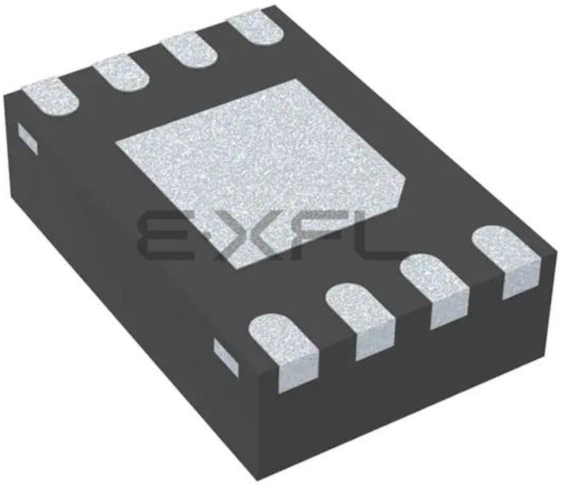


Welcome to E-XFL.COM

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	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	3
Program Memory Size	384B (256 x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VFDFN Exposed Pad
Supplier Device Package	8-DFN (2x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic10f200-e-mc

Pin Diagrams

FIGURE 1: 6-PIN SOT-23

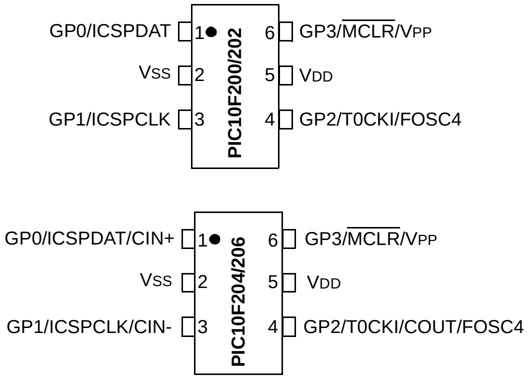


FIGURE 2: 8-PIN PDIP

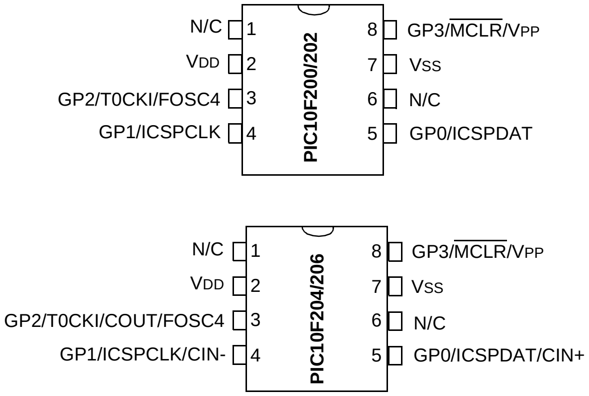


FIGURE 3: 8-PIN DFN

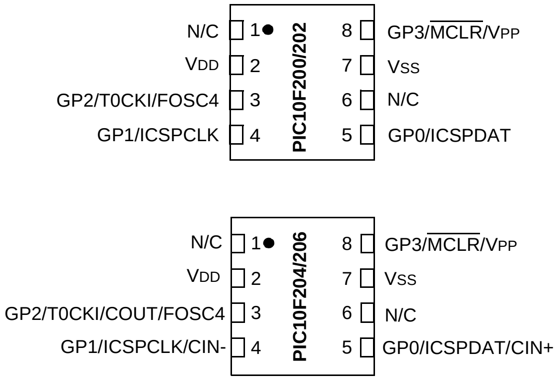


Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

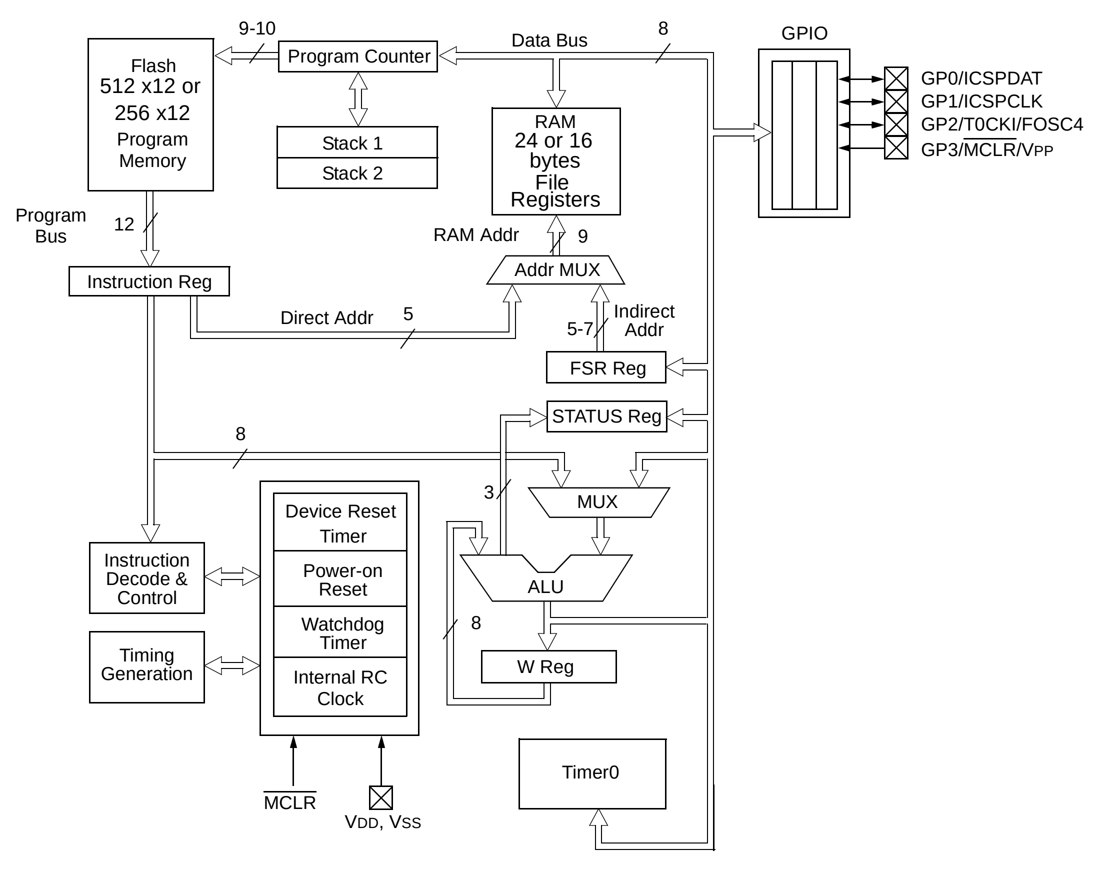
- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

