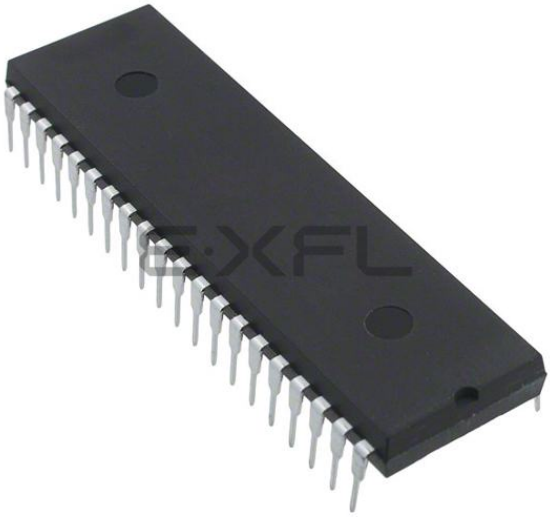


Welcome to E-XFL.COM

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	Active
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 ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c42a-33i-p

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3, and Q4. Internally, the program counter (PC) is incremented every Q1, and the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 3-3.

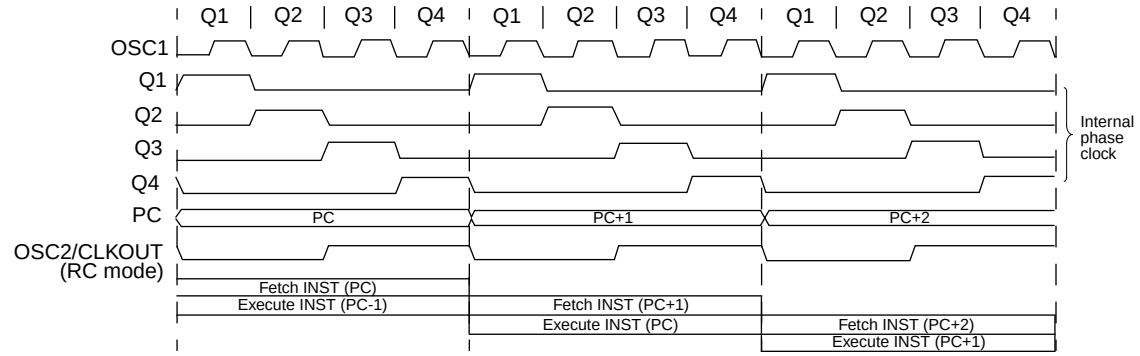
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3, and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. `GOTO`) then two cycles are required to complete the instruction (Example 3-2).

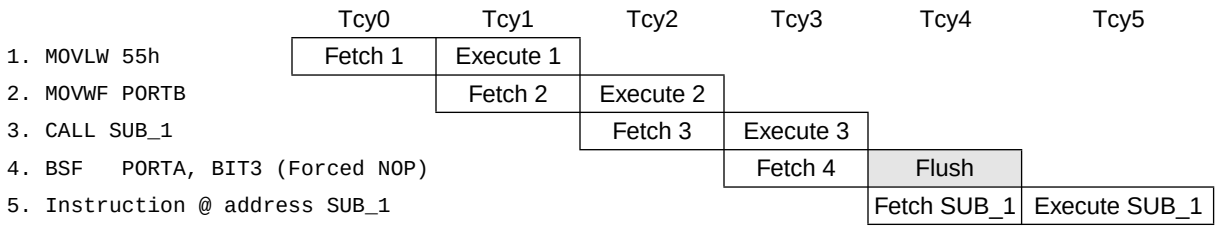
A fetch cycle begins with the program counter incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register (IR)” in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-2: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

5.2 Peripheral Interrupt Enable Register (PIE)

This register contains the individual flag bits for the Peripheral interrupts.

FIGURE 5-3: PIE REGISTER (ADDRESS: 17h, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **RBIE**: PORTB Interrupt on Change Enable bit
1 = Enable PORTB interrupt on change
0 = Disable PORTB interrupt on change

bit 6: **TMR3IE**: Timer3 Interrupt Enable bit
1 = Enable Timer3 interrupt
0 = Disable Timer3 interrupt

bit 5: **TMR2IE**: Timer2 Interrupt Enable bit
1 = Enable Timer2 interrupt
0 = Disable Timer2 interrupt

bit 4: **TMR1IE**: Timer1 Interrupt Enable bit
1 = Enable Timer1 interrupt
0 = Disable Timer1 interrupt

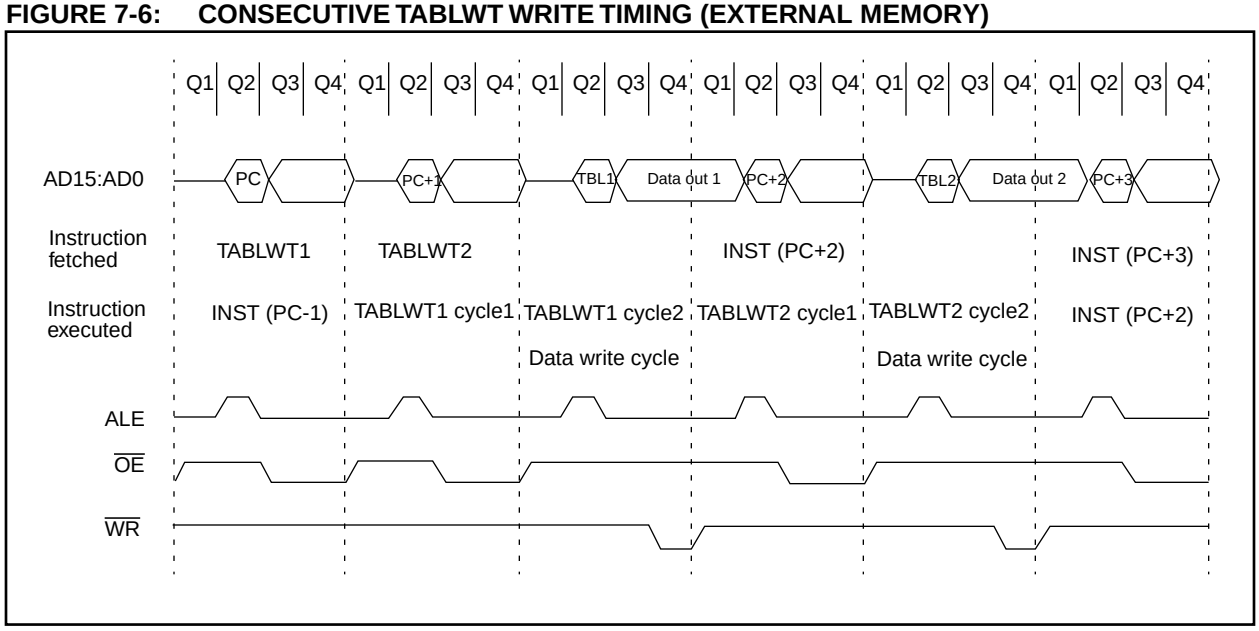
bit 3: **CA2IE**: Capture2 Interrupt Enable bit
1 = Enable Capture interrupt on RB1/CAP2 pin
0 = Disable Capture interrupt on RB1/CAP2 pin

bit 2: **CA1IE**: Capture1 Interrupt Enable bit
1 = Enable Capture interrupt on RB2/CAP1 pin
0 = Disable Capture interrupt on RB2/CAP1 pin

bit 1: **TXIE**: USART Transmit Interrupt Enable bit
1 = Enable Transmit buffer empty interrupt
0 = Disable Transmit buffer empty interrupt

bit 0: **RCIE**: USART Receive Interrupt Enable bit
1 = Enable Receive buffer full interrupt
0 = Disable Receive buffer full interrupt

FIGURE 7-6: CONSECUTIVE TABLWT WRITE TIMING (EXTERNAL MEMORY)



7.3 Table Reads

The table read allows the program memory to be read. This allows constant data to be stored in the program memory space, and retrieved into data memory when needed. Example 7-2 reads the 16-bit value at program memory address TBLPTR. After the dummy byte has been read from the TABLATH, the TABLATH is loaded with the 16-bit data from program memory address TBLPTR + 1. The first read loads the data into the latch, and can be considered a dummy read (unknown data loaded into 'f'). INDF0 should be configured for either auto-increment or auto-decrement.

