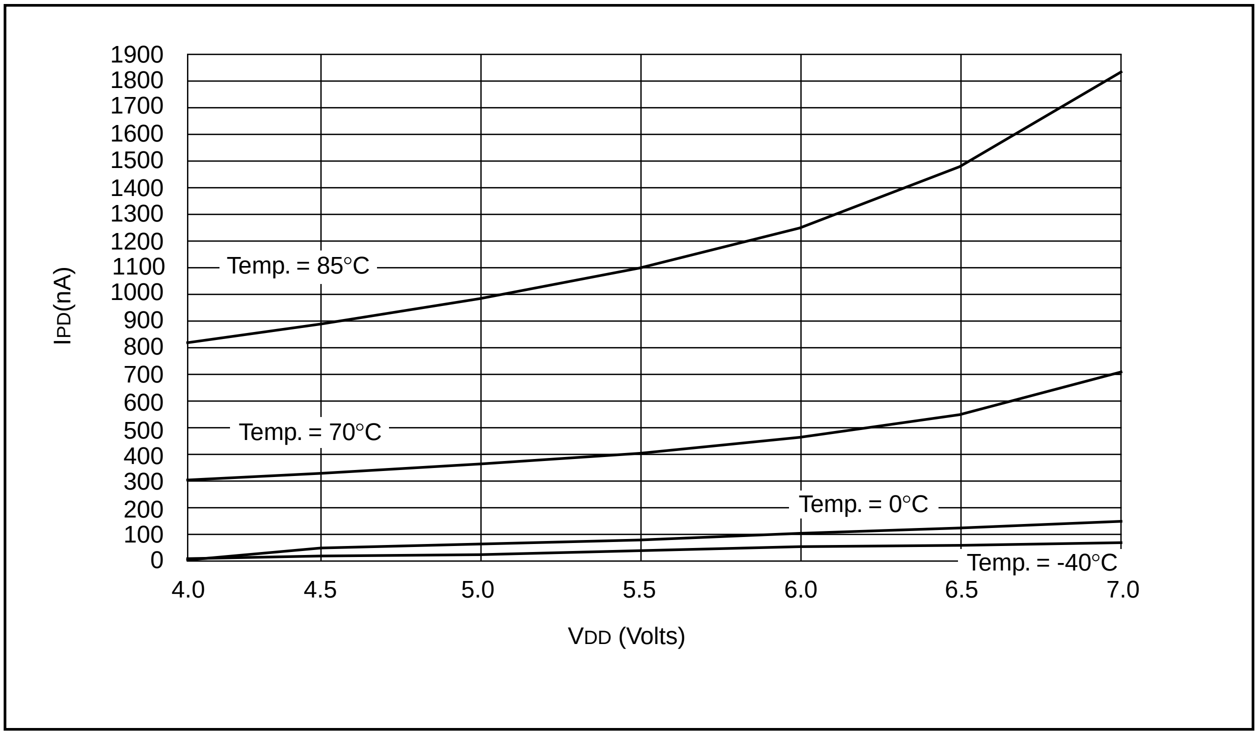


FIGURE 19-5: TIMER0 CLOCK TIMINGS

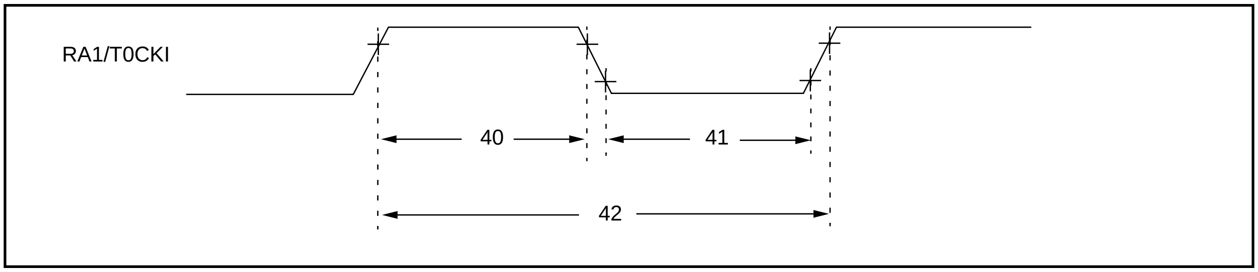


TABLE 19-5: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		Greater of: 20 ns or $\frac{\text{Tcy + 40 §}}{\text{N}}$	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