FIGURE 3-1: PIC10F200/202 BLOCK DIAGRAM



3.1 Clocking Scheme/Instruction Cycle

The clock is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the PC is incremented every Q1 and the instruction is fetched from program memory and latched into the instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-3 and Example 3-1.

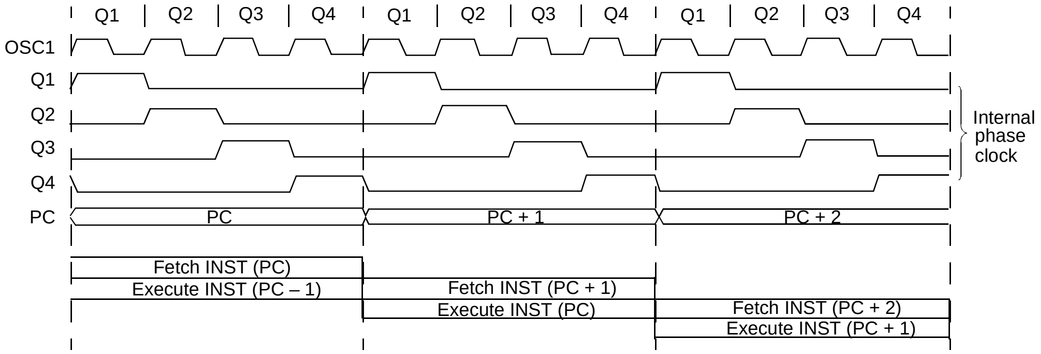
3.2 Instruction Flow/Pipelining

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 take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the PC to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the PC 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 3-1: INSTRUCTION PIPELINE FLOW

1. `MOVLW 03H`	Fetch 1	Execute 1				
2. `MOVWF GPIO`		Fetch 2	Execute 2			
3. `CALL SUB_1`			Fetch 3	Execute 3		
4. `BSF GPIO, BIT1`				Fetch 4	Flush	
					Fetch SUB_1	Execute SUB_1

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.

4.9 Indirect Data Addressing: INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

4.10 Indirect Addressing

- Register file 09 contains the value 10h
- Register file 0A contains the value 0Ah
- Load the value 09 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 0A)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = `0`) will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

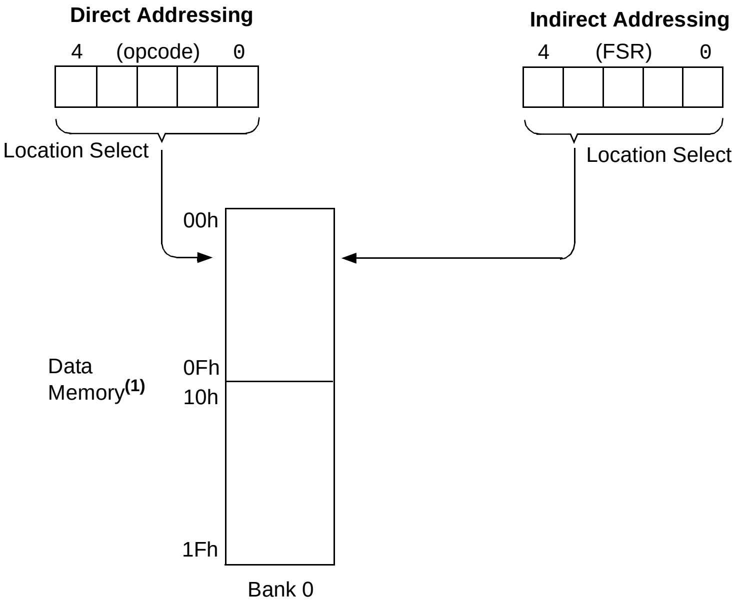
```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
                :       ;YES, continue
                :
```

The FSR is a 5-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

> **Note:** PIC10F200/202/204/206 – Do not use banking. FSR <7:5> are unimplemented and read as ‘1’s.

FIGURE 4-6: DIRECT/INDIRECT ADDRESSING (PIC10F200/202/204/206)



Note 1: For register map detail, see **Section 4.3 “Data Memory Organization”**.

TABLE 5-2: SUMMARY OF PORT 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
N/A	TRISGPIO	—	—	—	—	I/O Control Register				`---- 1111`	`---- 1111`
N/A	OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	`1111 1111`	`1111 1111`
03h	STATUS	GPWUF	CWUF	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`00-1 1xxx`	`qq-q quuu`$^{(1), (2)}$
06h	GPIO	—	—	—	—	GP3	GP2	GP1	GP0	`---- xxxx`	`---- uuuu`

Legend: Shaded cells are not used by PORT registers, read as ‘`0`’, – = unimplemented, read as ‘`0`’, `x` = unknown, `u` = unchanged, `q` = depends on condition.

Note 1: If Reset was due to wake-up on pin change, then bit 7 = `1`. All other Resets will cause bit 7 = `0`.

2: If Reset was due to wake-up on comparator change, then bit 6 = `1`. All other Resets will cause bit 6 = `0`.

5.4 I/O Programming Considerations

5.4.1 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The `BCF` and `BSF` instructions, for example, read the entire port into the CPU, execute the bit operation and rewrite the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a `BSF` operation on bit 2 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit 2 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bidirectional I/O pin (say bit 0), and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential Read-Modify-Write instructions (e.g., `BCF, BSF`, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin (“wired OR”, “wired AND”). The resulting high output currents may damage the chip.

EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;                 GPIO latch   GPIO pins
;                 ----------   ----------
 BCF    GPIO,  1 ;---- pp01    ---- pp11
 BCF    GPIO,  0 ;---- pp10    ---- pp11
 MOVLW  007h;
 TRIS   GPIO     ;---- pp10    ---- pp11
;
```

Note 1: The user may have expected the pin values to be `---- pp00`. The 2nd `BCF` caused GP1 to be latched as the pin value (High).

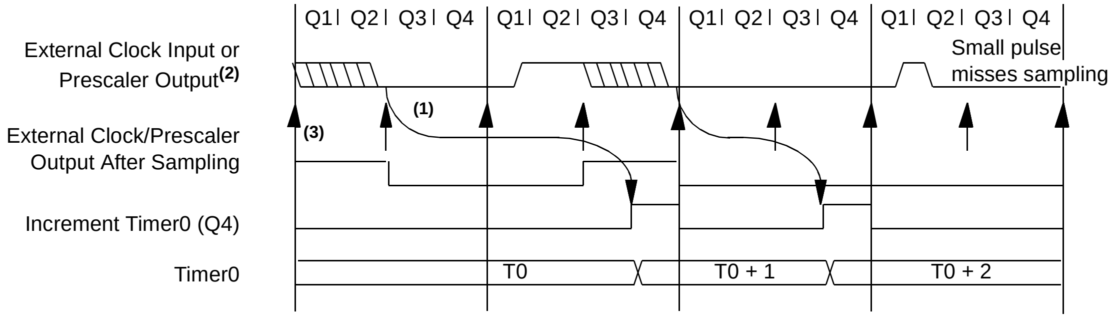
5.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a `NOP` or another instruction not accessing this I/O port.

6.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 6-4 shows the delay from the external clock edge to the timer incrementing.

FIGURE 6-4: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to Timer0 increment is 3 TOSC to 7 TOSC (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4 TOSC max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the points in time where sampling occurs.

6.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (see **Section 9.6 “Watchdog Timer (WDT)”**). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet.

Note: The prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x`, etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a Reset, the prescaler contains all ‘`0`’s.

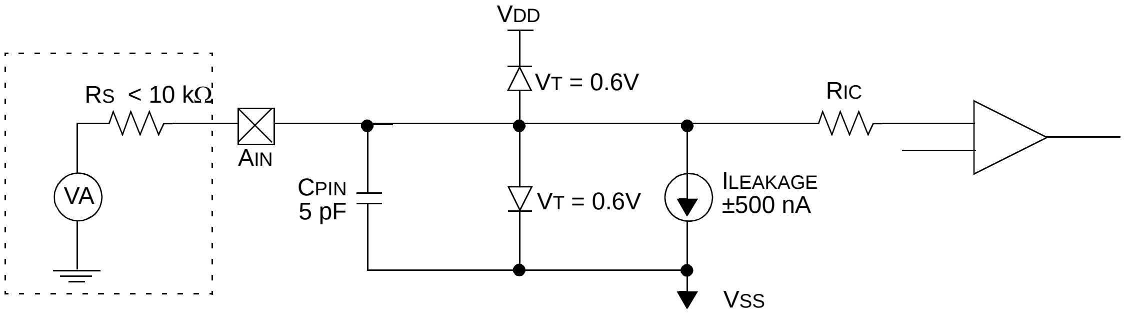
6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0 → WDT)

```
CLRWDT              ;Clear WDT
CLRF    TMR0        ;Clear TMR0 & Prescaler
MOVLW   ‘00xx1111’b ;These 3 lines (5, 6, 7)
OPTION              ;are required only if
                    ;desired
CLRWDT              ;PS<2:0> are 000 or 001
MOVLW   ‘00xx1xxx’b ;Set Postscaler to
OPTION              ;desired WDT rate
```

FIGURE 8-3: ANALOG INPUT MODE



Legend:
C_{PIN} = Input Capacitance
V_T = Threshold Voltage
$I_{LEAKAGE}$ = Leakage Current at the Pin
R_{IC} = Interconnect Resistance
R_S = Source Impedance
VA = Analog Voltage

TABLE 8-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other Resets
03h	STATUS	GPWUF	CWUF	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`00-1 1xxx`	`qq0q quuu`
07h	CMCON0	CMPOUT	$\overline{\text{COUTEN}}$	POL	$\overline{\text{CMPT0CS}}$	CMPON	CNREF	CPREF	$\overline{\text{CWU}}$	`1111 1111`	`uuuu uuuu`
N/A	TRISGPIO	—	—	—	—	I/O Control Register				`---- 1111`	`---- 1111`

Legend: `x` = Unknown, `u` = Unchanged, – = Unimplemented, read as ‘`0`’, `q` = Depends on condition.

9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

The PIC10F200/202/204/206 devices are offered with Internal Oscillator mode only.

- INTOSC: Internal 4 MHz Oscillator

9.2.2 INTERNAL 4 MHz OSCILLATOR

The internal oscillator provides a 4 MHz (nominal) system clock (see **Section 12.0 “Electrical Characteristics”** for information on variation over voltage and temperature).

In addition, a calibration instruction is programmed into the last address of memory, which contains the calibration value for the internal oscillator. This location is always uncode protected, regardless of the code-protect settings. This value is programmed as a `MOVLW xx` instruction where `xx` is the calibration value and is placed at the Reset vector. This will load the W register with the calibration value upon Reset and the PC will then roll over to the users program at address 0x000. The user then has the option of writing the value to the OSCCAL Register (05h) or ignoring it.

OSCCAL, when written to with the calibration value, will “trim” the internal oscillator to remove process variation from the oscillator frequency.

> **Note:** Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part so it can be reprogrammed correctly later.

9.3 Reset

The device differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT time-out Reset during normal operation
- WDT time-out Reset during Sleep
- Wake-up from Sleep on pin change
- Wake-up from Sleep on comparator change

Some registers are not reset in any way, they are unknown on POR and unchanged in any other Reset. Most other registers are reset to “Reset state” on Power-on Reset (POR), $\overline{\text{MCLR}}$, WDT or Wake-up on pin change Reset during normal operation. They are not affected by a WDT Reset during Sleep or $\overline{\text{MCLR}}$ Reset during Sleep, since these Resets are viewed as resumption of normal operation. The exceptions to this are $\overline{\text{TO}}$, $\overline{\text{PD}}$, GPWUF and CWUF bits. They are set or cleared differently in different Reset situations. These bits are used in software to determine the nature of Reset. See Table 9-1 for a full description of Reset states of all registers.

TABLE 9-1: RESET CONDITIONS FOR REGISTERS – PIC10F200/202/204/206

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset, WDT Time-out, Wake-up On Pin Change, Wake on Comparator Change
W	—	qqqq qqqu(1)	qqqq qqqu(1)
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	00-1 1xxx	q00q quuu(2)
STATUS(3)	03h	00-1 1xxx	qq0q quuu(2)
FSR	04h	111x xxxx	111u uuuu
OSCCAL	05h	1111 1110	uuuu uuuu
GPIO	06h	---- xxxx	---- uuuu
CMCON(3)	07h	1111 1111	uuuu uuuu
OPTION	—	1111 1111	1111 1111
TRISGPIO	—	---- 1111	---- 1111

Legend: u = unchanged, x = unknown, – = unimplemented bit, read as ‘0’, q = value depends on condition.
Note 1: Bits <7:2> of W register contain oscillator calibration values due to `MOVLW XX` instruction at top of memory.
2: See Table 9-2 for Reset value for specific conditions.
3: PIC10F204/206 only.

TABLE 10-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode	Status Affected	Notes
				MSb LSb		
ADDWF	f, d	Add W and f	1	0001 11df ffff	C, DC, Z	**1, 2, 4**
ANDWF	f, d	AND W with f	1	0001 01df ffff	Z	**2, 4**
CLRF	f	Clear f	1	0000 011f ffff	Z	**4**
CLRW	—	Clear W	1	0000 0100 0000	Z	
COMF	f, d	Complement f	1	0010 01df ffff	Z	
DECF	f, d	Decrement f	1	0000 11df ffff	Z	**2, 4**
DECFSZ	f, d	Decrement f, Skip if 0	$1^{(2)}$	0010 11df ffff	None	**2, 4**
INCF	f, d	Increment f	1	0010 10df ffff	Z	**2, 4**
INCFSZ	f, d	Increment f, Skip if 0	$1^{(2)}$	0011 11df ffff	None	**2, 4**
IORWF	f, d	Inclusive OR W with f	1	0001 00df ffff	Z	**2, 4**
MOVF	f, d	Move f	1	0010 00df ffff	Z	**2, 4**
MOVWF	f	Move W to f	1	0000 001f ffff	None	**1, 4**
NOP	—	No Operation	1	0000 0000 0000	None	
RLF	f, d	Rotate left f through Carry	1	0011 01df ffff	C	**2, 4**
RRF	f, d	Rotate right f through Carry	1	0011 00df ffff	C	**2, 4**
SUBWF	f, d	Subtract W from f	1	0000 10df ffff	C, DC, Z	**1, 2, 4**
SWAPF	f, d	Swap f	1	0011 10df ffff	None	**2, 4**
XORWF	f, d	Exclusive OR W with f	1	0001 10df ffff	Z	**2, 4**
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	0100 bbbf ffff	None	**2, 4**
BSF	f, b	Bit Set f	1	0101 bbbf ffff	None	**2, 4**
BTFSC	f, b	Bit Test f, Skip if Clear	$1^{(2)}$	0110 bbbf ffff	None	
BTFSS	f, b	Bit Test f, Skip if Set	$1^{(2)}$	0111 bbbf ffff	None	
LITERAL AND CONTROL OPERATIONS						
ANDLW	k	AND literal with W	1	1110 kkkk kkkk	Z	
CALL	k	Call Subroutine	2	1001 kkkk kkkk	None	**1**
CLRWDT		Clear Watchdog Timer	1	0000 0000 0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Unconditional branch	2	101k kkkk kkkk	None	
IORLW	k	Inclusive OR literal with W	1	1101 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	1100 kkkk kkkk	None	
OPTION	—	Load OPTION register	1	0000 0000 0010	None	
RETLW	k	Return, place Literal in W	2	1000 kkkk kkkk	None	
SLEEP	—	Go into Standby mode	1	0000 0000 0011	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register	1	0000 0000 0fff	None	**3**
XORLW	k	Exclusive OR literal to W	1	1111 kkkk kkkk	Z	

Note 1: The 9th bit of the program counter will be forced to a '`0`' by any instruction that writes to the PC except for `GOTO`. See **Section 4.7 "Program Counter"**.

2: When an I/O register is modified as a function of itself (e.g. `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '`1`' for a pin configured as input and is driven low by an external device, the data will be written back with a '`0`'.

3: The instruction `TRIS f`, where f = 6, causes the contents of the W register to be written to the tri-state latches of PORTB. A '`1`' forces the pin to a high-impedance state and disables the output buffers.

4: If this instruction is executed on the TMR0 register (and where applicable, d = `1`), the prescaler will be cleared (if assigned to TMR0).

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) – 1 → (dest)
Status Affected:	Z
Description:	Decrement register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) – 1 → d; skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are decremented. If ‘d’ is ‘0’, the result is placed in the W register. 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 a NOP is executed instead making it a 2-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 511
Operation:	k → PC<8:0>; STATUS<6:5> → PC<10:9>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a 2-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) + 1 → (dest)
Status Affected:	Z
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. If the result is ‘0’, then the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a 2-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. (k) → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the 8-bit literal ‘k’. The result is placed in the W register.

IORWF	**Inclusive OR W with f**
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W).OR. (f) → (dest)
Status Affected:	Z
Description:	Inclusive OR the W register with register ‘f’. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

MOVF	**Move f**
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register ‘f’ are moved to destination ‘d’. If ‘d’ is ‘0’, destination is the W register. If ‘d’ is ‘1’, the destination is file register ‘f’. ‘d’ = 1 is useful as a test of a file register, since status flag Z is affected.

MOVLW	**Move literal to W**
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Description:	The 8-bit literal ‘k’ is loaded into the W register. The “don’t cares” will assembled as ‘0’s.

MOVWF	**Move W to f**
Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 31
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from the W register to register ‘f’.

NOP	**No Operation**
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

OPTION	**Load OPTION Register**
Syntax:	[*label*] OPTION
Operands:	None
Operation:	(W) → Option
Status Affected:	None
Description:	The content of the W register is loaded into the OPTION register.

TABLE 12-1: COMPARATOR SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ T_A ≤ +125°C								
Param. No.	**Sym.**	**Characteristics**		**Min.**	**Typ.†**	**Max.**	**Units**	**Comments**
D300	V_{OS}	Input Offset Voltage		—	± 5.0	± 10	mV	(V_{DD} - 1.5)/2
D301	V_{CM}	Input Common Mode Voltage		0	—	V_{DD}–1.5*	V	
D302	CMRR	Common Mode Rejection Ratio		55*	—	—	dB	
D303*	T_{RT}	Response Time	Falling	—	150	600	ns	**(Note 1)**
			Rising	—	200	1000	ns	
D304*	T_{MC2COV}	Comparator Mode Change to Output Valid		—	—	10*	µs	
D305	V_{IVRF}	Internal Reference Voltage		0.55	0.6	0.65	V	2.0V ≤ V_{DD} ≤ 5.5V -40°C ≤ T_A ≤ ±125°C (extended)

* 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.

Note 1: Response time is measured with one comparator input at (V_{DD} - 1.5)/2 - 100 mV to (V_{DD} - 1.5)/2 + 20 mV.

TABLE 12-2: PULL-UP RESISTOR RANGES

V_{DD} (Volts)	**Temperature (°C)**	**Min.**	**Typ.**	**Max.**	**Units**
GP0/GP1					
2.0	-40	73K	105K	186K	Ω
	25	73K	113K	187K	Ω
	85	82K	123K	190K	Ω
	125	86K	132k	190K	Ω
5.5	-40	15K	21K	33K	Ω
	25	15K	22K	34K	Ω
	85	19K	26k	35K	Ω
	125	23K	29K	35K	Ω
GP3					
2.0	-40	63K	81K	96K	Ω
	25	77K	93K	116K	Ω
	85	82K	96k	116K	Ω
	125	86K	100K	119K	Ω
5.5	-40	16K	20k	22K	Ω
	25	16K	21K	23K	Ω
	85	24K	25k	28K	Ω
	125	26K	27K	29K	Ω

12.4 Timing Parameter Symbology and Load Conditions – PIC10F200/202/204/206

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

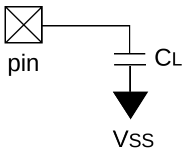
Lowercase subscripts (pp) and their meanings:

pp			
2	to	mc	$\overline{\text{MCLR}}$
ck	CLKOUT	osc	Oscillator
cy	Cycle time	t0	T0CKI
drt	Device Reset Timer	wdt	Watchdog Timer
io	I/O port	wdt	Watchdog Timer

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (high-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 12-2: LOAD CONDITIONS – PIC10F200/202/204/206



Legend:
C_L = 50 pF for all pins

TABLE 12-3: CALIBRATED INTERNAL RC FREQUENCIES – PIC10F200/202/204/206

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial), -40°C ≤ TA ≤ +125°C (extended) Operating Voltage VDD range is described in **Section 12.1 “DC Characteristics: PIC10F200/202/204/206 (Industrial)”**					
Param. No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
F10	FOSC	Internal Calibrated INTOSC Frequency**(1,2)**	± 1%	3.96	4.00	4.04	MHz	VDD=3.5V @ 25°C
			± 2%	3.92	4.00	4.08	MHz	2.5V ≤ VDD ≤ 5.5V 0°C ≤ TA ≤ +85°C (industrial)
			± 5%	3.80	4.00	4.20	MHz	2.0V ≤ VDD ≤ 5.5V -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)

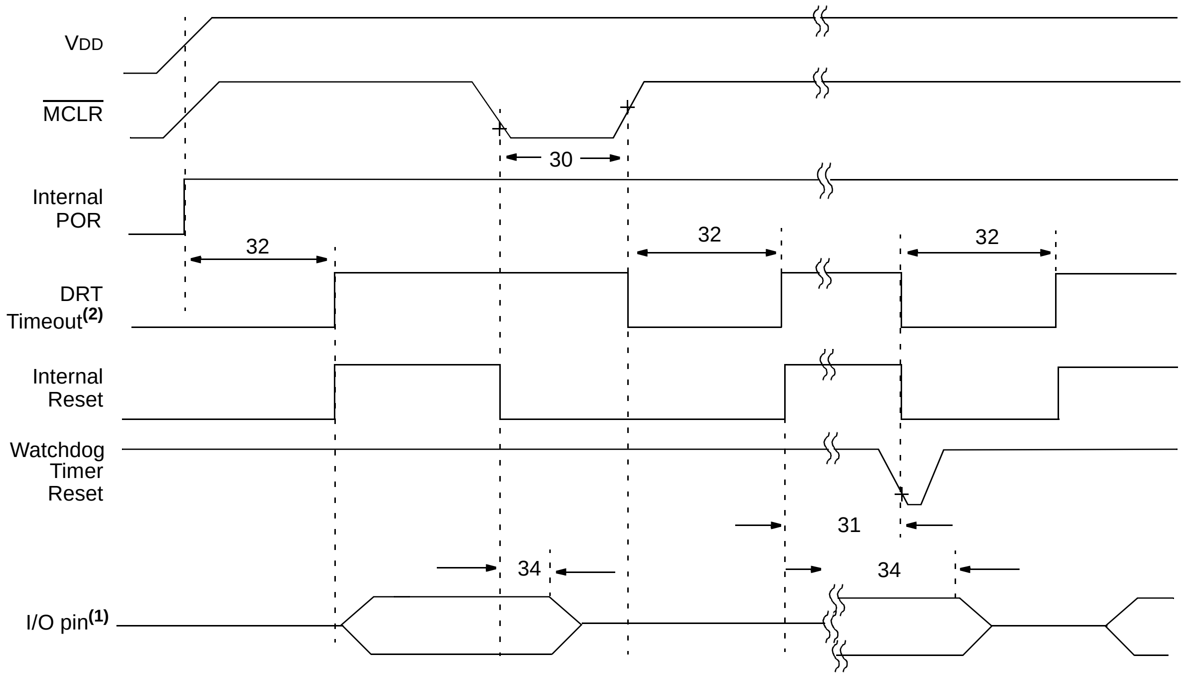
* These parameters are characterized but not tested.

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

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

2: Under stable VDD conditions.

FIGURE 12-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING – PIC10F200/202/204/206



Note 1: I/O pins must be taken out of High-Impedance mode by enabling the output drivers in software.

2: Runs on POR only.

FIGURE 13-2: TYPICAL IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

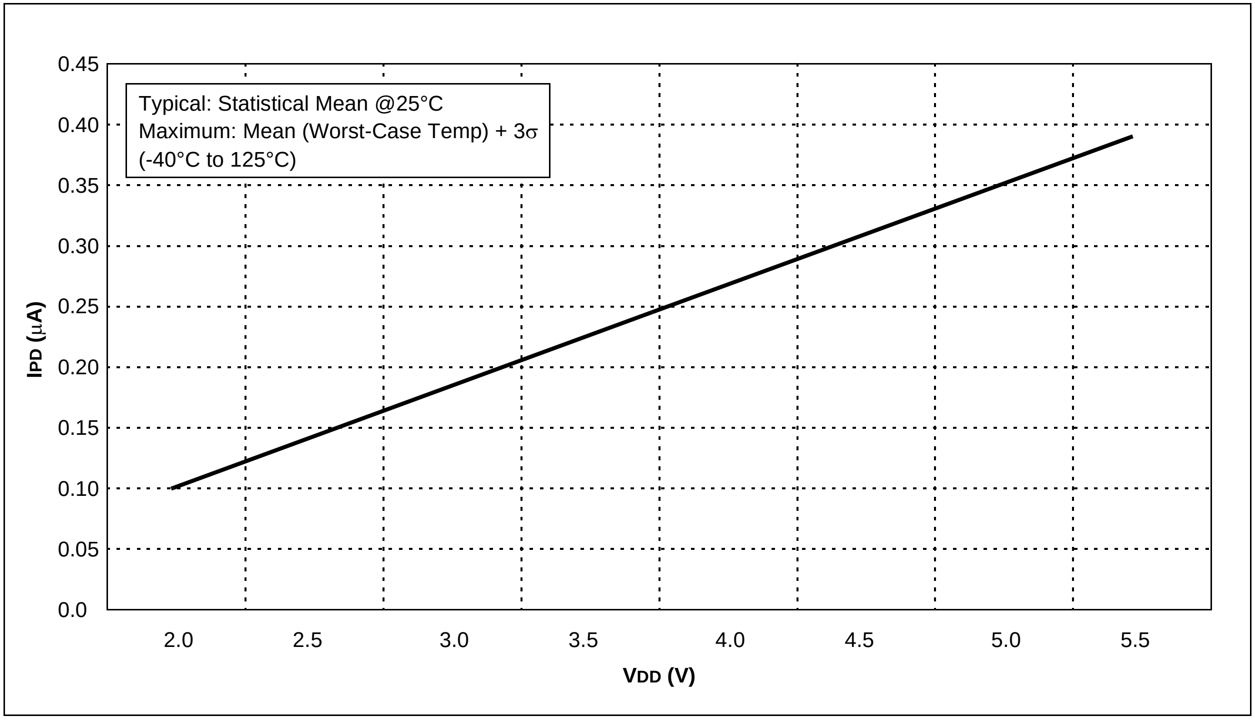


FIGURE 13-3: MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

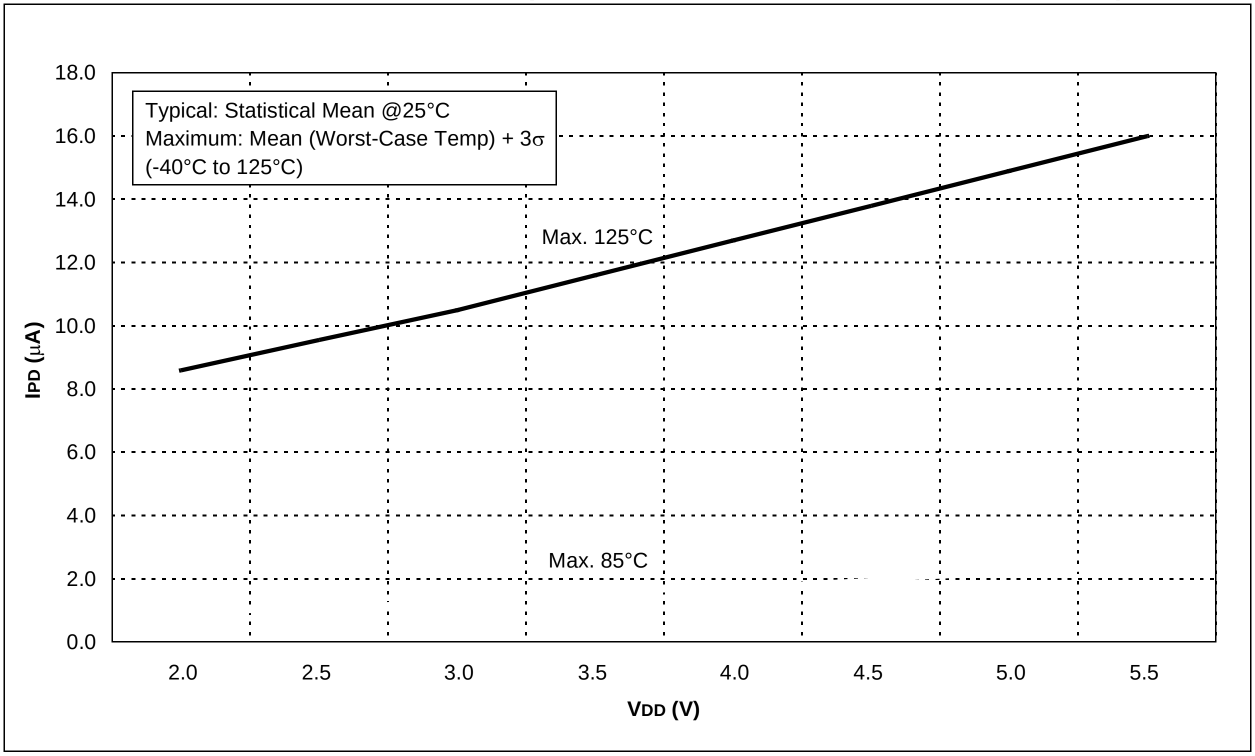


FIGURE 13-10: VOH VS. IOH OVER TEMPERATURE (VDD = 3.0V)

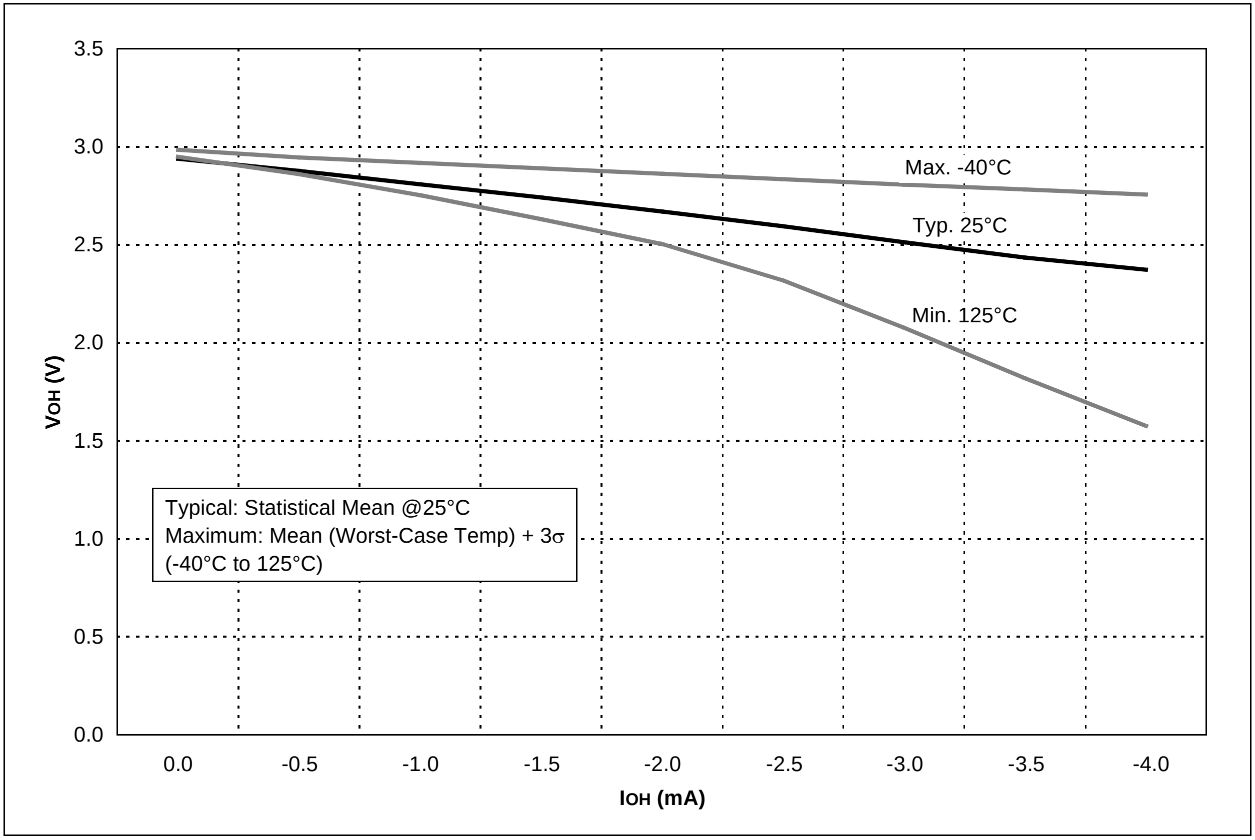
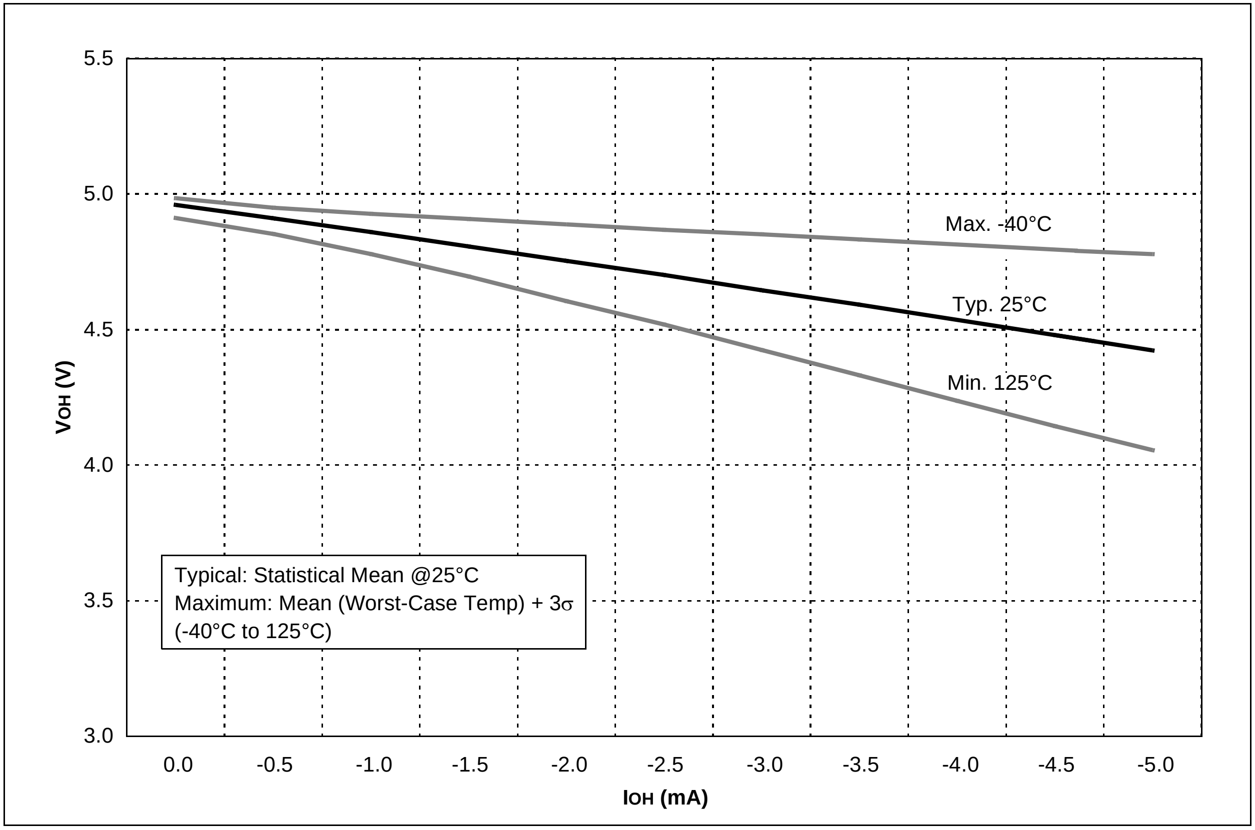
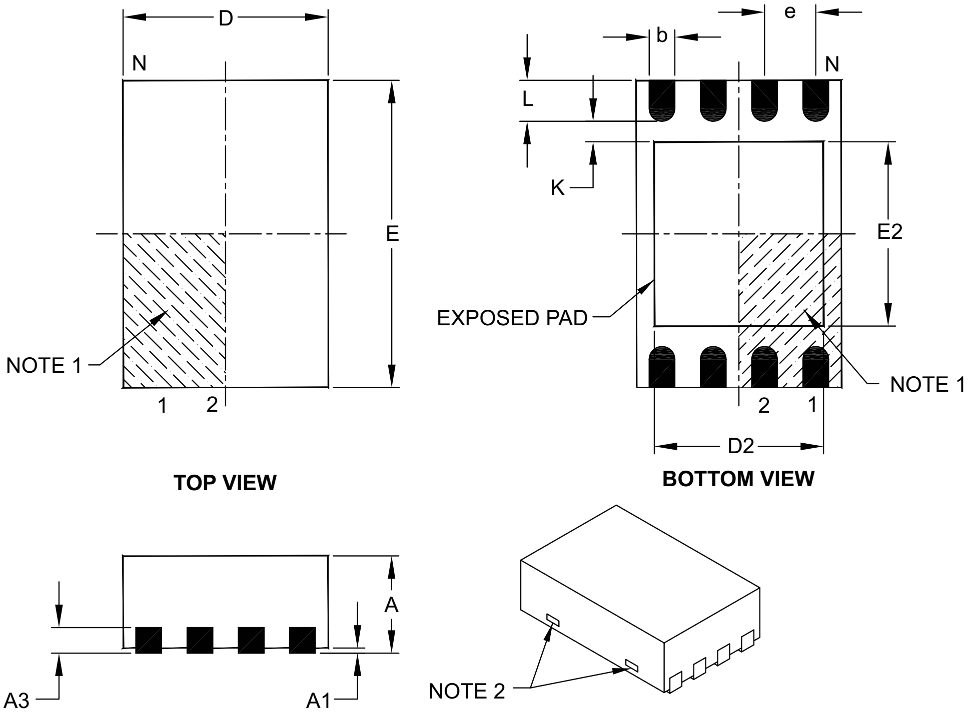


FIGURE 13-11: VOH VS. IOH OVER TEMPERATURE (VDD = 5.0V)



8-Lead Plastic Dual Flat, No Lead Package (MC) – 2x3x0.9 mm Body [DFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	0.50 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Length	D	2.00 BSC		
Overall Width	E	3.00 BSC		
Exposed Pad Length	D2	1.30	–	1.55
Exposed Pad Width	E2	1.50	–	1.75
Contact Width	b	0.20	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-123C

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X][(1)]	-	X	/XX	XXX
Device	**Tape and Reel Option**		**Temperature Range**	**Package**	**Pattern**

Device:	PIC10F200 PIC10F202 PIC10F204 PIC10F206 PIC10F200T (Tape & Reel) PIC10F202T (Tape & Reel) PIC10F204T (Tape & Reel) PIC10F206T (Tape & Reel)	
Tape and Reel Option:	Blank	= Standard packaging (tube or tray)
	T	= Tape and Reel[(1)]
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:	P	= 300 mil PDIP (Pb-free)
	OT	= SOT-23, 6-LD (Pb-free)
	MC	= DFN, 8-LD 2x3 (Pb-free)
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC10F202T - E/OT
Tape and Reel
Extended temperature
SOT-23 package (Pb-free)

b) PIC10F200 - I/P
Industrial temperature,
PDIP package (Pb-free)

c) PIC10F204 - I/MC
Industrial temperature
DFN package (Pb-free)

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC^{32} logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2004-2014, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63276-597-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.