EXAMPLE 7-2: TABLE READ

```
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH         ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL         ;
TABLRD  0,0,DUMMY       ; Dummy read,
                        ;  Updates TABLATCH
TLRD    1, INDF0        ; Read HI byte
                        ;   of TABLATCH
TABLRD  0,1,INDF0       ; Read LO byte
                        ;   of TABLATCH and
                        ;   Update TABLATCH
```

FIGURE 7-7: TABLRD TIMING

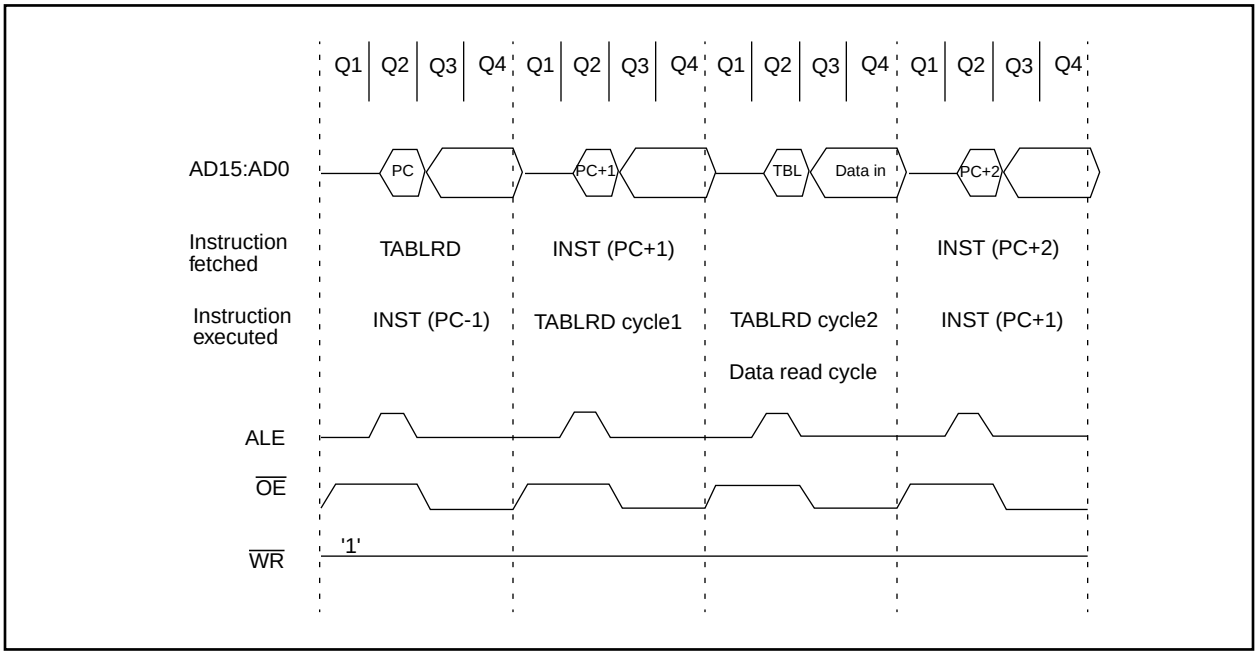
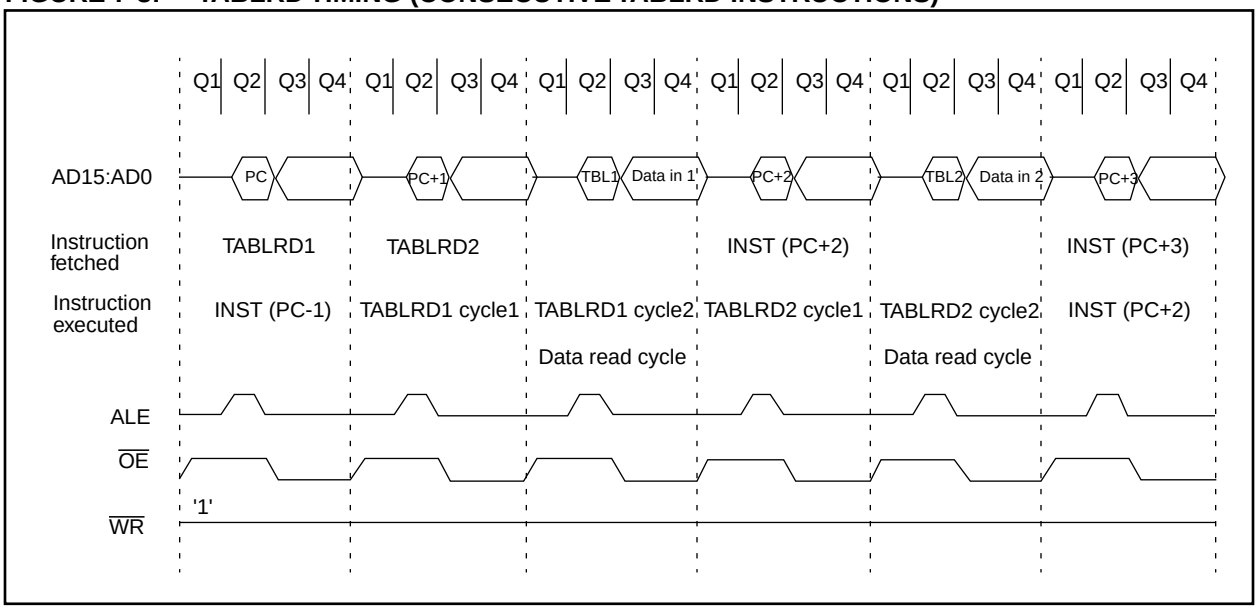


FIGURE 7-8: TABLRD TIMING (CONSECUTIVE TABLRD INSTRUCTIONS)



Example 9-1 shows the instruction sequence to initialize PORTB. The Bank Select Register (BSR) must be selected to Bank 0 for the port to be initialized.

EXAMPLE 9-1: INITIALIZING PORTB

```
MOVLB 0         ; Select Bank 0
CLRF  PORTB     ; Initialize PORTB by clearing
                ;  output data latches
MOVLW 0xCF      ; Value used to initialize
                ;  data direction
MOVWF DDRB      ;  Set RB<3:0> as inputs
                ;   RB<5:4> as outputs
                ;   RB<7:6> as inputs
```

TABLE 9-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/Output or the RB0/CAP1 input pin. Software programmable weak pull-up and interrupt on change features.
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pull-up and interrupt on change features.
RB2/PWM1	bit2	ST	Input/Output or the RB2/PWM1 output pin. Software programmable weak pull-up and interrupt on change features.
RB3/PWM2	bit3	ST	Input/Output or the RB3/PWM2 output pin. Software programmable weak pull-up and interrupt on change features.
RB4/TCLK12	bit4	ST	Input/Output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt on change features.
RB5/TCLK3	bit5	ST	Input/Output or the external clock input to Timer3. Software programmable weak pull-up and interrupt on change features.
RB6	bit6	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.
RB7	bit7	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.

Legend: ST = Schmitt Trigger input.

TABLE 9-4: REGISTERS/BITS ASSOCIATED WITH PORTB

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)
12h, Bank 0	PORTB	PORTB data latch								xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data direction register for PORTB								1111 1111	1111 1111
10h, Bank 0	PORTA	$\overline{RBPU}$	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 qq--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
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
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/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = Value depends on condition.
Shaded cells are not used by PORTB.

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

9.4.1 PORTE AND DDRE REGISTER

PORTE is a 3-bit bi-directional port. The corresponding data direction register is DDRE. A '1' in DDRE configures the corresponding port pin as an input. A '0' in the DDRE register configures the corresponding port pin as an output. Reading PORTE reads the status of the pins, whereas writing to it will write to the port latch. PORTE is multiplexed with the system bus. When operating as the system bus, PORTE contains the control signals for the address/data bus (AD15:AD0). These control signals are Address Latch Enable (ALE), Output Enable ($\overline{OE}$), and Write ($\overline{WR}$). The control signals $\overline{OE}$ and $\overline{WR}$ are active low signals. The timing for the system bus is shown in the Electrical Characteristics section.