FIGURE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS

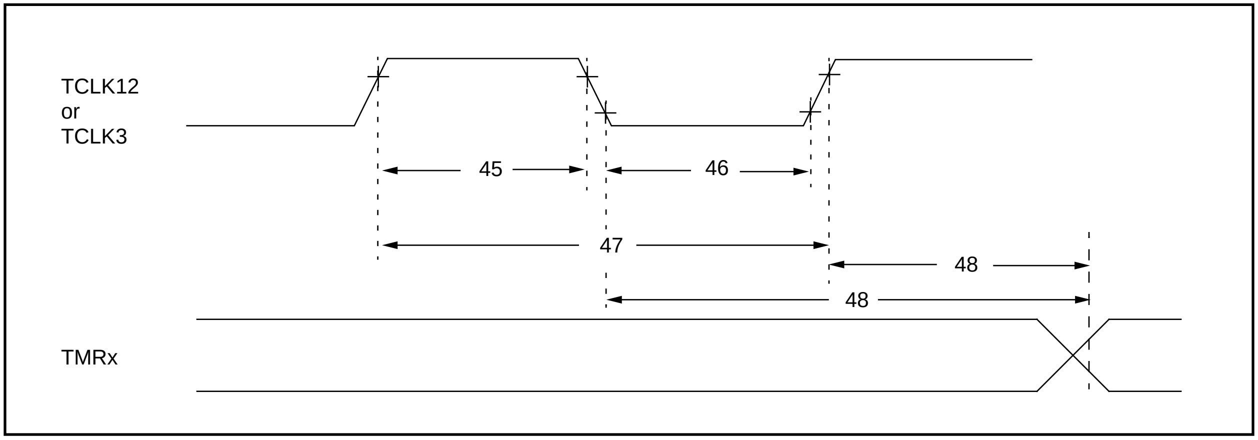


TABLE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

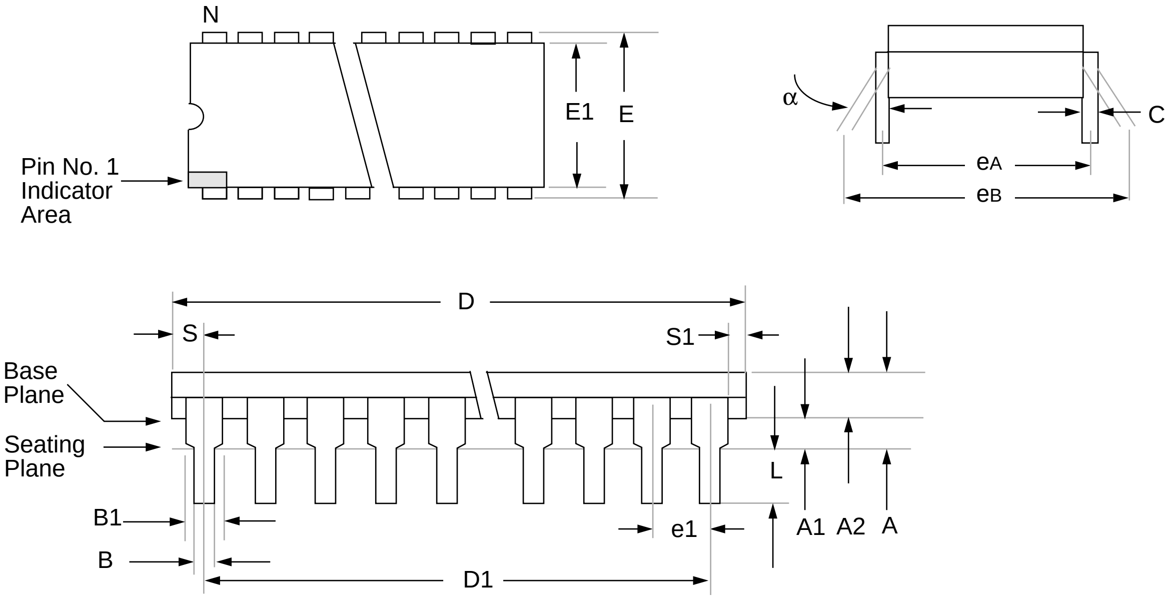
Parameter No.	Sym	Characteristic	Min	Typ †	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5TCY + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5TCY + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	$\frac{\text{TCY + 40}}{\text{N}}$ §	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrI	Delay from selected External Clock Edge to Timer increment	2TOSC §		6Tosc §		

