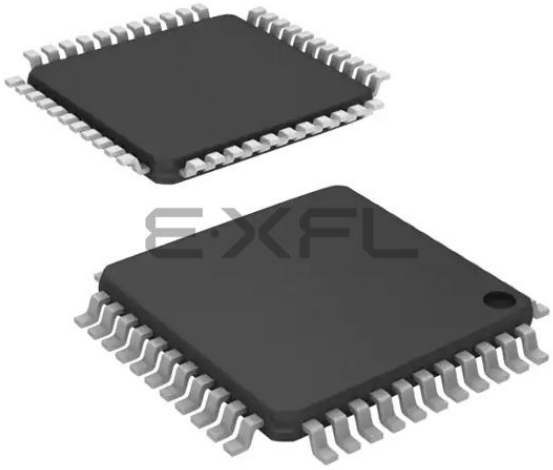


Welcome to E-XFL.COM

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	25MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°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/pic17c44-25i-pt

5.3 Peripheral Interrupt Request Register (PIR)

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

Note: These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral interrupt service routine.

FIGURE 5-4: PIR REGISTER (ADDRESS: 16h, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R - 1	R - 0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7							bit0

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

bit 7: **RBIF**: PORTB Interrupt on Change Flag bit
1 = One of the PORTB inputs changed (Software must end the mismatch condition)
0 = None of the PORTB inputs have changed

bit 6: **TMR3IF**: Timer3 Interrupt Flag bit
If Capture1 is enabled (CA1/$\overline{PR3}$ = 1)
1 = Timer3 overflowed
0 = Timer3 did not overflow

If Capture1 is disabled (CA1/$\overline{PR3}$ = 0)
1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value
0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value

bit 5: **TMR2IF**: Timer2 Interrupt Flag bit
1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value
0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value

bit 4: **TMR1IF**: Timer1 Interrupt Flag bit
If Timer1 is in 8-bit mode (T16 = 0)
1 = Timer1 value has rolled over to 0000h from equalling the period register (PR) value
0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value

If Timer1 is in 16-bit mode (T16 = 1)
1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value
0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value

bit 3: **CA2IF**: Capture2 Interrupt Flag bit
1 = Capture event occurred on RB1/CAP2 pin
0 = Capture event did not occur on RB1/CAP2 pin

bit 2: **CA1IF**: Capture1 Interrupt Flag bit
1 = Capture event occurred on RB0/CAP1 pin
0 = Capture event did not occur on RB0/CAP1 pin

bit 1: **TXIF**: USART Transmit Interrupt Flag bit
1 = Transmit buffer is empty
0 = Transmit buffer is full

bit 0: **RCIF**: USART Receive Interrupt Flag bit
1 = Receive buffer is full
0 = Receive buffer is empty

5.5 RA0/INT Interrupt

The external interrupt on the RA0/INT pin is edge triggered. Either the rising edge, if INTEDG bit (T0STA<7>) is set, or the falling edge, if INTEDG bit is clear. When a valid edge appears on the RA0/INT pin, the INTF bit (INTSTA<4>) is set. This interrupt can be disabled by clearing the INTE control bit (INTSTA<0>). The INT interrupt can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

5.6 TMR0 Interrupt

An overflow (FFFFh $\rightarrow$ 0000h) in TMR0 will set the T0IF (INTSTA<5>) bit. The interrupt can be enabled/ disabled by setting/clearing the T0IE control bit (INTSTA<1>). For operation of the Timer0 module, see Section 11.0.

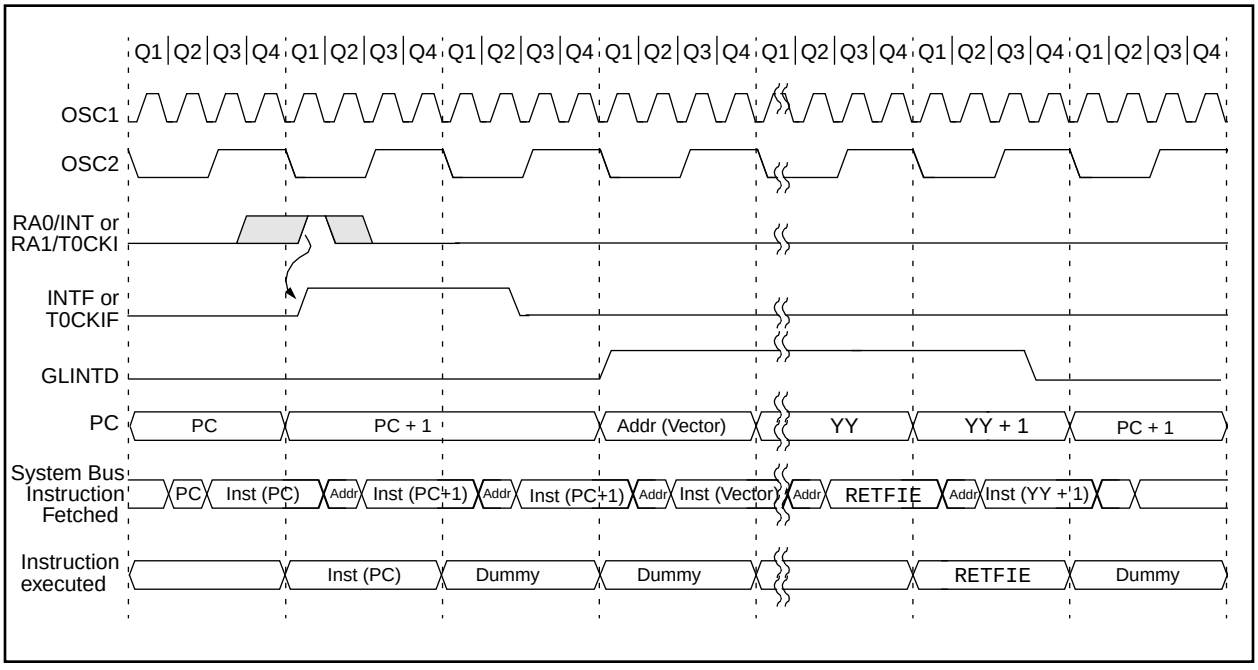
5.7 T0CKI Interrupt

The external interrupt on the RA1/T0CKI pin is edge triggered. Either the rising edge, if the T0SE bit (T0STA<6>) is set, or the falling edge, if the T0SE bit is clear. When a valid edge appears on the RA1/T0CKI pin, the T0CKIF bit (INTSTA<6>) is set. This interrupt can be disabled by clearing the T0CKIE control bit (INTSTA<2>). The T0CKI interrupt can wake up the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

5.8 Peripheral Interrupt

The peripheral interrupt flag indicates that at least one of the peripheral interrupts occurred (PEIF is set). The PEIF bit is a read only bit, and is a bit wise OR of all the flag bits in the PIR register AND'ed with the corresponding enable bits in the PIE register. Some of the peripheral interrupts can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

FIGURE 5-5: INT PIN / T0CKI PIN INTERRUPT TIMING



5.9 Context Saving During Interrupts

During an interrupt, only the returned PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt; e.g. WREG, ALUSTA and the BSR registers. This requires implementation in software.

Example 5-1 shows the saving and restoring of information for an interrupt service routine. The PUSH and POP routines could either be in each interrupt service routine or could be subroutines that were called. Depending on the application, other registers may also need to be saved, such as PCLATH.

EXAMPLE 5-1: SAVING STATUS AND WREG IN RAM

```
;
; The addresses that are used to store the CPUSTA and WREG values
; must be in the data memory address range of 18h - 1Fh. Up to
; 8 locations can be saved and restored using
; the MOVFP instruction. This instruction neither affects the status
; bits, nor corrupts the WREG register.
;
;
PUSH    MOVFP   WREG, TEMP_W         ; Save WREG
        MOVFP   ALUSTA, TEMP_STATUS  ; Save ALUSTA
        MOVFP   BSR, TEMP_BSR        ; Save BSR

ISR     :                            ; This is the interrupt service routine
        :
POP     MOVFP   TEMP_W, WREG         ; Restore WREG
        MOVFP   TEMP_STATUS, ALUSTA  ; Restore ALUSTA
        MOVFP   TEMP_BSR, BSR        ; Restore BSR
        RETFIE                       ; Return from Interrupts enabled
```

6.4.1 INDIRECT ADDRESSING REGISTERS

The PIC17C4X has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a NOP, and the status bits are not affected.

6.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVPF and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR.

A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

EXAMPLE 6-1: INDIRECT ADDRESSING

```
    MOVLW    0x20          ;
    MOVWF    FSR0          ; FSR0 = 20h
    BCF      ALUSTA, FS1   ; Increment FSR
    BSF      ALUSTA, FS0   ; after access
    BCF      ALUSTA, C     ; C = 0
    MOVLW    END_RAM + 1   ;
LP  CLRF     INDF0         ; Addr(FSR) = 0
    CPFSEQ   FSR0          ; FSR0 = END_RAM+1?
    GOTO     LP            ; NO, clear next
    :                      ; YES, All RAM is
    :                      ; cleared
```

6.5 Table Pointer (TBLPTRL and TBLPTRH)

File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

6.6 Table Latch (TBLATH, TBLATL)

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see descriptions of instructions TABLRD, TABLWT, TLRD and TLWT). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

6.8 Bank Select Register (BSR)

The BSR is used to switch between banks in the data memory area (Figure 6-13). In the PIC17C42, PIC17CR42, and PIC17C42A only the lower nibble is implemented. While in the PIC17C43, PIC17CR43, and PIC17C44 devices, the entire byte is implemented. The lower nibble is used to select the peripheral register bank. The upper nibble is used to select the general purpose memory bank.

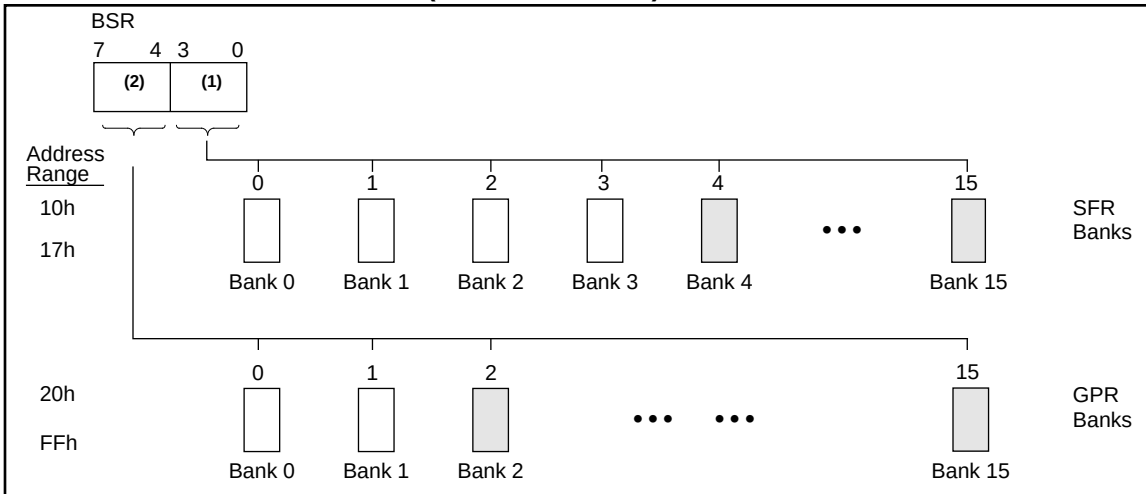
All the Special Function Registers (SFRs) are mapped into the data memory space. In order to accommodate the large number of registers, a banking scheme has been used. A segment of the SFRs, from address 10h to address 17h, is banked. The lower nibble of the bank select register (BSR) selects the currently active "peripheral bank." Effort has been made to group the peripheral registers of related functionality in one bank. However, it will still be necessary to switch from bank to bank in order to address all peripherals related to a single task. To assist this, a `MOVLB bank` instruction is in the instruction set.

For the PIC17C43, PIC17CR43, and PIC17C44 devices, the need for a large general purpose memory space dictated a general purpose RAM banking scheme. The upper nibble of the BSR selects the currently active general purpose RAM bank. To assist this, a `MOVLR bank` instruction has been provided in the instruction set.

If the currently selected bank is not implemented (such as Bank 13), any read will read all '0's. Any write is completed to the bit bucket and the ALU status bits will be set/cleared as appropriate.

Note:	Registers in Bank 15 in the Special Function Register area, are reserved for Microchip use. Reading of registers in this bank may cause random values to be read.

FIGURE 6-13: BSR OPERATION (PIC17C43/R43/44)



Note 1: Only Banks 0 through Bank 3 are implemented. Selection of an unimplemented bank is not recommended**.**
Bank 15 is reserved for Microchip use, reading of registers in this bank may cause random values to be read.
2: Only Banks 0 and Bank 1 are implemented. Selection of an unimplemented bank is not recommended.

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: $1 \rightarrow (f<b>)$

Status Affected: None

Encoding:

1000	0bbb	ffff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
BSF     FLAG_REG, 7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **Bit Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1001	1bbb	ffff	ffff

Description: If bit 'b' in register 'f' is 0 then the next instruction is skipped.
If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	NOP

If skip:

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

Example:
```
HERE    BTFSC   FLAG,1
FALSE   :
TRUE    :
```
Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (TRUE)
If FLAG<1> = 1;
PC = address (FALSE)

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

0000	0000	0000	0010

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register PCL*	Execute	NOP
Forced NOP	NOP	Execute	NOP

* Remember reading PCL causes PCLATH to be updated. This will be the high address of where the RETURN instruction is located.

Example: RETURN

After Interrupt
PC = TOS

RLCF **Rotate Left f through Carry**

Syntax: [*label*] RLCF f,d

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

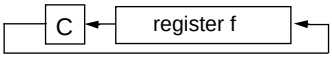
Operation: f<n> → d<n+1>;
f<7> → C;
C → d<0>

Status Affected: C

Encoding:

0001	101d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed 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:
```
RLCF    REG,0
```

Before Instruction
```
REG  =  1110 0110
C    =  0
```

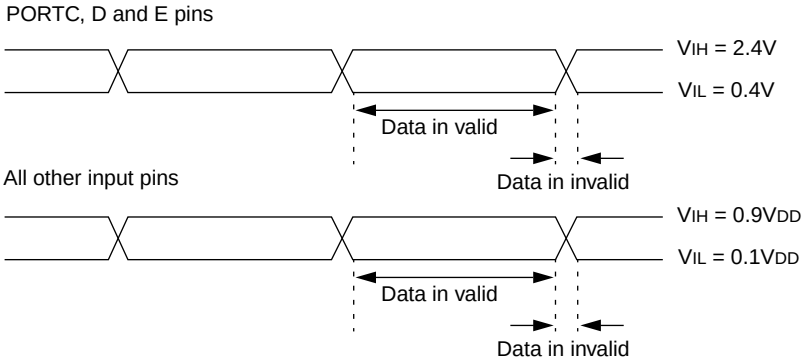
After Instruction
```
REG  =  1110 0110
WREG =  1100 1100
C    =  1
```

Applicable Devices	42	R42	42A	43	R43	44

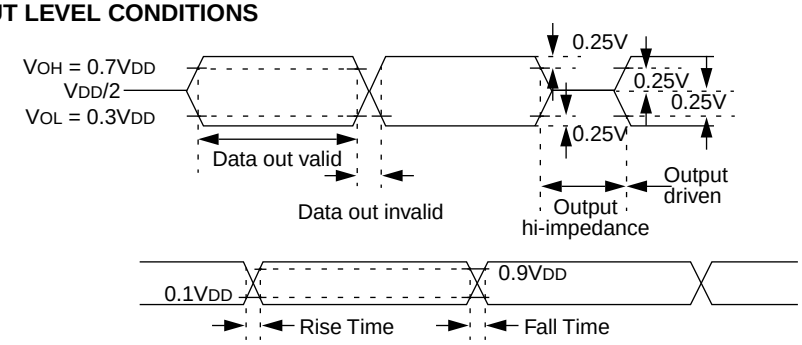
FIGURE 17-1: PARAMETER MEASUREMENT INFORMATION

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

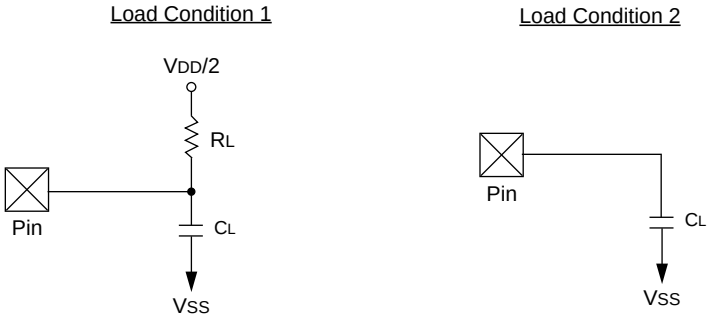
INPUT LEVEL CONDITIONS



OUTPUT LEVEL CONDITIONS



LOAD CONDITIONS



R_L = 464
$C_L \leq$ 50 pF

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

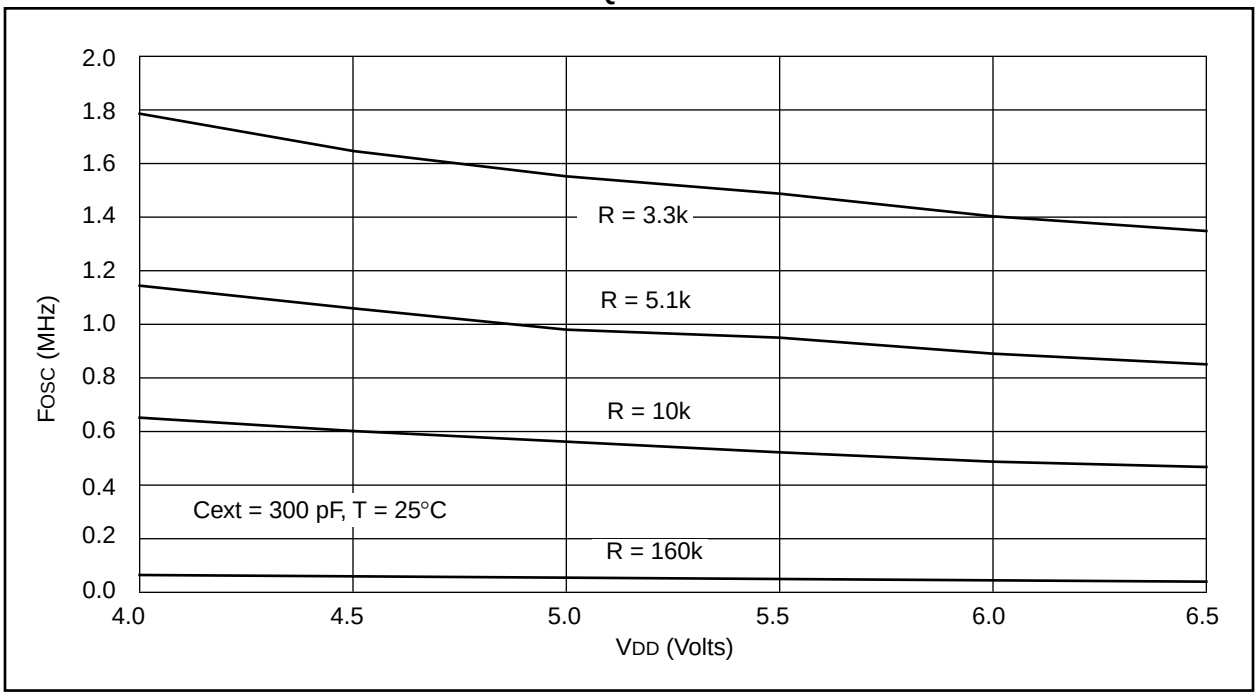


TABLE 18-2: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
22 pF	10k	3.33 MHz	± 12%
	100k	353 kHz	± 13%
100 pF	3.3k	3.54 MHz	± 10%
	5.1k	2.43 MHz	± 14%
	10k	1.30 MHz	± 17%
	100k	129 kHz	± 10%
300 pF	3.3k	1.54 MHz	± 14%
	5.1k	980 kHz	± 12%
	10k	564 kHz	± 16%
	160k	35 kHz	± 18%

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

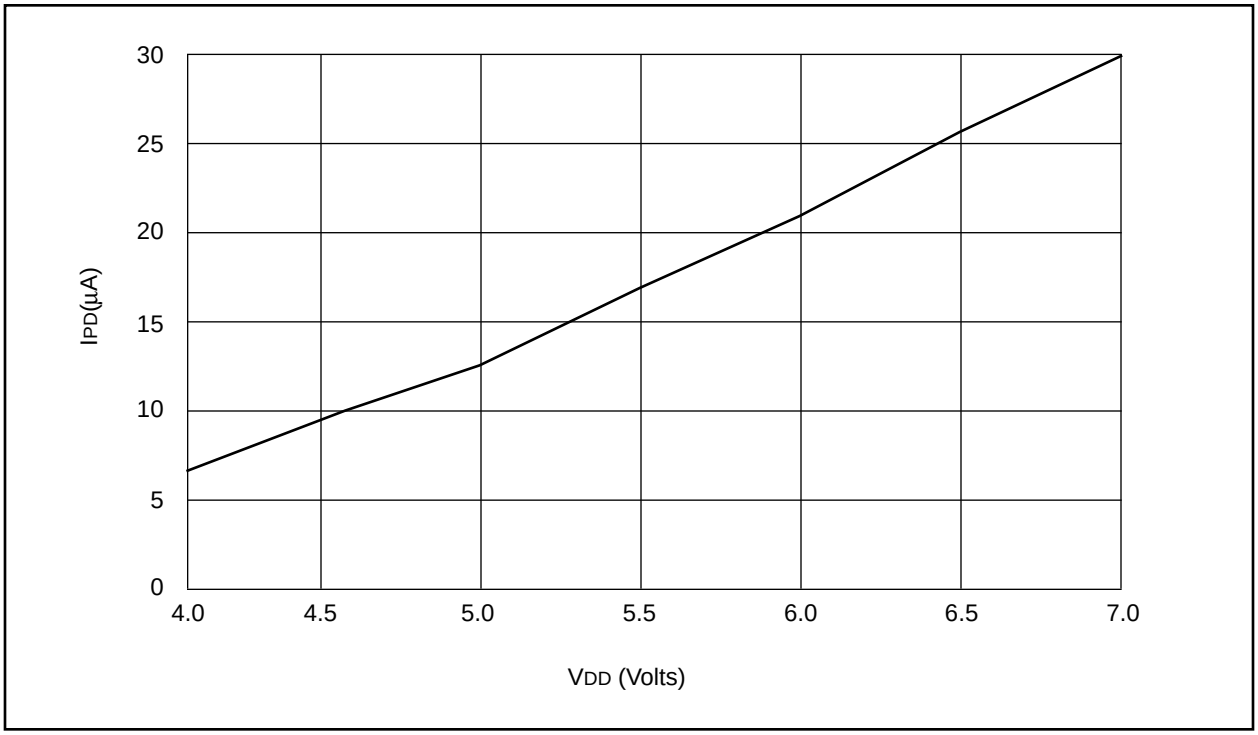
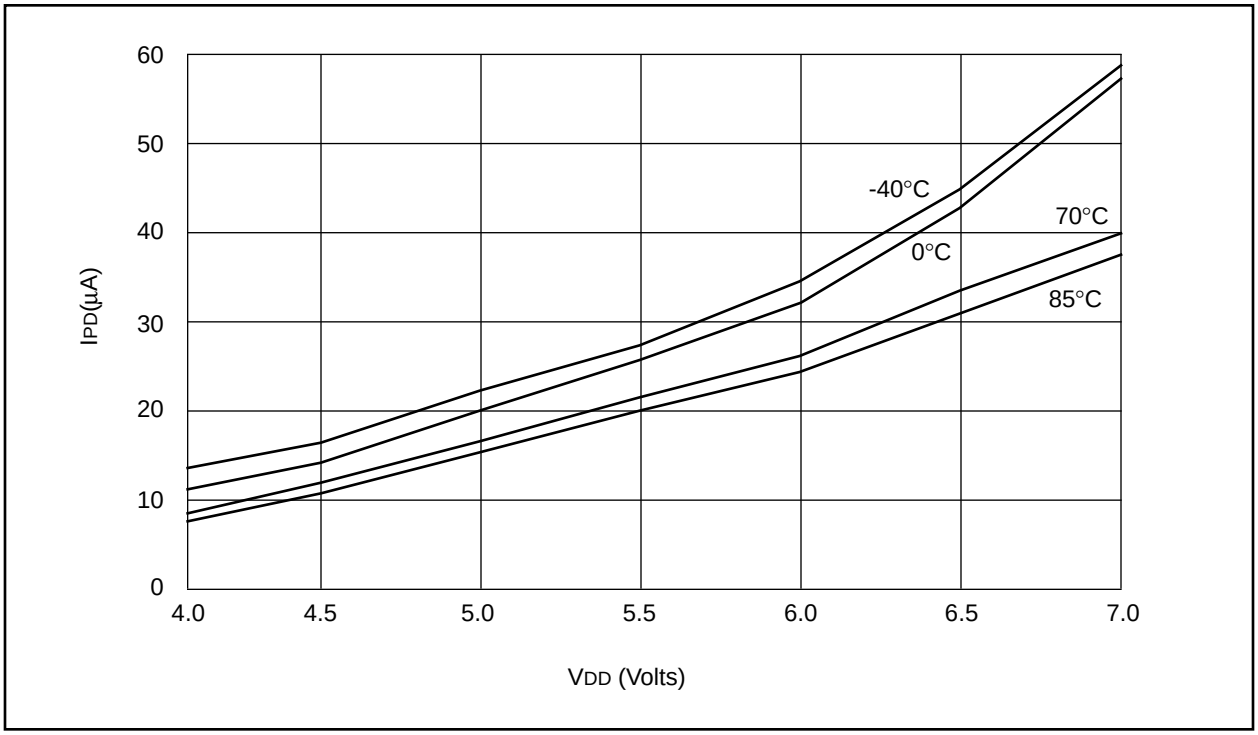


FIGURE 18-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, AND POWER-UP TIMER TIMING

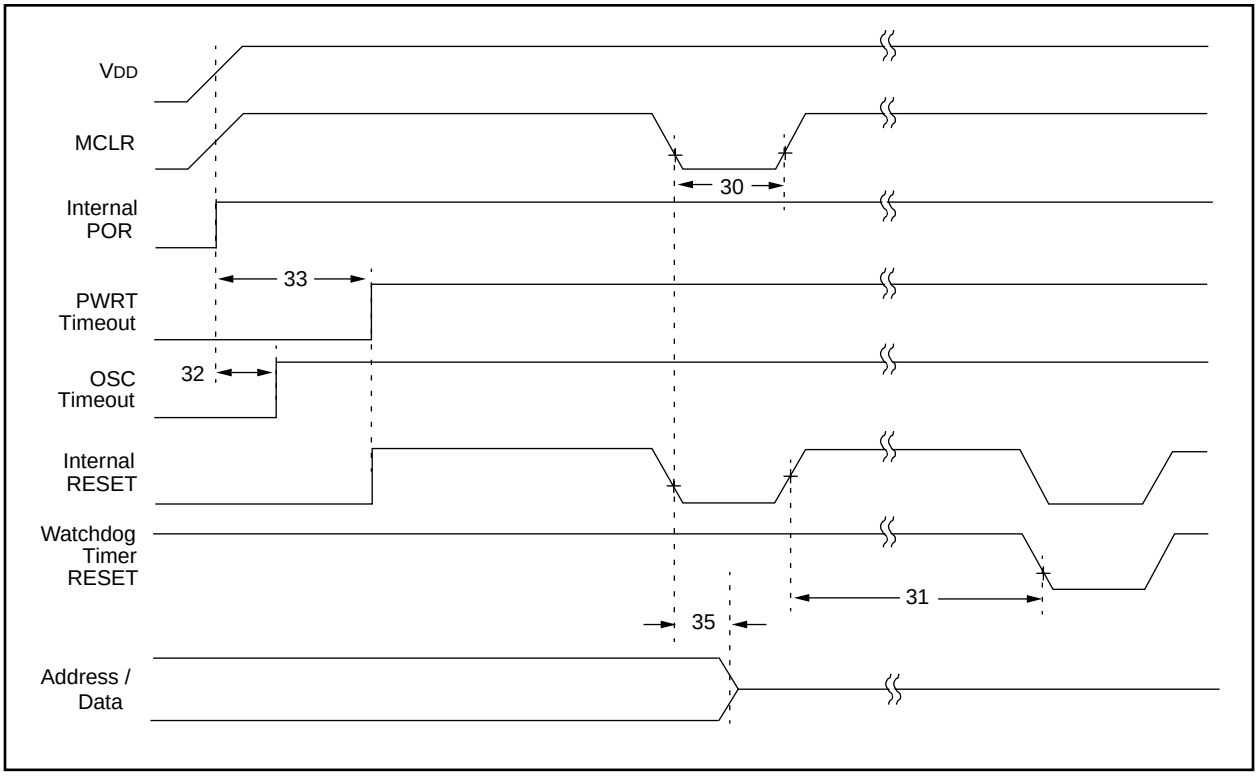


TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)		100 *	—	—	ns	VDD = 5V
31	Twdt	Watchdog Timer Time-out Period (Prescale = 1)		5 *	12	25 *	ms	VDD = 5V
32	Tost	Oscillation Start-up Timer Period		—	1024TOSC§	—	ms	TOSC = OSC1 period
33	Tpwrt	Power-up Timer Period		40 *	96	200 *	ms	VDD = 5V
35	TmcL2adl	$\overline{\text{MCLR}}$ to System Interface bus (AD15:AD0>) invalid	PIC17CR42/42A/43/R43/44	—	—	100 *	ns	
			PIC17LCR42/42A/43/R43/44	—	—	120 *	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.

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

§ This specification ensured by design.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 20-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

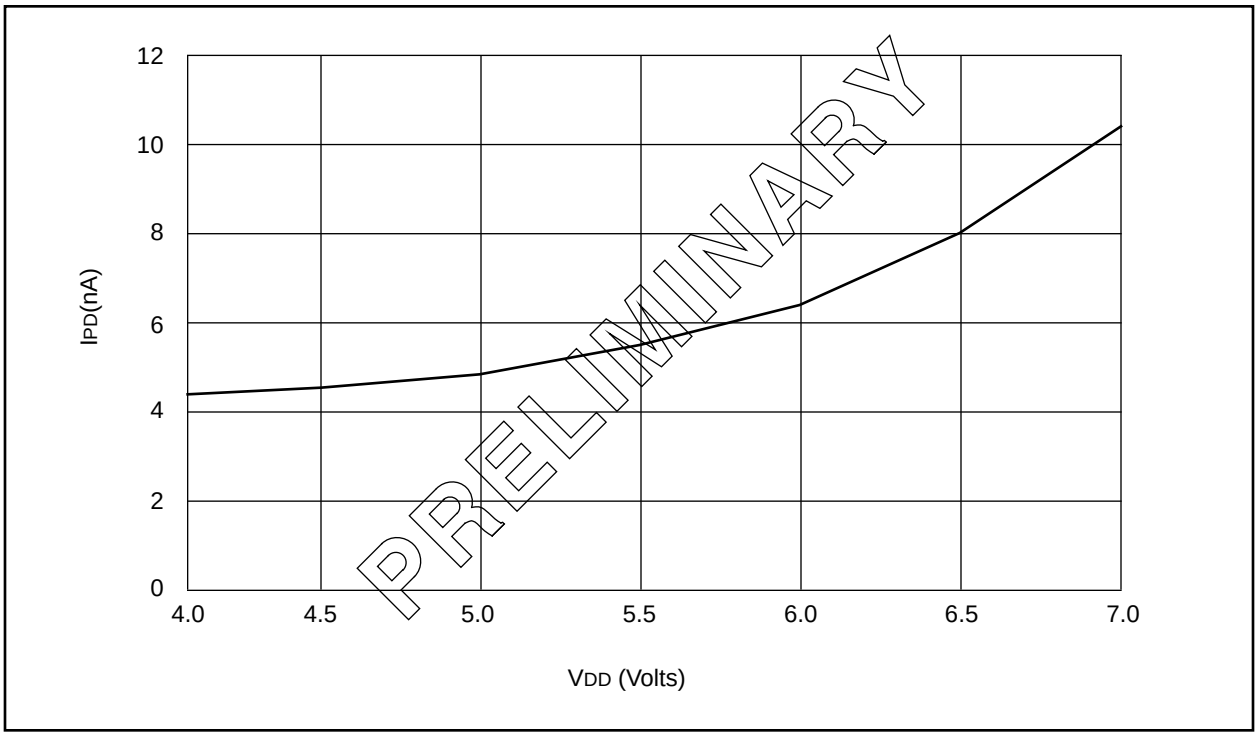
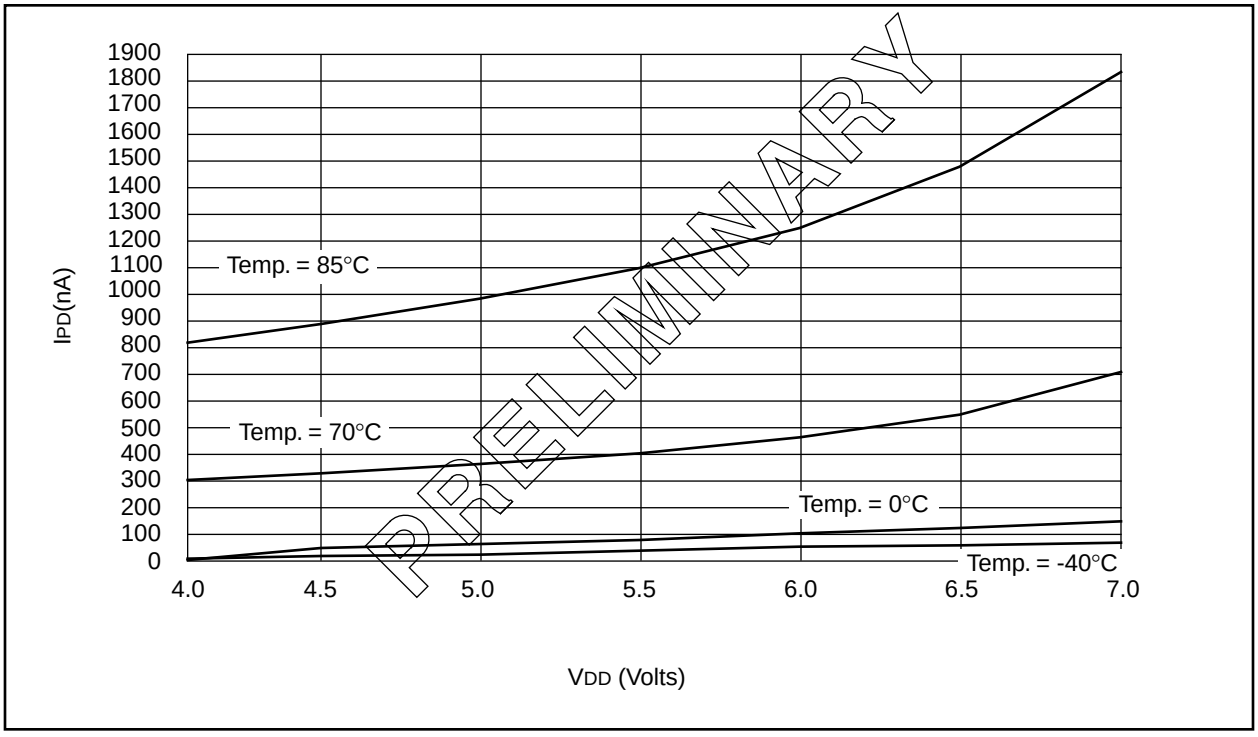


FIGURE 20-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 20-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

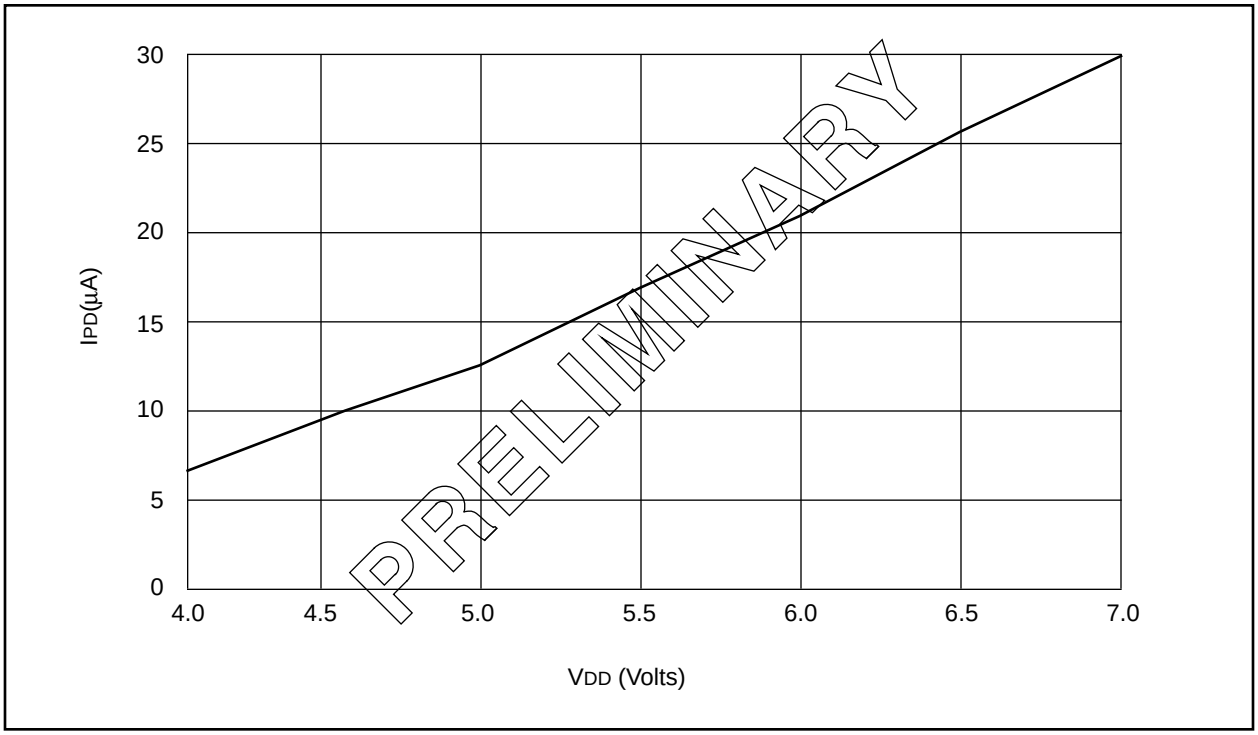
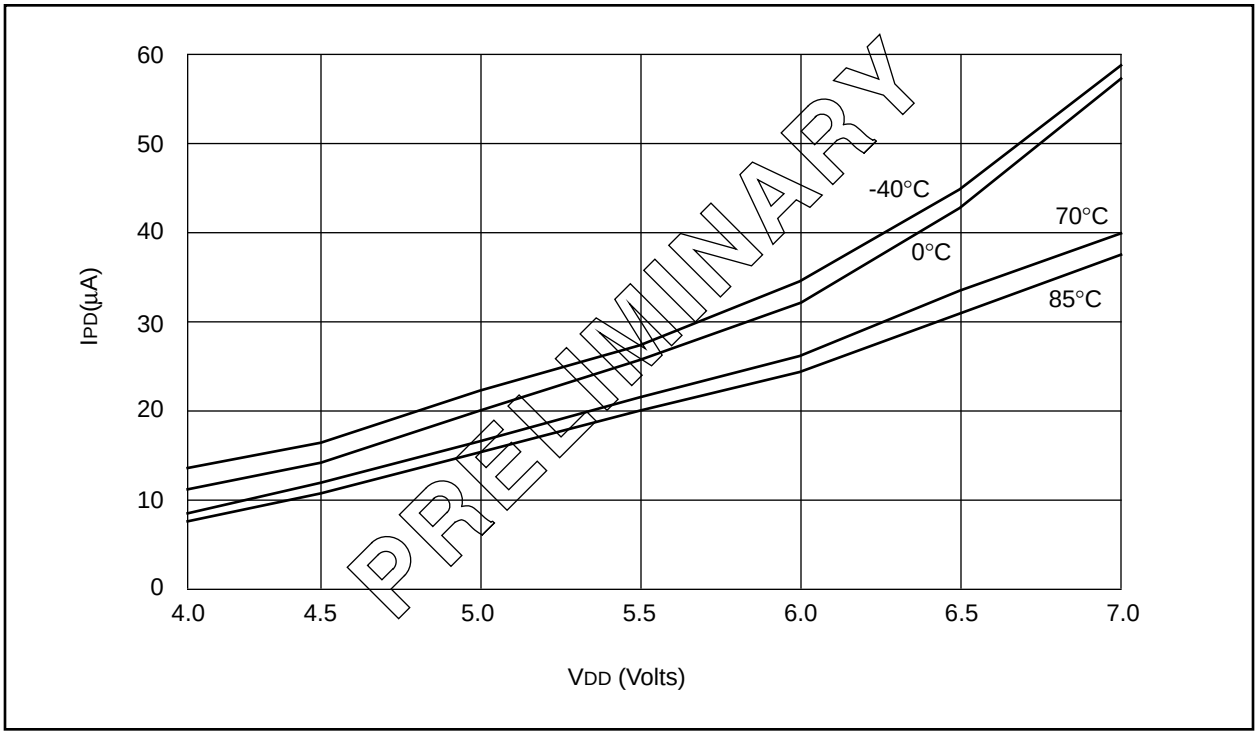


FIGURE 20-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 20-17: I_{OL} vs. V_{OL}, V_{DD} = 5V

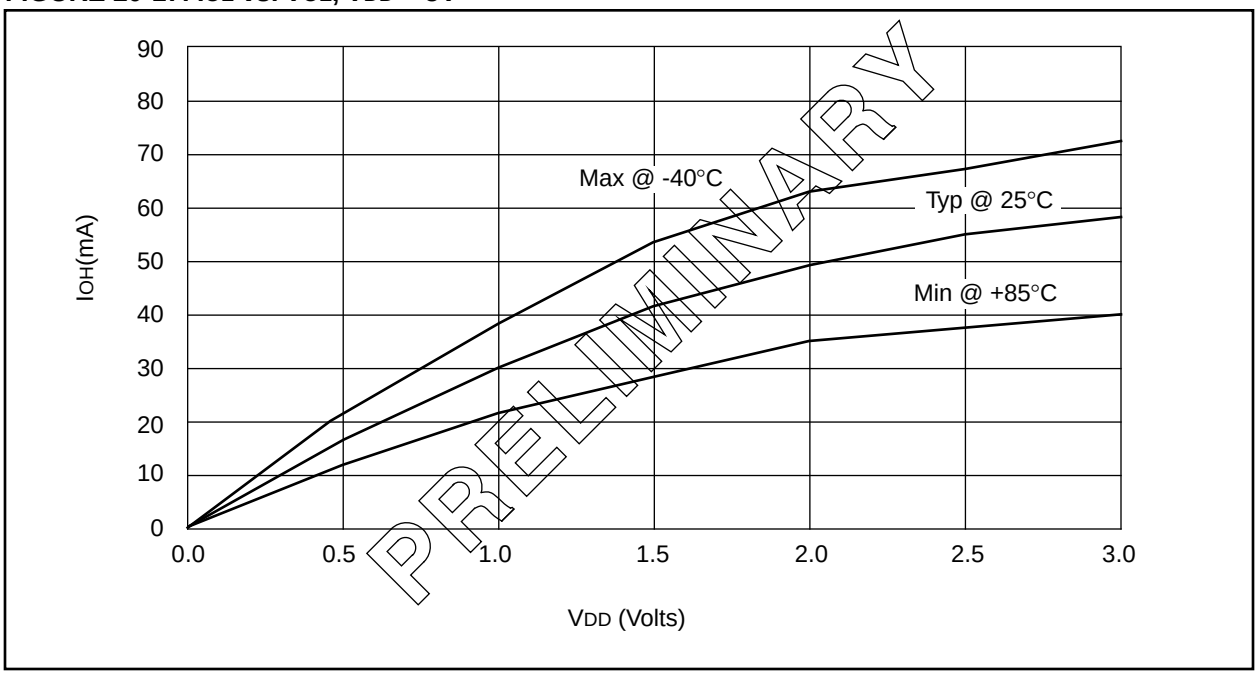
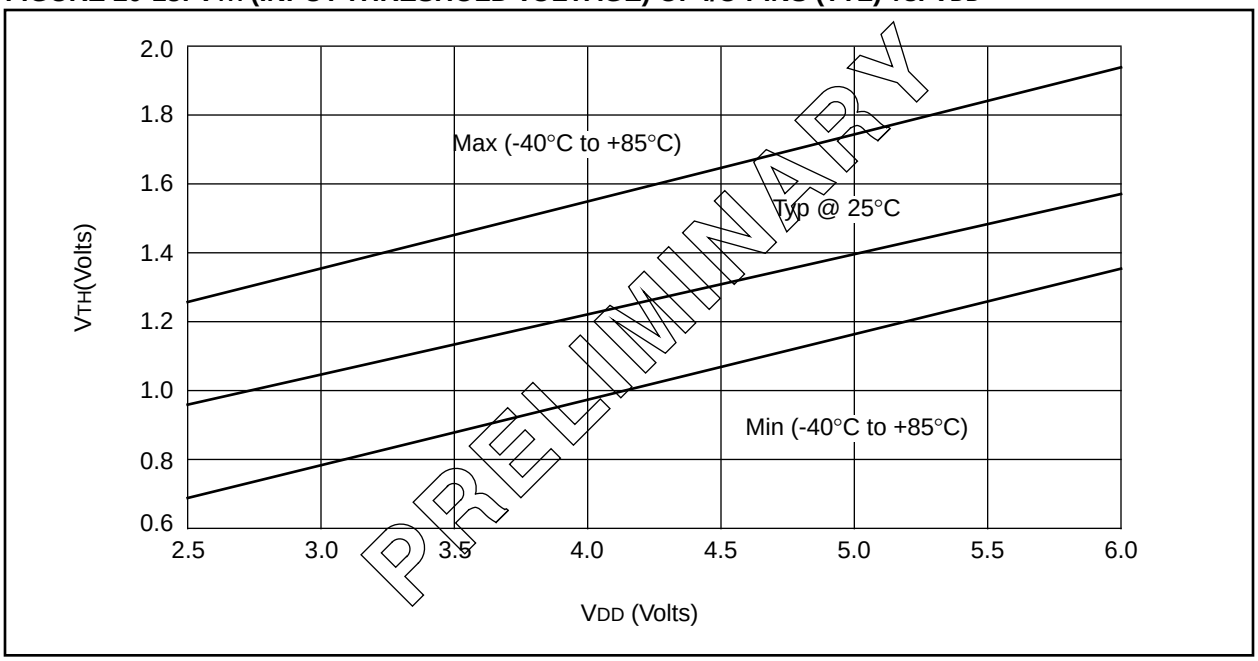
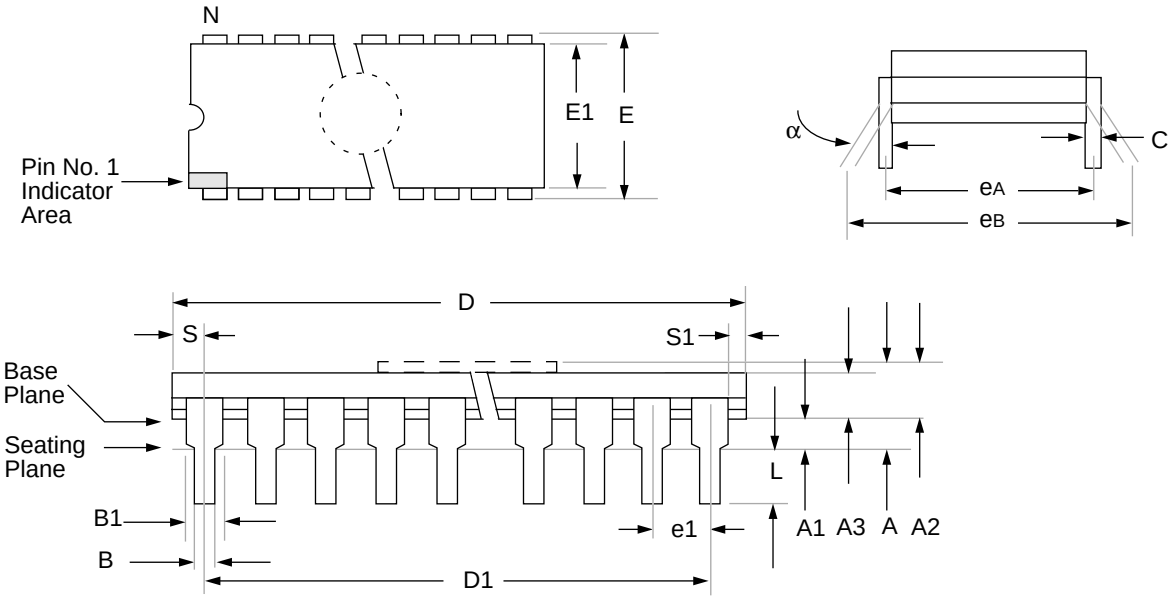


FIGURE 20-18: V_{TH} (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. V_{DD}



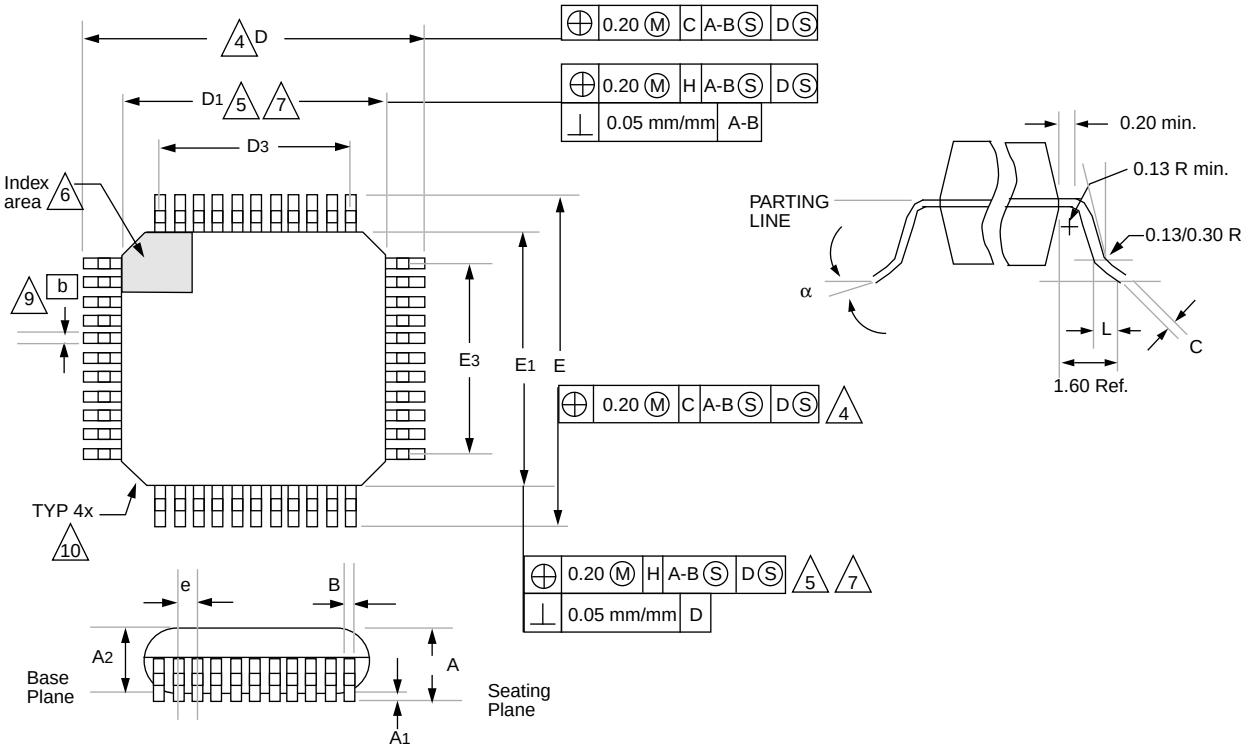
21.0 PACKAGING INFORMATION

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	

21.4 44-Lead Plastic Surface Mount (MQFP 10x10 mm Body 1.6/0.15 mm Lead Form)



Package Group: Plastic MQFP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	7°		0°	7°	
A	2.000	2.350		0.078	0.093	
A1	0.050	0.250		0.002	0.010	
A2	1.950	2.100		0.768	0.083	
b	0.300	0.450	**Typical**	0.011	0.018	**Typical**
C	0.150	0.180		0.006	0.007	
D	12.950	13.450		0.510	0.530	
D1	9.900	10.100		0.390	0.398	
D3	8.000	8.000	**Reference**	0.315	0.315	**Reference**
E	12.950	13.450		0.510	0.530	
E1	9.900	10.100		0.390	0.398	
E3	8.000	8.000	**Reference**	0.315	0.315	**Reference**
e	0.800	0.800		0.031	0.032	
L	0.730	1.030		0.028	0.041	
N	44	44		44	44	
CP	0.102	–		0.004	–	

APPENDIX C: WHAT'S NEW

The structure of the document has been made consistent with other data sheets. This ensures that important topics are covered across all PIC16/17 families. Here is an overview of new features.

Added the following devices:

PIC17CR42

PIC17C42A

PIC17CR43

A 33 MHz option is now available.

APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This allows control bits that have the same function, to have the same name (regardless of processor family). Care must still be taken, since they may not be at the same special function register address. The following shows the register and bit names that have been changed:

Old Name	New Name
TX8/9	TX9
RC8/9	RX9
RCD8	RX9D
TXD8	TX9D

Instruction DECFSNZ corrected to DCFSNZ

Instruction INCFSNZ corrected to INFSNZ

Enhanced discussion on PWM to include equation for determining bits of PWM resolution.

Section 13.2.2 and 13.3.2 have had the description of updating the FERR and RX9 bits enhanced.

The location of configuration bit PM2 was changed (Figure 6-1 and Figure 14-1).

Enhanced description of the operation of the INTSTA register.

Added note to discussion of interrupt operation.

Tightened electrical spec D110.

Corrected steps for setting up USART Asynchronous Reception.

APPENDIX F: ERRATA FOR PIC17C42 SILICON

The PIC17C42 devices that you have received have the following anomalies. At present there is no intention for future revisions to the present PIC17C42 silicon. If these cause issues for the application, it is recommended that you select the PIC17C42A device.

Note:	New designs should use the PIC17C42A.

1. When the Oscillator Start-Up Timer (OST) is enabled (in LF or XT oscillator modes), any interrupt that wakes the processor may cause a WDT reset. This occurs when the WDT is greater than or equal to 50% time-out period when the SLEEP instruction is executed. This will not occur in either the EC or RC oscillator modes.

   Work-arounds

a) Always ensure that the CLRWDT instruction is executed before the WDT increments past 50% of the WDT period. This will keep the “false” WDT reset from occurring.
b) When using the WDT as a normal timer (WDT disabled), ensure that the WDT is less than or equal to 50% time-out period when the SLEEP instruction is executed. This can be done by monitoring the $\overline{TO}$ bit for changing state from set to clear. Example 1 shows putting the PIC17C42 to sleep.

EXAMPLE F-1: PIC17C42 TO SLEEP

```
      BTFSS   CPUSTA, TO  ; TO = 0?
      CLRWDT              ; YES, WDT = 0
LOOP  BTFSC   CPUSTA, TO  ; WDT rollover?
      GOTO    LOOP        ; NO, Wait
      SLEEP               ; YES, goto Sleep
```

2. When the clock source of Timer1 or Timer2 is selected to external clock, the overflow interrupt flag will be set twice, once when the timer equals the period, and again when the timer value is reset to 0h. If the latency to clear TMRxIF is greater than the time to the next clock pulse, no problems will be noticed. If the latency is less than the time to the next timer clock pulse, the interrupt will be serviced twice.

   Work-arounds

a) Ensure that the timer has rolled over to 0h before clearing the flag bit.
b) Clear the timer in software. Clearing the timer in software causes the period to be one count less than expected.

Design considerations

The device must not be operated outside of the specified voltage range. An external reset circuit must be used to ensure the device is in reset when a brown-out occurs or the V_{DD} rise time is too long. Failure to ensure that the device is in reset when device voltage is out of specification may cause the device to lock-up and ignore the $\overline{MCLR}$ pin.

S

T

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.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — ISO 9001 / QS-9000 REGISTERED FIRM

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.