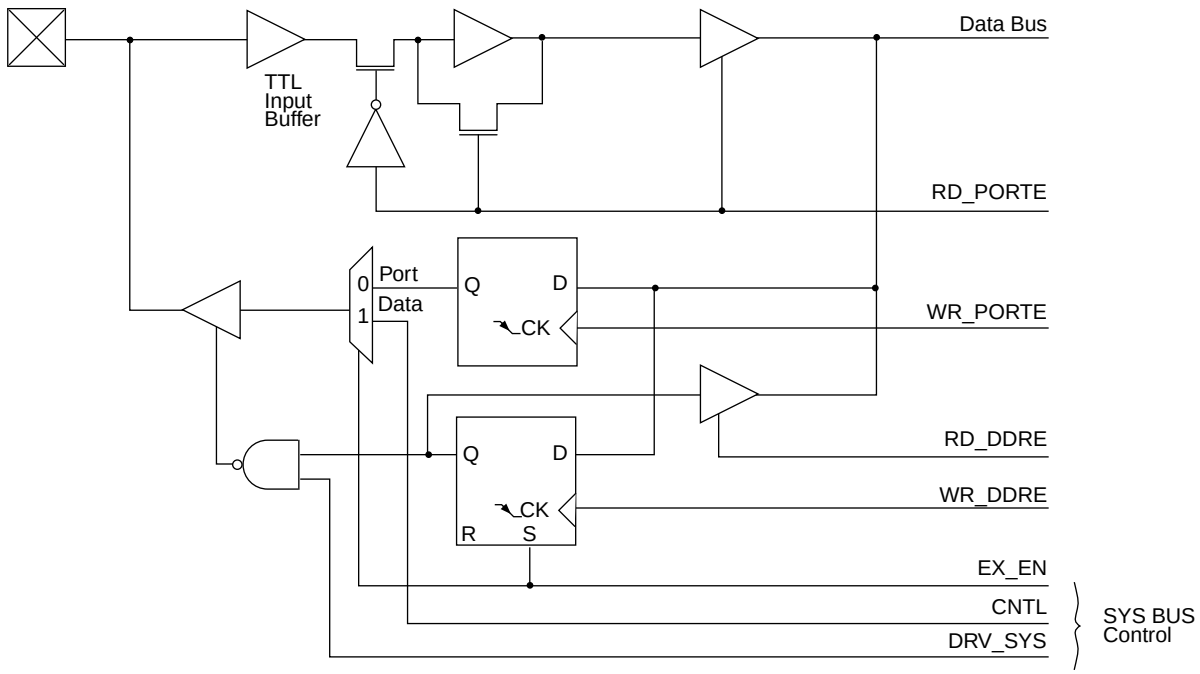
Note:	This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-4 shows the instruction sequence to initialize PORTE. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

EXAMPLE 9-4: INITIALIZING PORTE

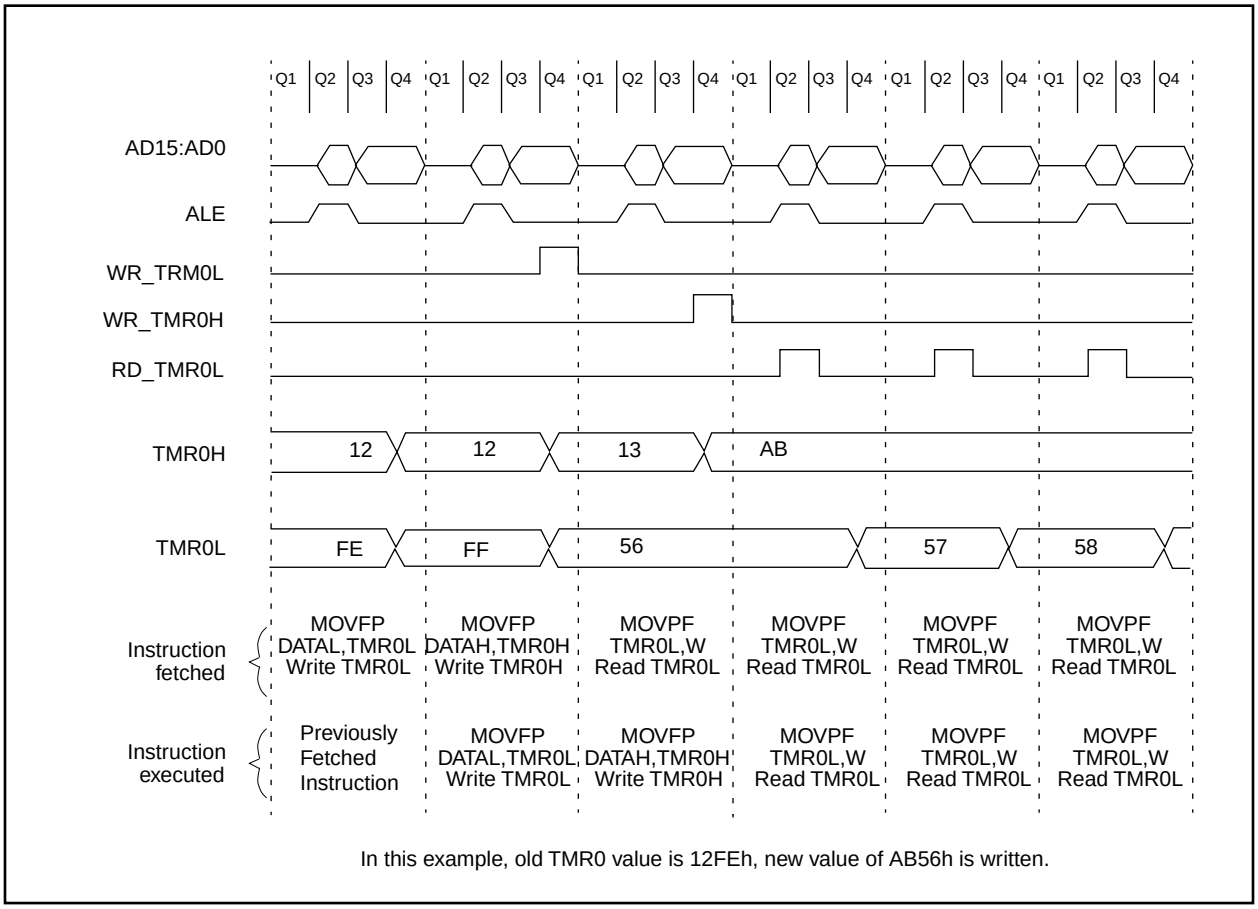
```
MOVLB  1          ;  Select Bank 1
CLRF   PORTE      ;  Initialize PORTE data
                  ;   latches before setting
                  ;   the data direction
                  ;   register
MOVLW  0x03       ;  Value used to initialize
                  ;   data direction
MOVWF  DDRE       ;  Set RE<1:0> as inputs
                  ;   RE<2> as outputs
                  ;   RE<7:3> are always
                  ;   read as '0'
```

FIGURE 9-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and Vss.

FIGURE 11-5: TMR0 READ/WRITE IN TIMER MODE



In this example, old TMR0 value is 12FEh, new value of AB56h is written.

TABLE 11-1: REGISTERS/BITS ASSOCIATED WITH TIMER0

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)
05h, Unbanked	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—	0000 000-	0000 000-
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 qq--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
0Bh, Unbanked	TMR0L	TMR0 register; low byte								xxxx xxxx	uuuu uuuu
0Ch, Unbanked	TMR0H	TMR0 register; high byte								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 Timer0.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer Reset.

14.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC17CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- OSC selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection

The PIC17CXX has a Watchdog Timer which can be shut off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external reset, Watchdog Timer Reset or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 14-1.

FIGURE 14-1: CONFIGURATION WORD

R/P - 1	U - x	U - x	U - x	U - x	U - x	U - x	U - x
PM2 (1)	—	—	—	—	—	—	—
bit15-7							bit0

U - x	R/P - 1	U - x	R/P - 1	R/P - 1	R/P - 1	R/P - 1	R/P - 1
—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0
bit15-7							bit0

R = Readable bit
P = Programmable bit
U = Unimplemented
- n = Value for Erased Device
(x = unknown)

bit 15-9: **Unimplemented**: Read as a '1'

bit 15,6,4:**PM2, PM1, PM0**, Processor Mode Select bits
111 = Microprocessor Mode
110 = Microcontroller mode
101 = Extended microcontroller mode
000 = Code protected microcontroller mode

bit 7, 5: **Unimplemented**: Read as a '0'

bit 3-2: **WDTPS1:WDTPS0**, WDT Postscaler Select bits
11 = WDT enabled, postscaler = 1
10 = WDT enabled, postscaler = 256
01 = WDT enabled, postscaler = 64
00 = WDT disabled, 16-bit overflow timer

bit 1-0: **FOSC1:FOSC0**, Oscillator Select bits
11 = EC oscillator
10 = XT oscillator
01 = RC oscillator
00 = LF oscillator

Note 1: This bit does not exist on the PIC17C42. Reading this bit will return an unknown value (x).

TABLE 15-2: PIC17CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	**f,d**	ADD WREG to f	1	0000 111d ffff ffff	OV,C,DC,Z	
ADDWFC	**f,d**	ADD WREG and Carry bit to f	1	0001 000d ffff ffff	OV,C,DC,Z	
ANDWF	**f,d**	AND WREG with f	1	0000 101d ffff ffff	Z	
CLRF	**f,s**	Clear f, or Clear f and Clear WREG	1	0010 100s ffff ffff	None	3
COMF	**f,d**	Complement f	1	0001 001d ffff ffff	Z	
CPFSEQ	**f**	Compare f with WREG, skip if f = WREG	1 (2)	0011 0001 ffff ffff	None	6,8
CPFSGT	**f**	Compare f with WREG, skip if f > WREG	1 (2)	0011 0010 ffff ffff	None	2,6,8
CPFSLT	**f**	Compare f with WREG, skip if f < WREG	1 (2)	0011 0000 ffff ffff	None	2,6,8
DAW	**f,s**	Decimal Adjust WREG Register	1	0010 111s ffff ffff	C	3
DECF	**f,d**	Decrement f	1	0000 011d ffff ffff	OV,C,DC,Z	
DECFSZ	**f,d**	Decrement f, skip if 0	1 (2)	0001 011d ffff ffff	None	6,8
DCFSNZ	**f,d**	Decrement f, skip if not 0	1 (2)	0010 011d ffff ffff	None	6,8
INCF	**f,d**	Increment f	1	0001 010d ffff ffff	OV,C,DC,Z	
INCFSZ	**f,d**	Increment f, skip if 0	1 (2)	0001 111d ffff ffff	None	6,8
INFSNZ	**f,d**	Increment f, skip if not 0	1 (2)	0010 010d ffff ffff	None	6,8
IORWF	**f,d**	Inclusive OR WREG with f	1	0000 100d ffff ffff	Z	
MOVFP	**f,p**	Move f to p	1	011p pppp ffff ffff	None	
MOVPF	**p,f**	Move p to f	1	010p pppp ffff ffff	Z	
MOVWF	**f**	Move WREG to f	1	0000 0001 ffff ffff	None	
MULWF	**f**	Multiply WREG with f	1	0011 0100 ffff ffff	None	9
NEGW	**f,s**	Negate WREG	1	0010 110s ffff ffff	OV,C,DC,Z	1,3
NOP	**—**	No Operation	1	0000 0000 0000 0000	None	
RLCF	**f,d**	Rotate left f through Carry	1	0001 101d ffff ffff	C	
RLNCF	**f,d**	Rotate left f (no carry)	1	0010 001d ffff ffff	None	
RRCF	**f,d**	Rotate right f through Carry	1	0001 100d ffff ffff	C	
RRNCF	**f,d**	Rotate right f (no carry)	1	0010 000d ffff ffff	None	
SETF	**f,s**	Set f	1	0010 101s ffff ffff	None	3
SUBWF	**f,d**	Subtract WREG from f	1	0000 010d ffff ffff	OV,C,DC,Z	1
SUBWFB	**f,d**	Subtract WREG from f with Borrow	1	0000 001d ffff ffff	OV,C,DC,Z	1
SWAPF	**f,d**	Swap f	1	0001 110d ffff ffff	None	
TABLRD	**t,i,f**	Table Read	2 (3)	1010 10ti ffff ffff	None	7

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.

IORWF **Inclusive OR WREG with f**

Syntax: [*label*] IORWF f,d

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

Operation: (WREG) .OR. (f) → (dest)

Status Affected: Z

Encoding:

0000	100d	ffff	ffff

Description: Inclusive OR WREG with register 'f'. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
IORWF  RESULT, 0
```
Before Instruction
RESULT = 0x13
WREG = 0x91

After Instruction
RESULT = 0x13
WREG = 0x93

LCALL **Long Call**

Syntax: [*label*] LCALL k

Operands: $0 \le k \le 255$

Operation: PC + 1 → TOS;
k → PCL, (PCLATH) → PCH

Status Affected: None

Encoding:

1011	0111	kkkk	kkkk

Description: LCALL allows an unconditional subroutine call to anywhere within the 64k program memory space.

First, the return address (PC + 1) is pushed onto the stack. A 16-bit destination address is then loaded into the program counter. The lower 8-bits of the destination address is embedded in the instruction. The upper 8-bits of PC is loaded from PC high holding latch, PCLATH.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Execute	Write register PCL
Forced NOP	NOP	Execute	NOP

Example:
```
MOVLW  HIGH(SUBROUTINE)
MOVPF  WREG, PCLATH
LCALL  LOW(SUBROUTINE)
```
Before Instruction
SUBROUTINE = 16-bit Address
PC = ?

After Instruction
PC = Address (SUBROUTINE)

MULLW **Multiply Literal with WREG**

Syntax: [*label*] MULLW k

Operands: $0 \le k \le 255$

Operation: (k x WREG) → PRODH:PRODL

Status Affected: None

Encoding:

1011	1100	kkkk	kkkk

Description: An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte.
WREG is unchanged.
None of the status flags are affected.
Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Execute	Write registers PRODH: PRODL

Example: `MULLW 0xC4`

Before Instruction
WREG = 0xE2
PRODH = ?
PRODL = ?

After Instruction
WREG = 0xC4
PRODH = 0xAD
PRODL = 0x08

Note: This instruction is not available in the PIC17C42 device.

MULWF **Multiply WREG with f**

Syntax: [*label*] MULWF f

Operands: $0 \le f \le 255$

Operation: (WREG x f) → PRODH:PRODL

Status Affected: None

Encoding:

0011	0100	ffff	ffff

Description: An unsigned multiplication is carried out between the contents of WREG and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte.
Both WREG and 'f' are unchanged.
None of the status flags are affected.
Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write registers PRODH: PRODL

Example: `MULWF REG`

Before Instruction
WREG = 0xC4
REG = 0xB5
PRODH = ?
PRODL = ?

After Instruction
WREG = 0xC4
REG = 0xB5
PRODH = 0x8A
PRODL = 0x94

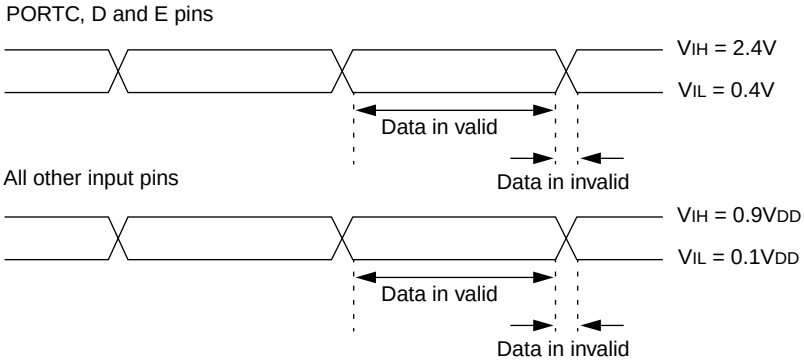
Note: This instruction is not available in the PIC17C42 device.

Applicable Devices	42	R42	42A	43	R43	44

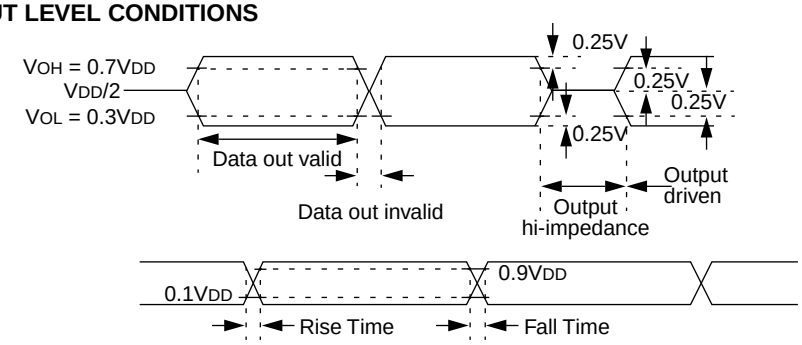
FIGURE 17-1: PARAMETER MEASUREMENT INFORMATION

All timings are measure between high and low measurement points as indicated in the figures below.

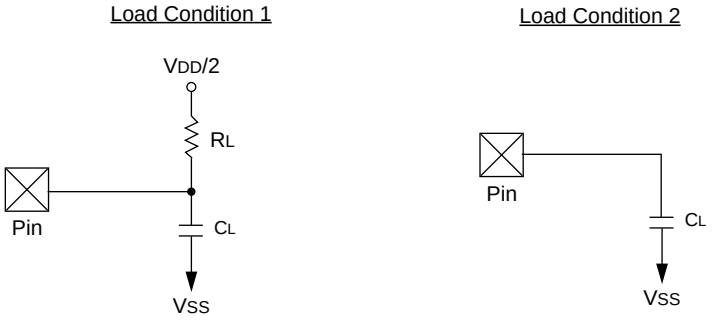
INPUT LEVEL CONDITIONS



OUTPUT LEVEL CONDITIONS



LOAD CONDITIONS



R_L = 464
$C_L \leq$ 50 pF

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 17-5: TIMER0 CLOCK TIMINGS

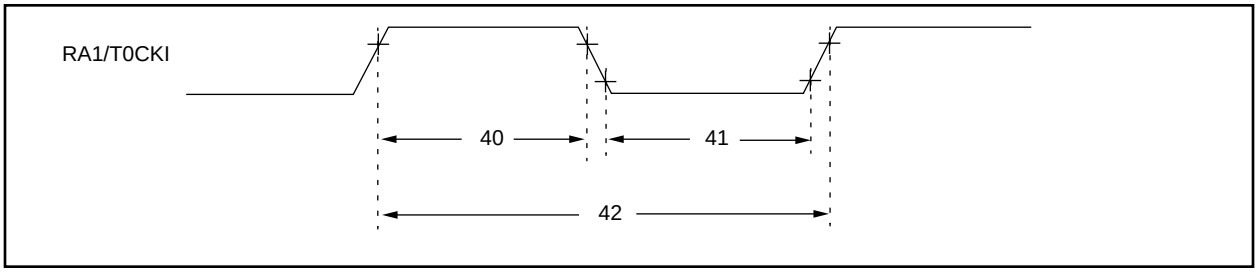


TABLE 17-5: TIMER0 CLOCK REQUIREMENTS

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		TCY + 40 § / 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 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS

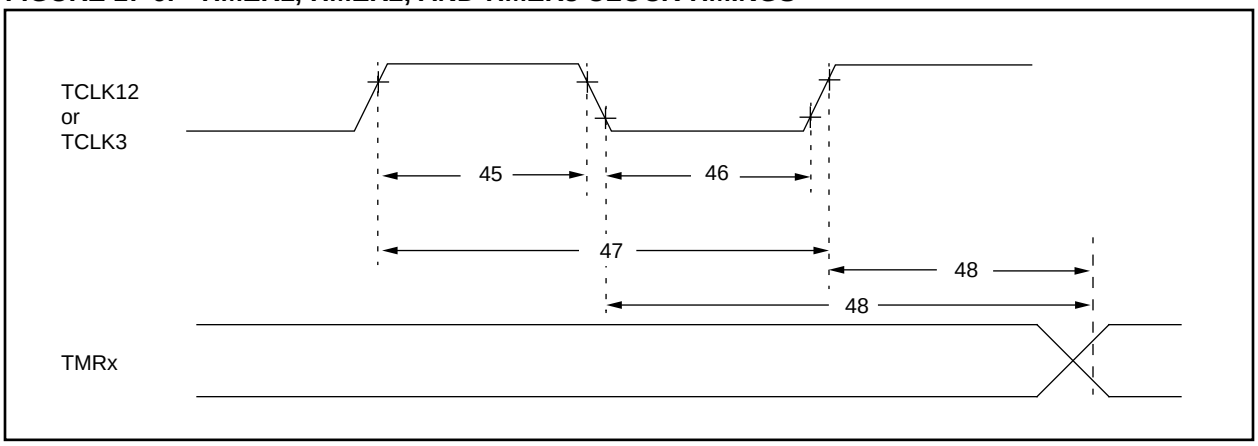


TABLE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

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

* 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.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

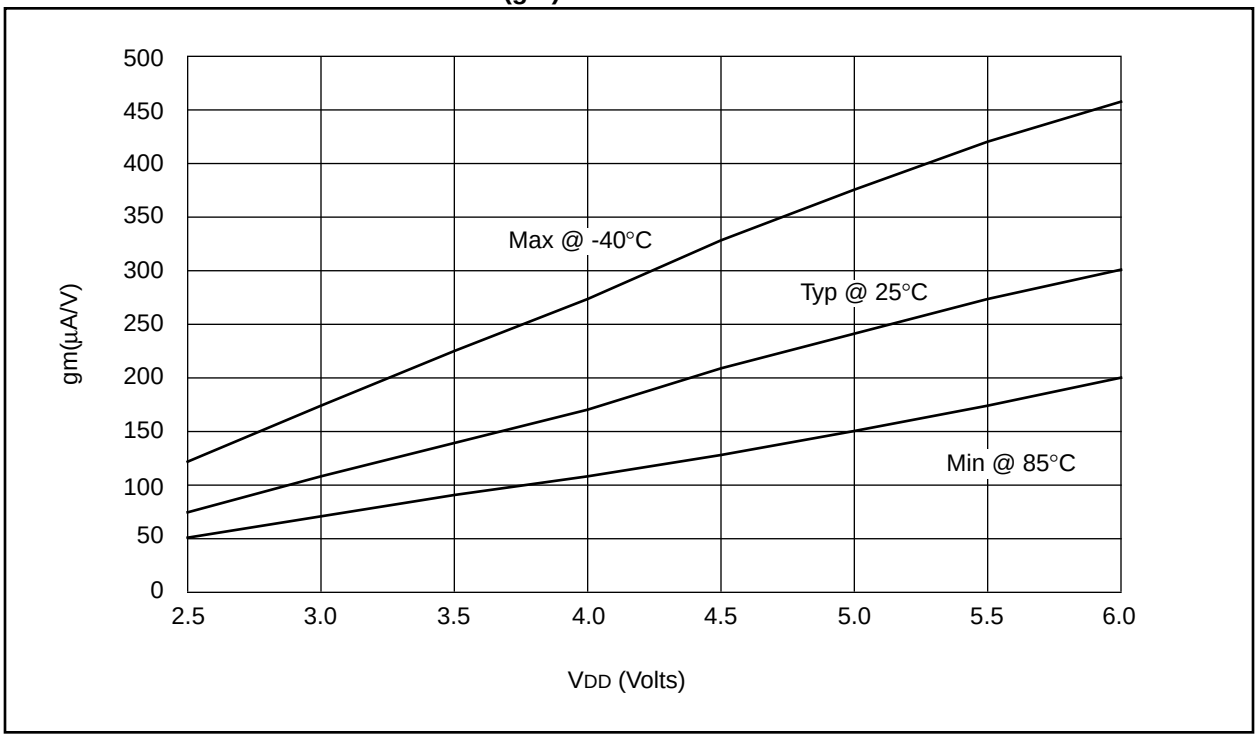
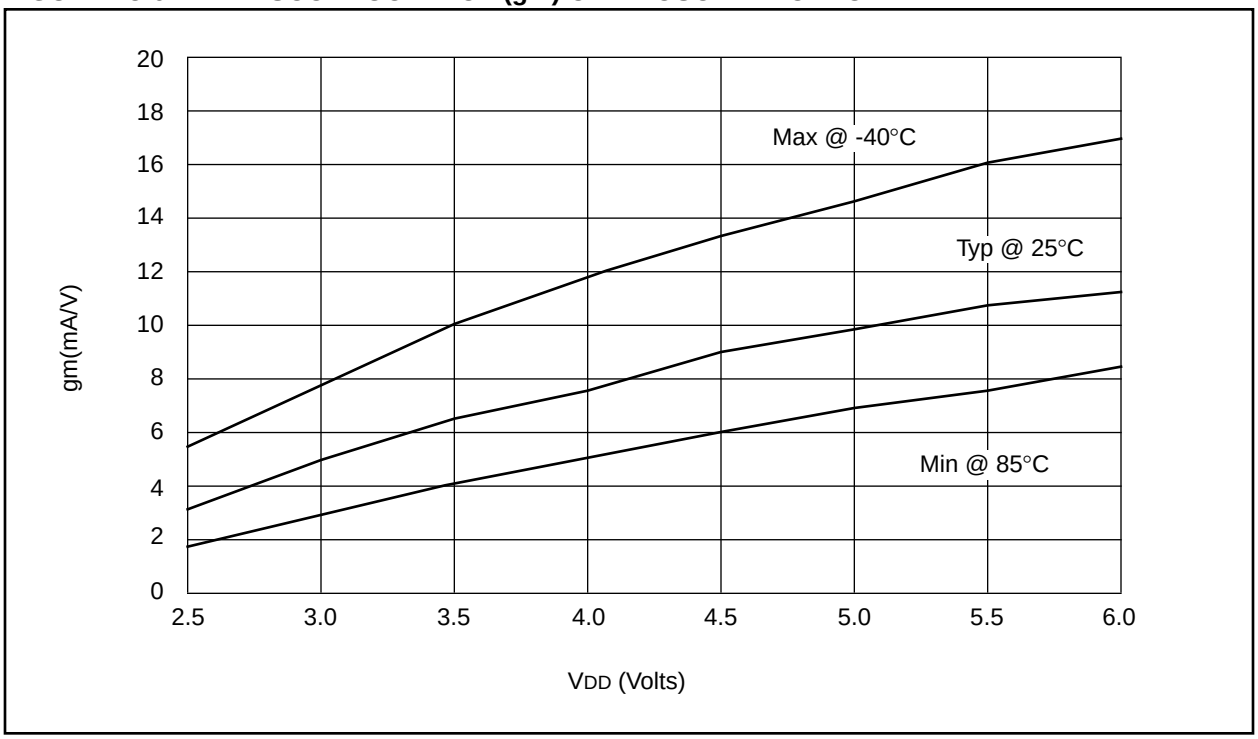


FIGURE 18-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD



Applicable Devices	42	R42	42A	43	R43	44

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

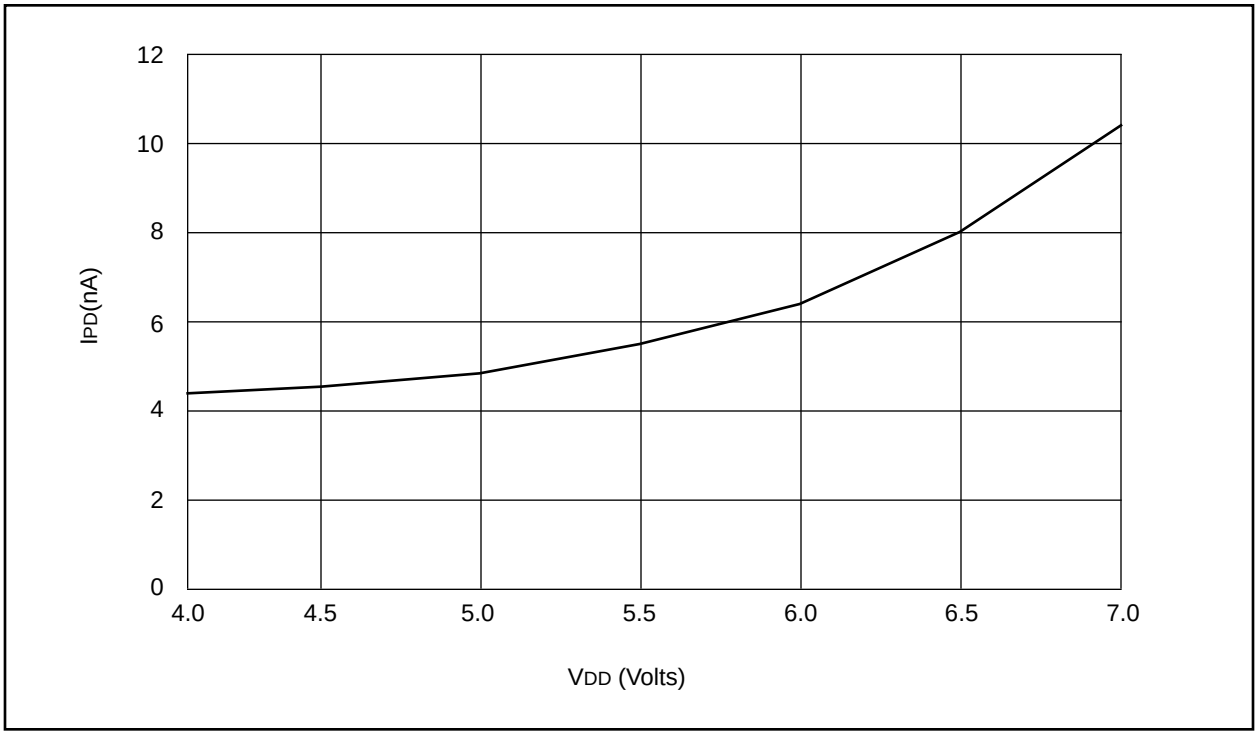
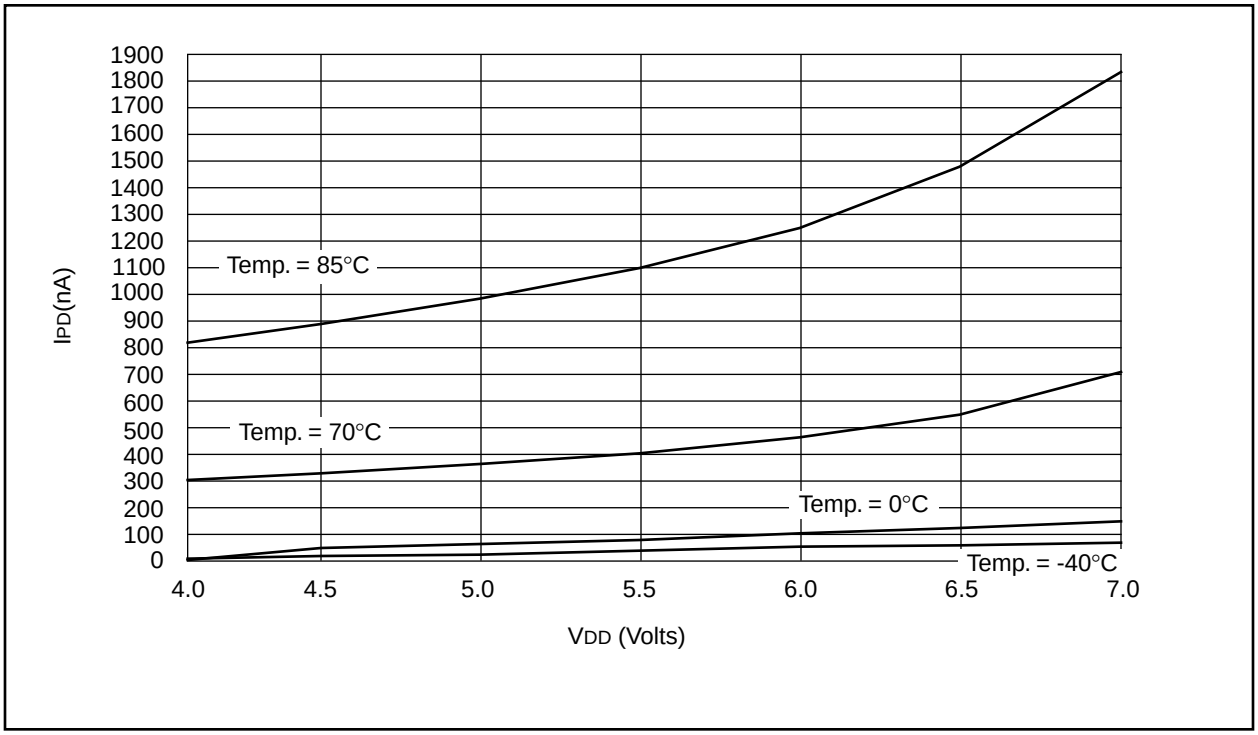


FIGURE 18-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-17: IOL vs. VOL, VDD = 5V

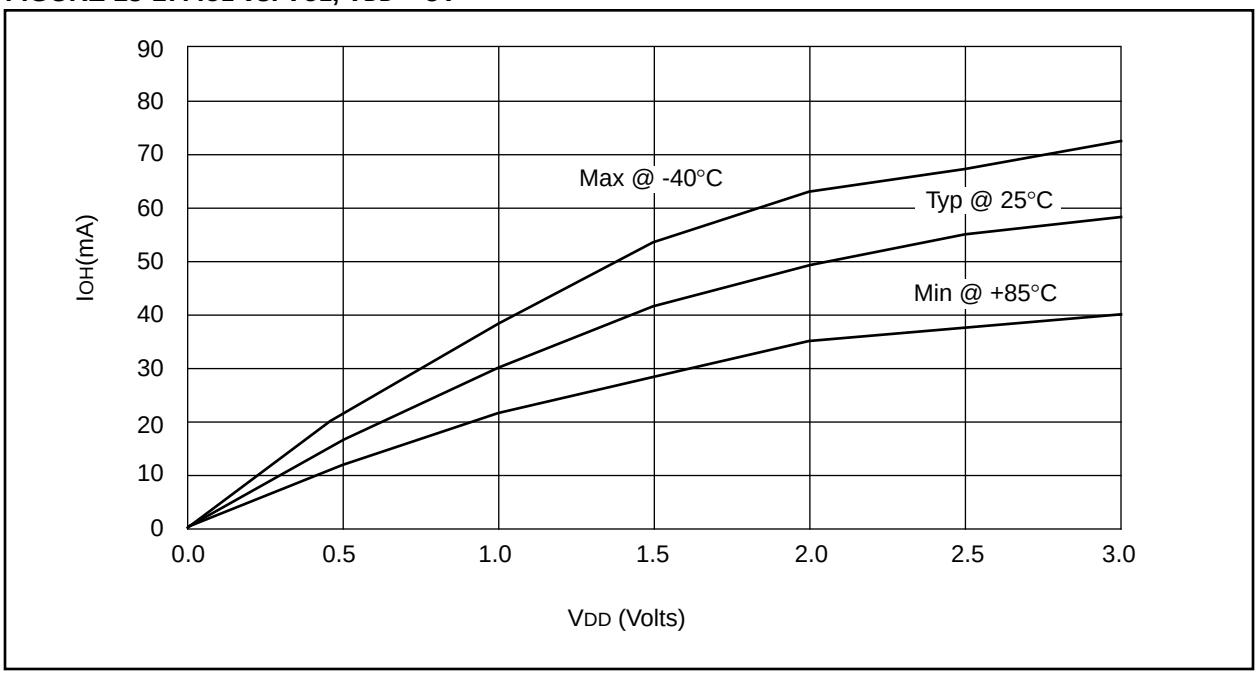
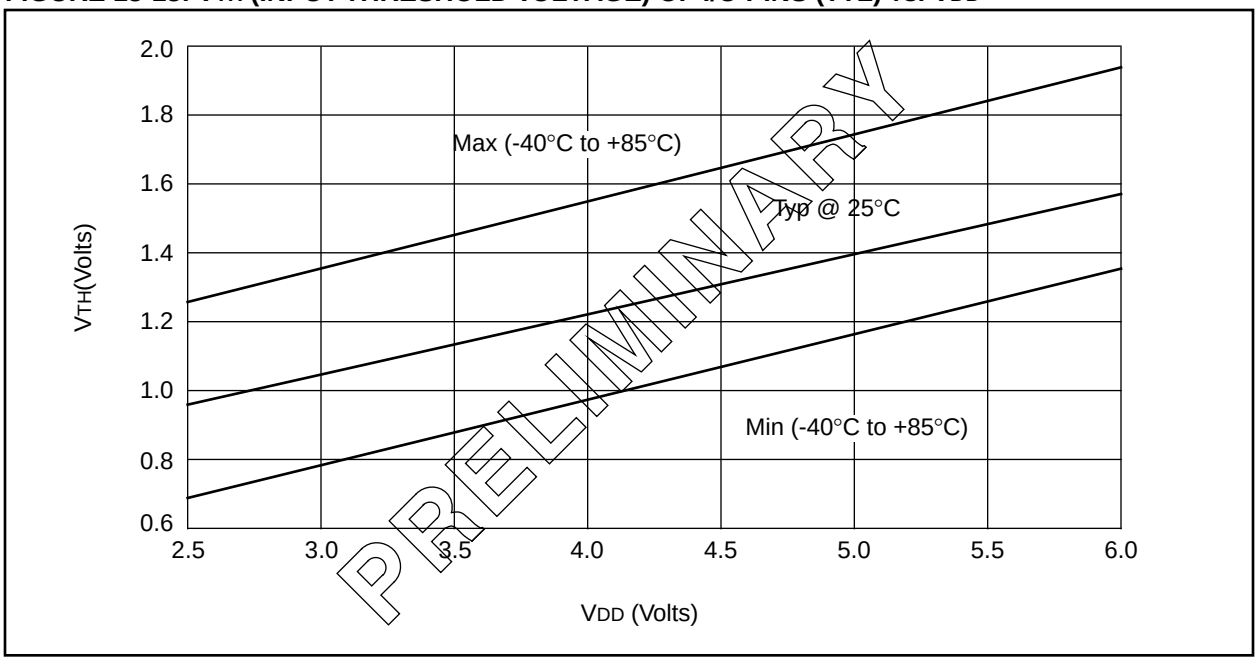


FIGURE 18-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. VDD



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 19-9: USART MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

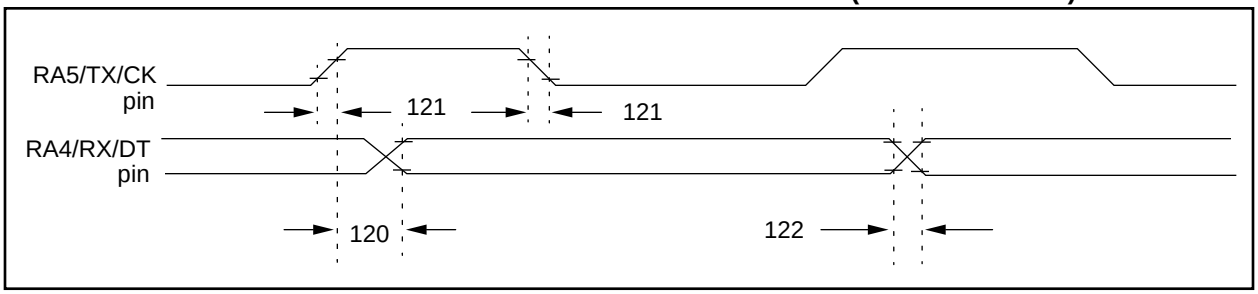


TABLE 19-9: SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	PIC17CR42/42A/43/R43/44	—	—	50	ns	
			PIC17LCR42/42A/43/R43/44	—	—	75	ns	
121	TckRF	Clock out rise time and fall time (Master Mode)	PIC17CR42/42A/43/R43/44	—	—	25	ns	
			PIC17LCR42/42A/43/R43/44	—	—	40	ns	
122	TdtRF	Data out rise time and fall time	PIC17CR42/42A/43/R43/44	—	—	25	ns	
			PIC17LCR42/42A/43/R43/44	—	—	40	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 19-10: USART MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

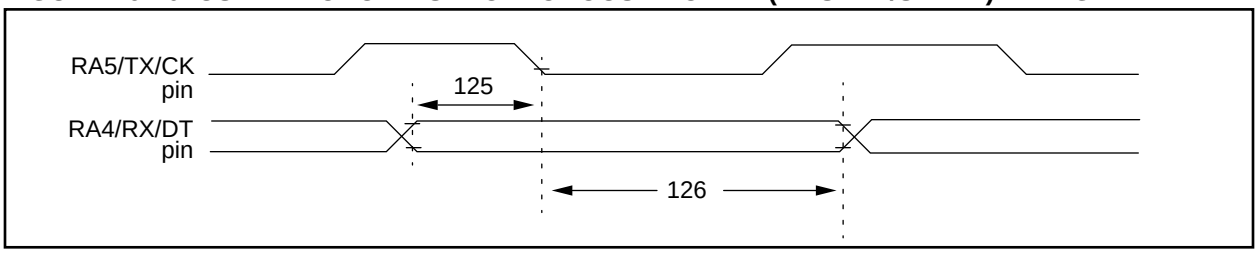


TABLE 19-10: SYNCHRONOUS RECEIVE REQUIREMENTS

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

20.0 PIC17CR42/42A/43/R43/44 DC AND AC CHARACTERISTICS

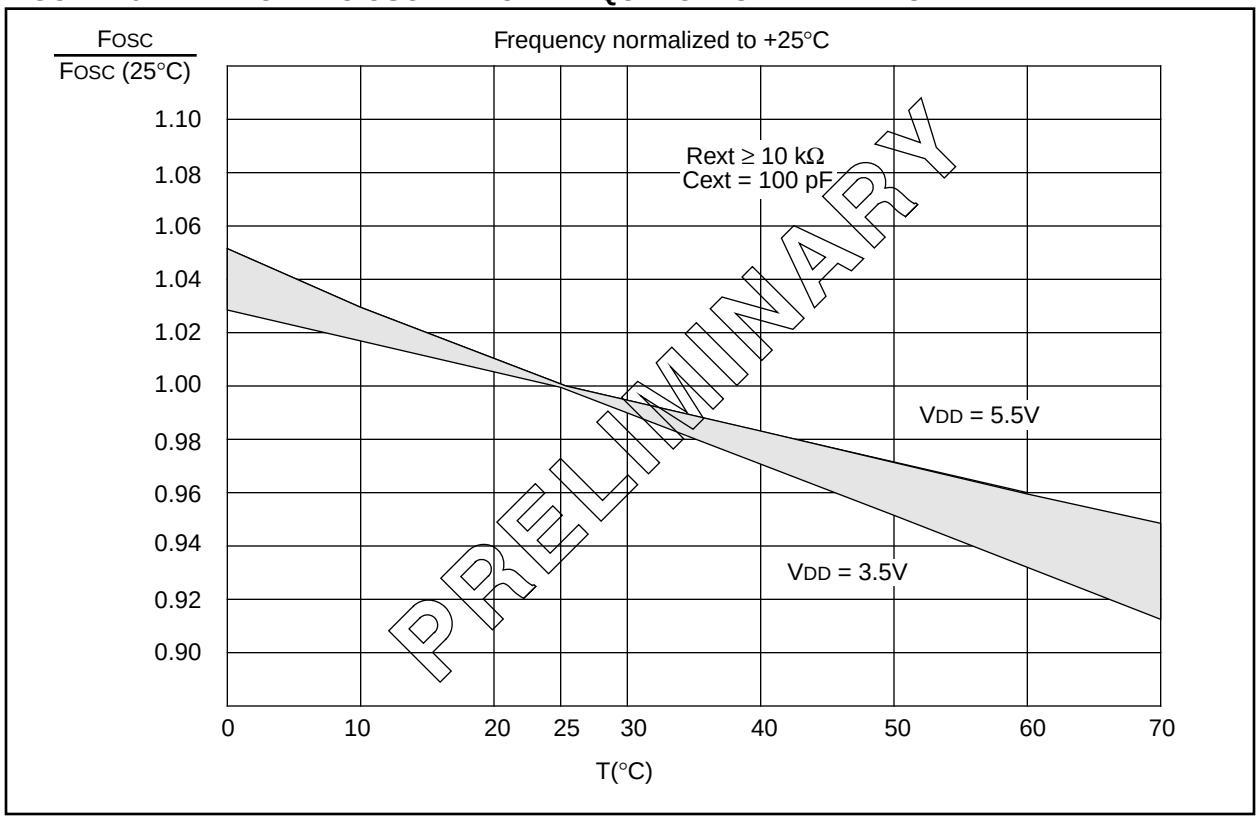
The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are ensured to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

TABLE 20-1: PIN CAPACITANCE PER PACKAGE TYPE

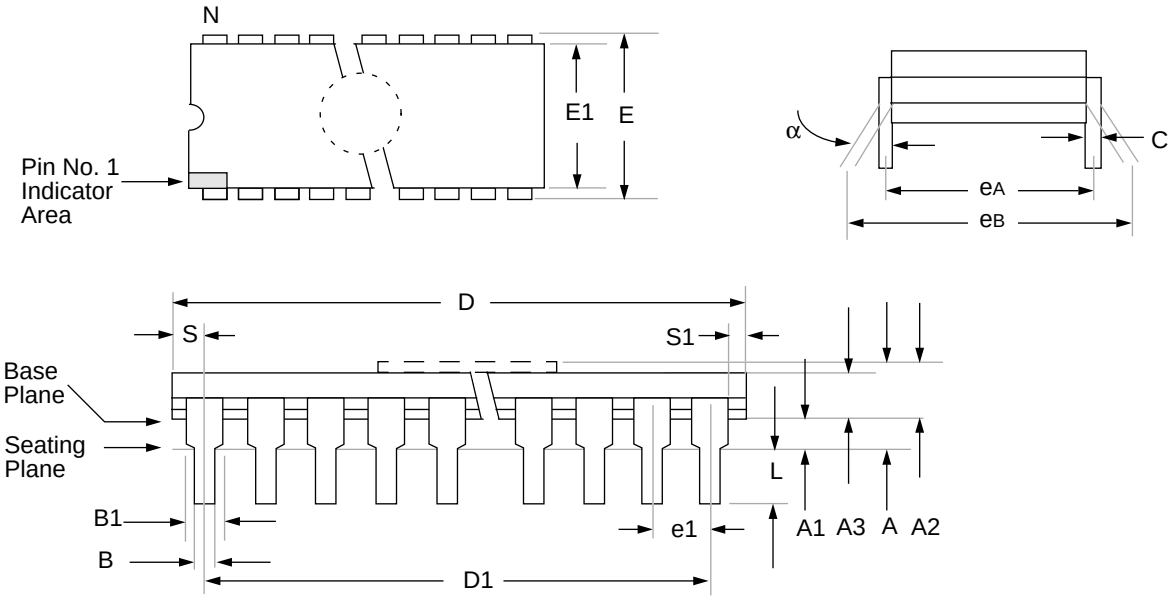
Pin Name	Typical Capacitance (pF)			
	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP
All pins, except $\overline{MCLR}$, VDD, and VSS	10	10	10	10
$\overline{MCLR}$ pin	20	20	20	20

FIGURE 20-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE



21.0 PACKAGING INFORMATION

21.1 40-Lead Ceramic CERDIP Dual In-line, and CERDIP Dual In-line with Window (600 mil)



Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	**Typical**	0.050	0.065	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	**Reference**	1.900	1.900	**Reference**
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	**Reference**	0.100	0.100	**Reference**
eA	14.986	16.002	**Typical**	0.590	0.630	**Typical**
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	40	40		40	40	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

LIST OF EQUATIONS