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

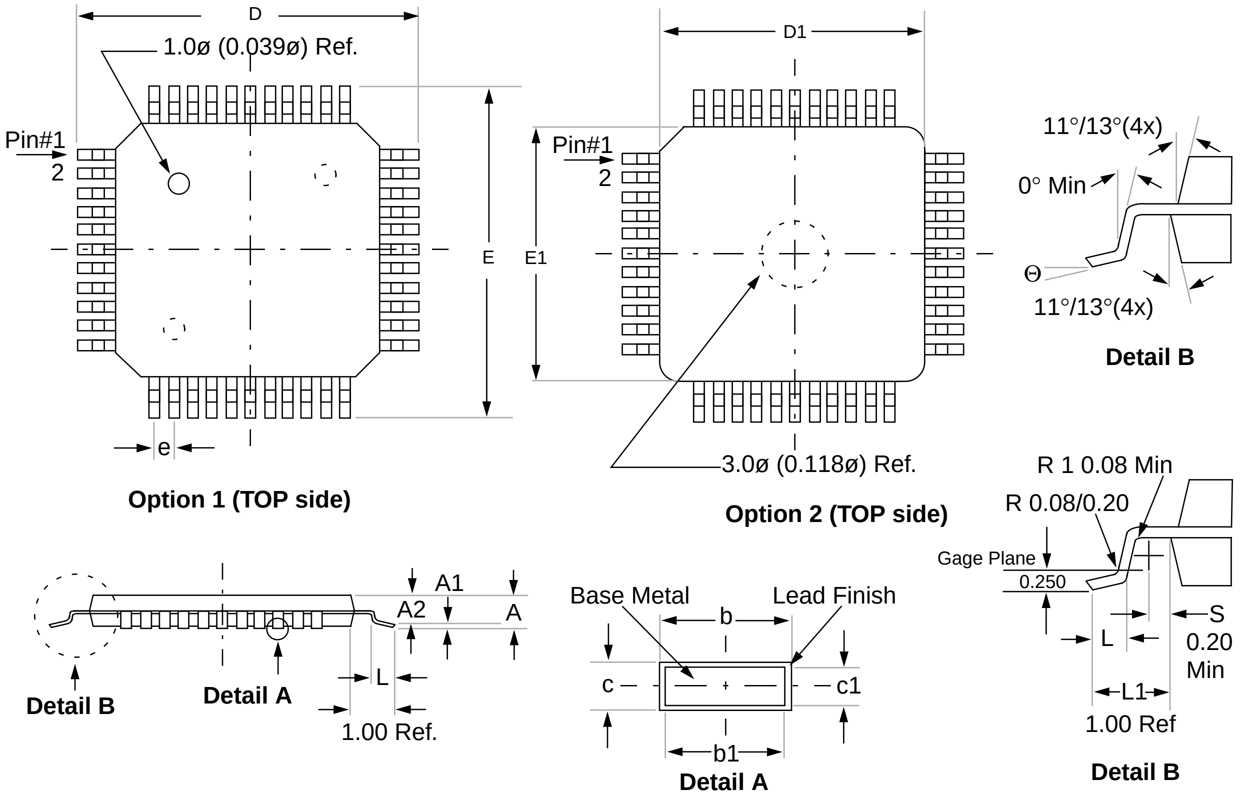
§ This specification ensured by design.

21.2 40-Lead Plastic Dual In-line (600 mil)



Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.381	–		0.015	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	**Typical**	0.050	0.070	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	**Reference**	1.900	1.900	**Reference**
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	**Typical**	0.098	0.102	**Typical**
eA	15.240	15.240	**Reference**	0.600	0.600	**Reference**
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	40	40		40	40	
S	1.270	–		0.050	–	
S1	0.508	–		0.020	–	

21.5 44-Lead Plastic Surface Mount (TQFP 10x10 mm Body 1.0/0.10 mm Lead Form)



Option 1 (TOP side)

Option 2 (TOP side)

Detail B

Detail B

Detail A

Detail B

Detail A

Detail B

Package Group: Plastic TQFP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.00	1.20		0.039	0.047	
A1	0.05	0.15		0.002	0.006	
A2	0.95	1.05		0.037	0.041	
D	11.75	12.25		0.463	0.482	
D1	9.90	10.10		0.390	0.398	
E	11.75	12.25		0.463	0.482	
E1	9.90	10.10		0.390	0.398	
L	0.45	0.75		0.018	0.030	
e	0.80 BSC			0.031 BSC		
b	0.30	0.45		0.012	0.018	
b1	0.30	0.40		0.012	0.016	
c	0.09	0.20		0.004	0.008	
c1	0.09	0.16		0.004	0.006	
N	44	44		44	44	
Θ	0°	7°		0°	7°	

Note 1: Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25m/m (0.010”) per side. D1 and E1 dimensions including mold mismatch.

2: Dimension “b” does not include Dambar protrusion, allowable Dambar protrusion shall be 0.08m/m (0.003”)max.

3: This outline conforms to JEDEC MS-026.

APPENDIX A: MODIFICATIONS

The following is the list of modifications over the PIC16CXX microcontroller family:

1. Instruction word length is increased to 16-bit. This allows larger page sizes both in program memory (8 Kwords verses 2 Kwords) and register file (256 bytes versus 128 bytes).
2. Four modes of operation: microcontroller, protected microcontroller, extended microcontroller, and microprocessor.
3. 22 new instructions.
   The `MOVF`, `TRIS` and `OPTION` instructions have been removed.
4. 4 new instructions for transferring data between data memory and program memory. This can be used to “self program” the EPROM program memory.
5. Single cycle data memory to data memory transfers possible (`MOVPF` and `MOVFP` instructions). These instructions do not affect the Working register (WREG).
6. W register (WREG) is now directly addressable.
7. A PC high latch register (PCLATH) is extended to 8-bits. The PCLATCH register is now both readable and writable.
8. Data memory paging is redefined slightly.
9. DDR registers replaces function of TRIS registers.
10. Multiple Interrupt vectors added. This can decrease the latency for servicing the interrupt.
11. Stack size is increased to 16 deep.
12. BSR register for data memory paging.
13. Wake up from SLEEP operates slightly differently.
14. The Oscillator Start-Up Timer (OST) and Power-Up Timer (PWRT) operate in parallel and not in series.
15. PORTB interrupt on change feature works on all eight port pins.
16. TMR0 is 16-bit plus 8-bit prescaler.
17. Second indirect addressing register added (FSR1 and FSR2). Configuration bits can select the FSR registers to auto-increment, auto-decrement, remain unchanged after an indirect address.
18. Hardware multiplier added (8 x 8 → 16-bit) (PIC17C43 and PIC17C44 only).
19. Peripheral modules operate slightly differently.
20. Oscillator modes slightly redefined.
21. Control/Status bits and registers have been placed in different registers and the control bit for globally enabling interrupts has inverse polarity.
22. Addition of a test mode pin.
23. In-circuit serial programming is not implemented.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16CXX to PIC17CXX, the user should take the following steps:

1. Remove any `TRIS` and `OPTION` instructions, and implement the equivalent code.
2. Separate the interrupt service routine into its four vectors.
3. Replace:
   ```
   MOVF    REG1, W
   ```
   with:
   ```
   MOVFP   REG1, WREG
   ```
4. Replace:
   ```
   MOVF    REG1, W
   MOVWF   REG2
   ```
   with:
   ```
   MOVPF   REG1, REG2 ; Addr(REG1)<20h
   ```
   or
   ```
   MOVFP  REG1, REG2 ; Addr(REG2)<20h
   ```

> **Note:** If REG1 and REG2 are both at addresses greater then 20h, two instructions are required.
> ```
> MOVFP   REG1, WREG ;
> MOVPF   WREG, REG2 ;
> ```

5. Ensure that all bit names and register names are updated to new data memory map location.
6. Verify data memory banking.
7. Verify mode of operation for indirect addressing.
8. Verify peripheral routines for compatibility.
9. Weak pull-ups are enabled on reset.

To convert code from the PIC17C42 to all the other PIC17C4X devices, the user should take the following steps:

1. If the hardware multiply is to be used, ensure that any variables at address 18h and 19h are moved to another address.
2. Ensure that the upper nibble of the BSR was not written with a non-zero value. This may cause unexpected operation since the RAM bank is no longer 0.
3. The disabling of global interrupts has been enhanced so there is no additional testing of the GLINTD bit after a `BSF CPUSTA, GLINTD` instruction.

INDEX

NOTES: