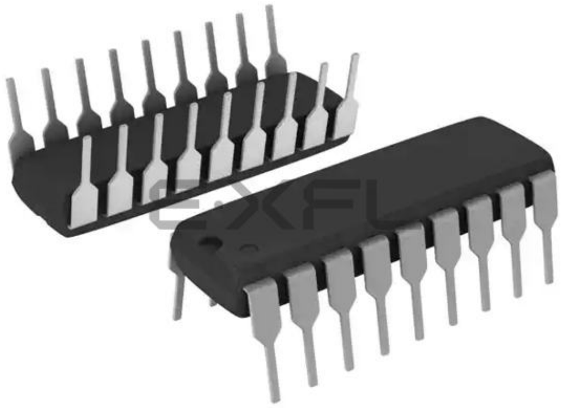


What is "Embedded - Microcontrollers"?

"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	10MHz
Connectivity	I²C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf819-i-ptsl

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
Bank 0											
00h(1)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	23
01h	TMR0	Timer0 Module Register								xxxx xxxx	53, 17
02h(1)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	23
03h(1)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	16
04h(1)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	23
05h	PORTA	PORTA Data Latch when written; PORTA pins when read								xxx0 0000	39
06h	PORTB	PORTB Data Latch when written; PORTB pins when read								xxxx xxxx	43
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	23
0Bh(1)	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	18
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	20
0Dh	PIR2	—	—	—	EEIF	—	—	—	—	---0 ----	21
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	57
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	57
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	57
11h	TMR2	Timer2 Module Register								0000 0000	63
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	64
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	71, 76
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	73
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	66, 67, 68
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	66, 67, 68
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	65
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	81
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	81

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from any bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

3: Pin 5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.

2.2.2.5 PIR1 Register

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

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1 (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are a transmission/reception has taken place.
0 = No SSP interrupt condition has occurred

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
<u>Capture mode:</u>
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
<u>Compare mode:</u>
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
<u>PWM mode:</u>
Unused in this mode.

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.0 DATA EEPROM AND FLASH PROGRAM MEMORY

The data EEPROM and Flash program memory are readable and writable during normal operation (over the full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are six SFRs used to read and write this memory:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

This section focuses on reading and writing data EEPROM and Flash program memory during normal operation. Refer to the appropriate device programming specification document for serial programming information.

When interfacing the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 128 or 256 bytes of data EEPROM, with an address range from 00h to 0FFh. Addresses from 80h to FFh are unimplemented on the PIC16F818 device and will read 00h. When writing to unimplemented locations, the charge pump will be turned off.

When interfacing the program memory block, the EEDATA and EEDATH registers form a two-byte word that holds the 14-bit data for read/write and the EEADR and EEADRH registers form a two-byte word that holds the 13-bit address of the EEPROM location being accessed. These devices have 1K or 2K words of program Flash, with an address range from 0000h to 03FFh for the PIC16F818 and 0000h to 07FFh for the PIC16F819. Addresses above the range of the respective device will wraparound to the beginning of program memory.

The EEPROM data memory allows single byte read and write. The Flash program memory allows single-word reads and four-word block writes. Program memory writes must first start with a 32-word block erase, then write in 4-word blocks. A byte write in data EEPROM memory automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory. Depending on the settings of the write-protect bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed. When code-protected, the device programmer can no longer access data or program memory; this does NOT inhibit internal reads or writes.

3.1 EEADR and EEADRH

The EEADRH:EEADR register pair can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program EEPROM. When selecting a data address value, only the LSB of the address is written to the EEADR register. When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADR register.

If the device contains less memory than the full address reach of the address register pair, the Most Significant bits of the registers are not implemented. For example, if the device has 128 bytes of data EEPROM, the Most Significant bit of EEADR is not implemented on access to data EEPROM.

3.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

Control bit, EEPGD, determines if the access will be a program or data memory access. When clear, as it is when Reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

Control bits, RD and WR, initiate read and write, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write or erase operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write (or erase) operation is interrupted by a $\overline{\text{MCLR}}$ or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit, EEIF in the PIR2 register, is set when the write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the EEPROM write sequence.

4.5.3 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 4-2) controls several aspects of the system clock's operation.

The Internal Oscillator Select bits, IRCF2:IRCF0, select the frequency output of the internal oscillator block that is used to drive the system clock. The choices are the INTRC source (31.25 kHz), the INTOSC source (8 MHz) or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). Changing the configuration of these bits has an immediate change on the multiplexor's frequency output.

4.5.4 MODIFYING THE IRCF BITS

The IRCF bits can be modified at any time regardless of which clock source is currently being used as the system clock. The internal oscillator allows users to change the frequency during run time. This is achieved by modifying the IRCF bits in the OSCCON register. The sequence of events that occur after the IRCF bits are modified is dependent upon the initial value of the IRCF bits before they are modified. If the INTRC (31.25 kHz, IRCF<2:0> = `000`) is running and the IRCF bits are modified to any other value than '`000`', a 4 ms (approx.) clock switch delay is turned on. Code execution continues at a higher than expected frequency while the new frequency stabilizes. Time sensitive code should wait for the IOFS bit in the OSCCON register to become set before continuing. This bit can be monitored to ensure that the frequency is stable before using the system clock in time critical applications.

If the IRCF bits are modified while the internal oscillator is running at any other frequency than INTRC (31.25 kHz, IRCF<2:0> ≠ `000`), there is no need for a 4 ms (approx.) clock switch delay. The new INTOSC frequency will be stable immediately after the **eight** falling edges. The IOFS bit will remain set after clock switching occurs.

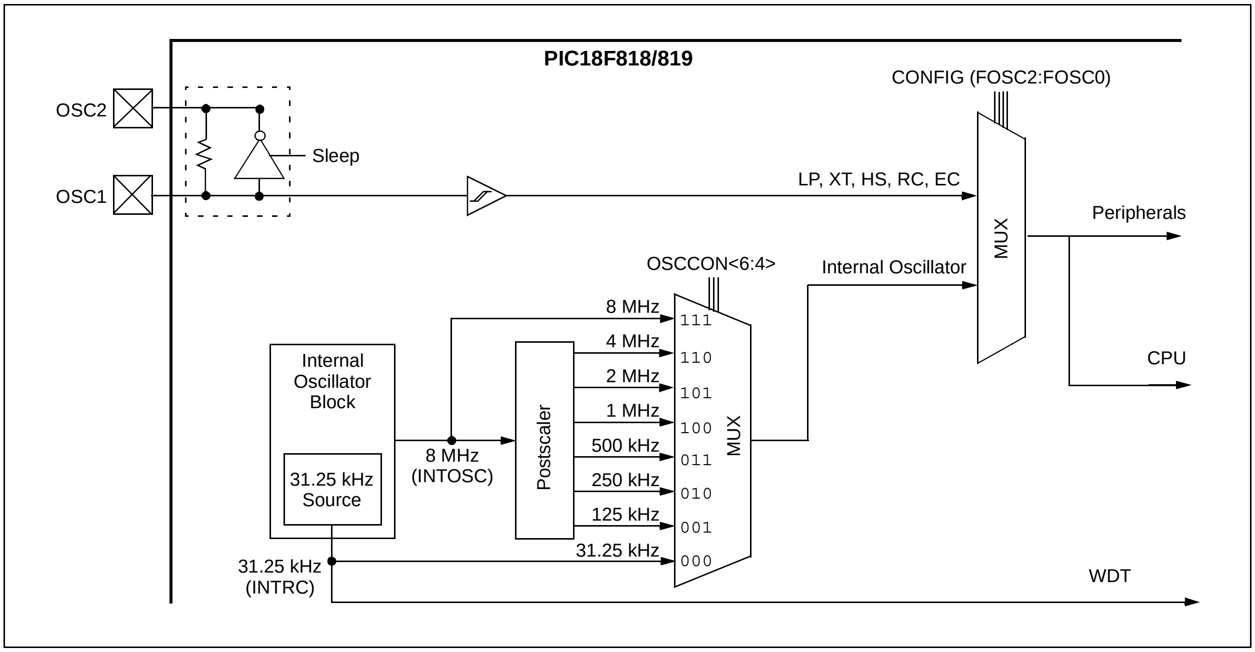
Note:	Caution must be taken when modifying the IRCF bits using `BCF` or `BSF` instructions. It is possible to modify the IRCF bits to a frequency that may be out of the V_{DD} specification range; for example, V_{DD} = 2.0V and IRCF = `111` (8 MHz).

4.5.5 CLOCK TRANSITION SEQUENCE WHEN THE IRCF BITS ARE MODIFIED

Following are three different sequences for switching the internal RC oscillator frequency.

- Clock before switch: 31.25 kHz (IRCF<2:0> = `000`)
  1. IRCF bits are modified to an INTOSC/INTOSC postscaler frequency.
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. The IOFS bit is clear to indicate that the clock is unstable and a 4 ms (approx.) delay is started. Time dependent code should wait for IOFS to become set.
  5. Switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ `000`)
  1. IRCF bits are modified to INTRC (IRCF<2:0> = `000`).
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. Oscillator switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ `000`)
  1. IRCF bits are modified to a different INTOSC/INTOSC postscaler frequency.
  2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
  3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
  4. The IOFS bit is set.
  5. Oscillator switchover is complete.

FIGURE 4-6: PIC16F818/819 CLOCK DIAGRAM



REGISTER 4-2: OSCCON: OSCILLATOR CONTROL REGISTER (ADDRESS 8Fh)

U-0	R/W-0	R/W-0	R/W-0	U-0	R-0	U-0	U-0
—	IRCF2	IRCF1	IRCF0	—	IOFS	—	—
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
111 = 8 MHz (8 MHz source drives clock directly)
110 = 4 MHz
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31.25 kHz (INTRC source drives clock directly)

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **IOFS:** INTOSC Frequency Stable bit
1 = Frequency is stable
0 = Frequency is not stable

bit 1-0 **Unimplemented:** Read as ‘0’

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

5.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORTB's pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

PORTB is multiplexed with several peripheral functions (see Table 5-3). PORTB pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTB pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (`BSF, BCF, XORWF`) with TRISB as the destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

TABLE 5-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit 0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1/SDI/SDA	bit 1	TTL/ST(5)	Input/output pin, SPI data input pin or I^2C™ data I/O pin. Internal software programmable weak pull-up.
RB2/SDO/CCP1	bit 2	TTL/ST(4)	Input/output pin, SPI data output pin or Capture input/Compare output/PWM output pin. Internal software programmable weak pull-up.
RB3/CCP1/PGM(3)	bit 3	TTL/ST(2)	Input/output pin, Capture input/Compare output/PWM output pin or programming in LVP mode. Internal software programmable weak pull-up.
RB4/SCK/SCL	bit 4	TTL/ST(5)	Input/output pin or SPI and I^2C clock pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5/$\overline{SS}$	bit 5	TTL	Input/output pin or SPI slave select pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/T1OSO/T1CKI/ PGC	bit 6	TTL/ST(2)	Input/output pin, Timer1 oscillator output pin, Timer1 clock input pin or serial programming clock (with interrupt-on-change). Internal software programmable weak pull-up.
RB7/T1OSI/PGD	bit 7	TTL/ST(2)	Input/output pin, Timer1 oscillator input pin or serial programming data (with interrupt-on-change). Internal software programmable weak pull-up.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: Low-Voltage ICSP™ Programming (LVP) is enabled by default which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 18-pin mid-range devices.

4: This buffer is a Schmitt Trigger input when configured for CCP or SSP mode.

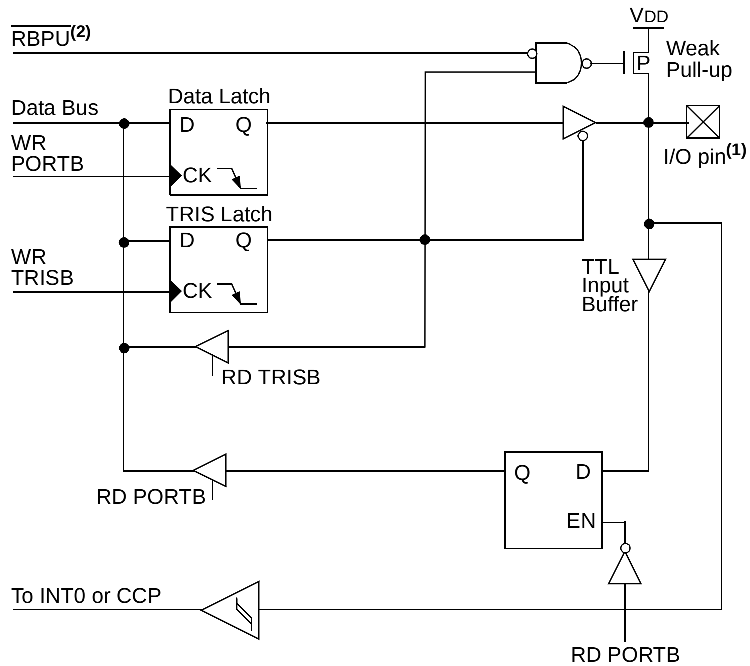
5: This buffer is a Schmitt Trigger input when configured for SPI or I^2C mode.

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	`uuuu uuuu`
86h, 186h	TRISB	PORTB Data Direction Register								`1111 1111`	`1111 1111`
81h, 181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	`1111 1111`	`1111 1111`

Legend: `x` = unknown, `u` = unchanged. Shaded cells are not used by PORTB.

FIGURE 5-8: BLOCK DIAGRAM OF RB0 PIN



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.

9.3 PWM Mode

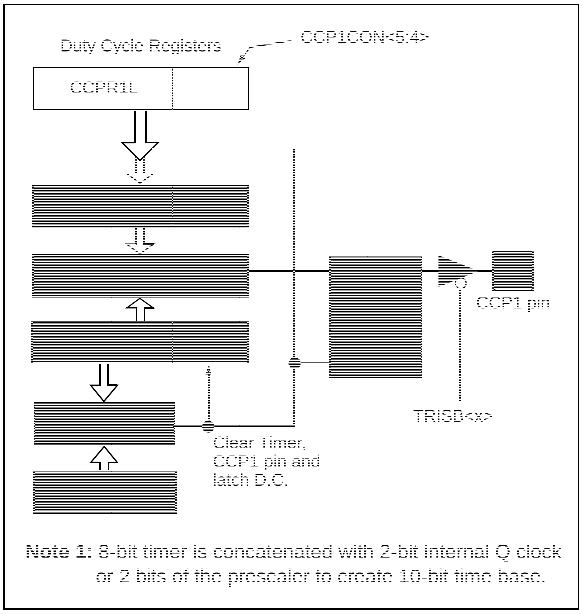
In Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTB data latch, the TRISB<x> bit must be cleared to make the CCP1 pin an output.

Note:	Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTB I/O data latch.

Figure 9-3 shows a simplified block diagram of the CCP module in PWM mode.

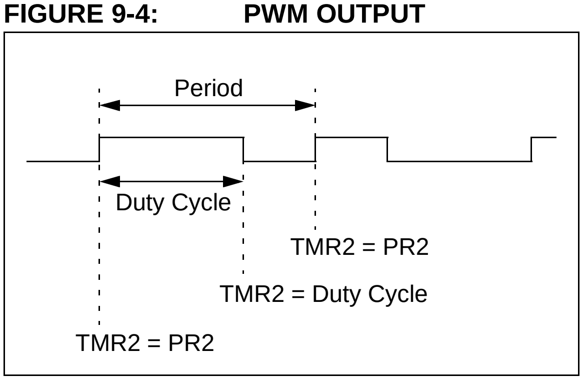
For a step by step procedure on how to set up the CCP module for PWM operation, see **Section 9.3.3 “Setup for PWM Operation”**.

FIGURE 9-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 9-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 9-4: PWM OUTPUT



9.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula.

EQUATION 9-1:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note:	The Timer2 postscaler (see **Section 8.0 “Timer2 Module”**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

9.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time.

EQUATION 9-2:

$$\text{PWM Duty Cycle} = (CCPR1L{:}CCP1CON<5{:}4>) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

12.2 Reset

The PIC16F818/819 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset during normal operation
- WDT wake-up during Sleep
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), on the $\overline{\text{MCLR}}$ and WDT Reset, on $\overline{\text{MCLR}}$ Reset during Sleep and Brown-out Reset (BOR). They are not affected by a WDT wake-up which is viewed as the resumption of normal operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations as indicated in Table 12-3. These bits are used in software to determine the nature of the Reset. Upon a POR, BOR or wake-up from Sleep, the CPU requires approximately 5-10 μs to become ready for code execution. This delay runs in parallel with any other timers. See Table 12-4 for a full description of Reset states of all registers.

A simplified block diagram of the on-chip Reset circuit is shown in Figure 12-1.

FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

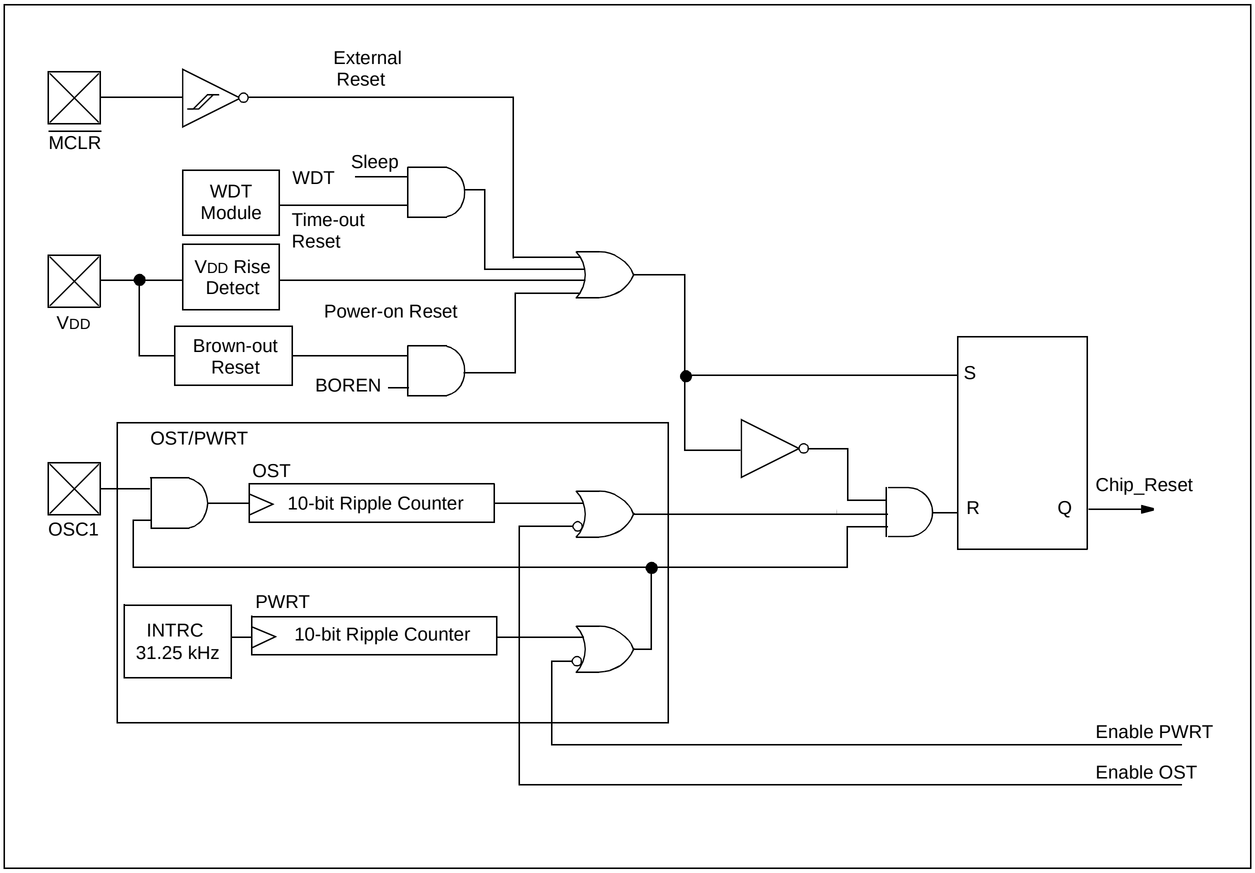


FIGURE 12-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH PULL-UP RESISTOR)

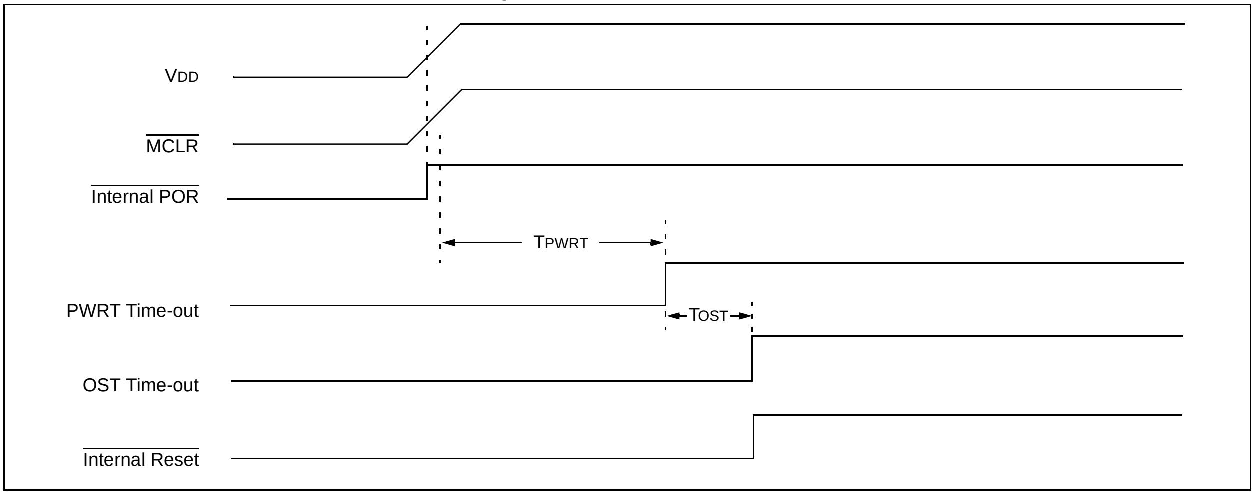


FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH RC NETWORK): CASE 1

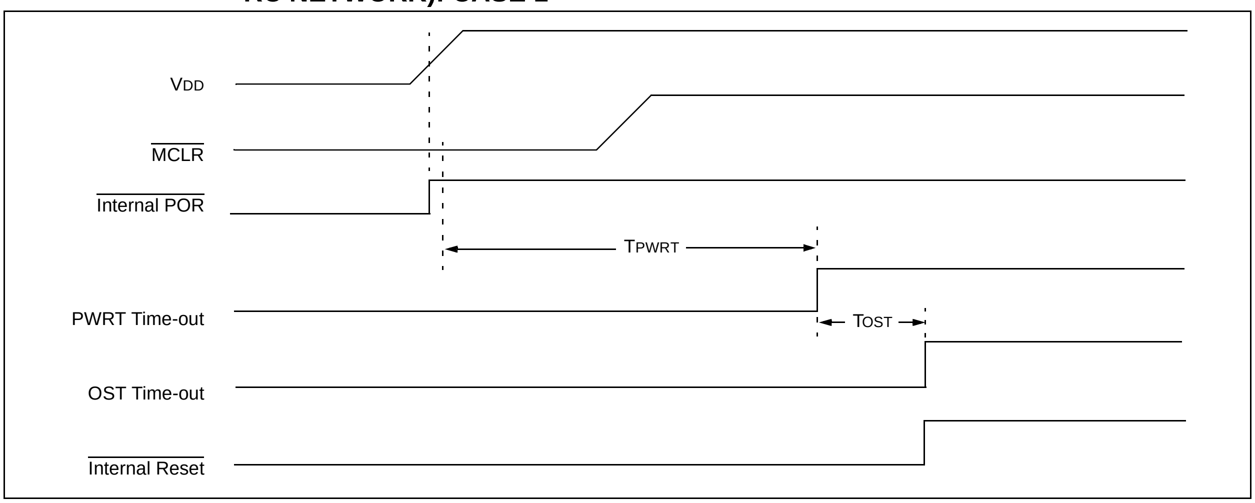
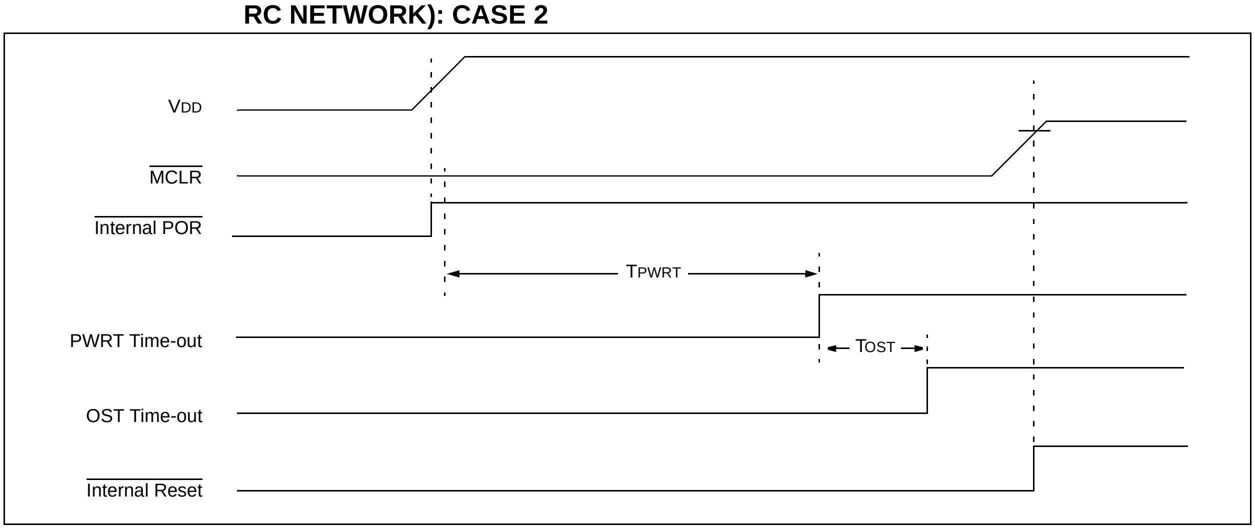


FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD THROUGH RC NETWORK): CASE 2



13.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories are presented in Figure 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ assembler. A complete description of each instruction is also available in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven-bit constant or literal value

One instruction cycle consists of four oscillator periods. For an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a `NOP`.

Note:	To maintain upward compatibility with future PIC16F818/819 products, do not use the `OPTION` and `TRIS` instructions.

All instruction examples use the format '`0xhh`' to represent a hexadecimal number, where '`h`' signifies a hexadecimal digit.

13.1 READ-MODIFY-WRITE OPERATIONS

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a "`CLRF PORTB`" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared.

TABLE 13-1: OPCODE FIELD DESCRIPTIONS

Field	Description
`f`	Register file address (0x00 to 0x7F)
`W`	Working register (accumulator)
`b`	Bit address within an 8-bit file register
`k`	Literal field, constant data or label
`x`	Don't care location (= `0` or `1`). The assembler will generate code with x = `0`. It is the recommended form of use for compatibility with all Microchip software tools.
`d`	Destination select; d = `0`: store result in W, d = `1`: store result in file register f. Default is d = `1`.
`PC`	Program Counter
`TO`	Time-out bit
`PD`	Power-Down bit

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 – 8	7	6 – 0
OPCODE	d	f (FILE #)

d = `0` for destination W
d = `1` for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 – 10	9 – 7	6 – 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 – 8	7 – 0
OPCODE	k (literal)

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 – 11	10 – 0
OPCODE	k (literal)

k = 11-bit immediate value

TABLE 13-2: PIC16F818/819 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ANDWF	**f, d**	AND W with f	1	00	0101	dfff	ffff	Z	1, 2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1, 2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	**f, d**	Decrement f, Skip if 0	1 (2)	00	1011	dfff	ffff		1, 2, 3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	**f, d**	Increment f, Skip if 0	1 (2)	00	1111	dfff	ffff		1, 2, 3
IORWF	**f, d**	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1, 2
RRF	**f, d**	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1, 2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1, 2
XORWF	**f, d**	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1, 2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1, 2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	**k**	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	**k**	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: 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`'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = `1`), the prescaler will be cleared if assigned to the Timer0 module.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

FIGURE 15-1: PIC16F818/819 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL, EXTENDED)

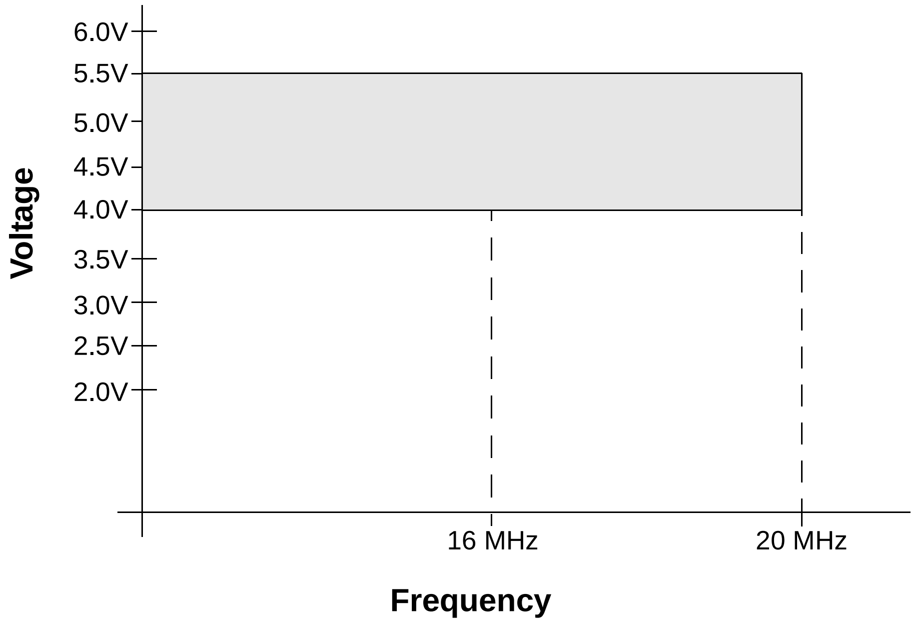
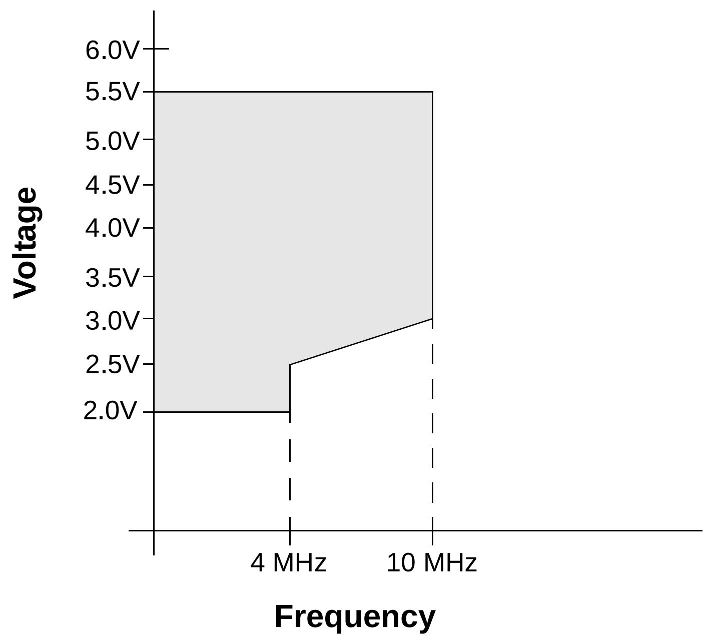


FIGURE 15-2: PIC16LF818/819 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



F_{MAX} = (12 MHz/V) ($V_{DDAPPMIN}$ – 2.5V) + 4 MHz

Note 1: $V_{DDAPPMIN}$ is the minimum voltage of the PIC® device in the application.

2: F_{MAX} has a maximum frequency of 10 MHz.

15.2 DC Characteristics: Power-Down and Supply Current PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial) (Continued)

PIC16LF818/819 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial
PIC16F818/819 (Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial -40°C ≤ T_A ≤ +125°C for extended

Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (I_{DD})(2,3)						
	All devices	1.8	2.3	mA	-40°C	V_{DD} = 4.0V	F_{OSC} = 20 MHz (HS Oscillator)
		1.6	2.2	mA	+25°C		
		1.3	2.2	mA	+85°C		
	All devices	3.0	4.2	mA	-40°C	V_{DD} = 5.0V	
		2.5	4.0	mA	+25°C		
		2.5	4.0	mA	+85°C		
	Extended devices	3.0	5.0	mA	+125°C		

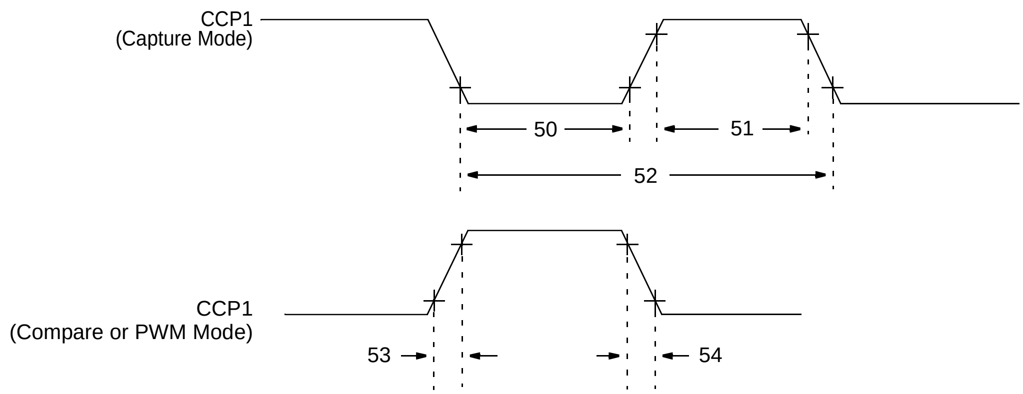
Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all I_{DD} measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD};
$\overline{MCLR}$ = V_{DD}; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be estimated by the formula $Ir = V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

FIGURE 15-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)



Note: Refer to Figure 15-3 for load conditions.

TABLE 15-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)

Param No.	Symbol	Characteristic			Min	Typ†	Max	Units	Conditions
50*	TCCL	CCP1 Input Low Time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	PIC16**F**818/819	10	—	—	ns	
				PIC16**LF**818/819	20	—	—	ns	
51*	TCCH	CCP1 Input High Time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	PIC16**F**818/819	10	—	—	ns	
				PIC16**LF**818/819	20	—	—	ns	
52*	TCCP	CCP1 Input Period			$\frac{3\ T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TCCR	CCP1 Output Rise Time		PIC16**F**818/819	—	10	25	ns	
				PIC16**LF**818/819	—	25	50	ns	
54*	TCCF	CCP1 Output Fall Time		PIC16**F**818/819	—	10	25	ns	
				PIC16**LF**818/819	—	25	45	ns	

* These parameters are characterized but not tested.

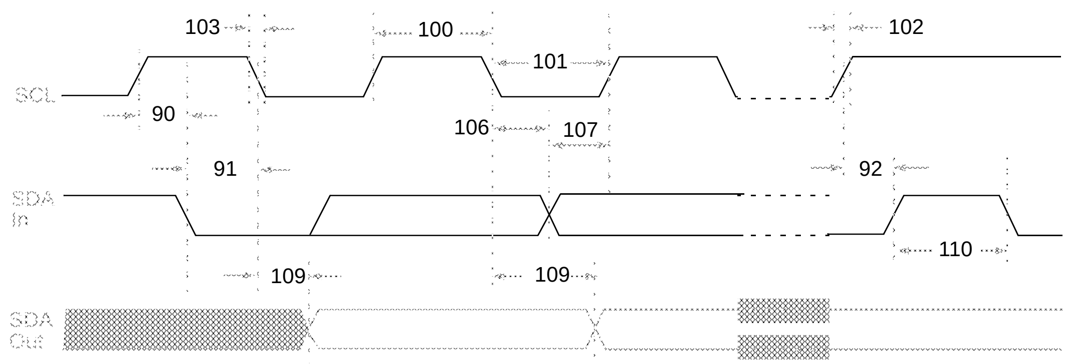
† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 15-7: I²C™ BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
90*	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
91*	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
92*	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 15-15: I²C™ BUS DATA TIMING



Note: Refer to Figure 15-3 for load conditions.

FIGURE 16-5: TYPICAL IDD vs. FOSC OVER VDD (LP MODE)

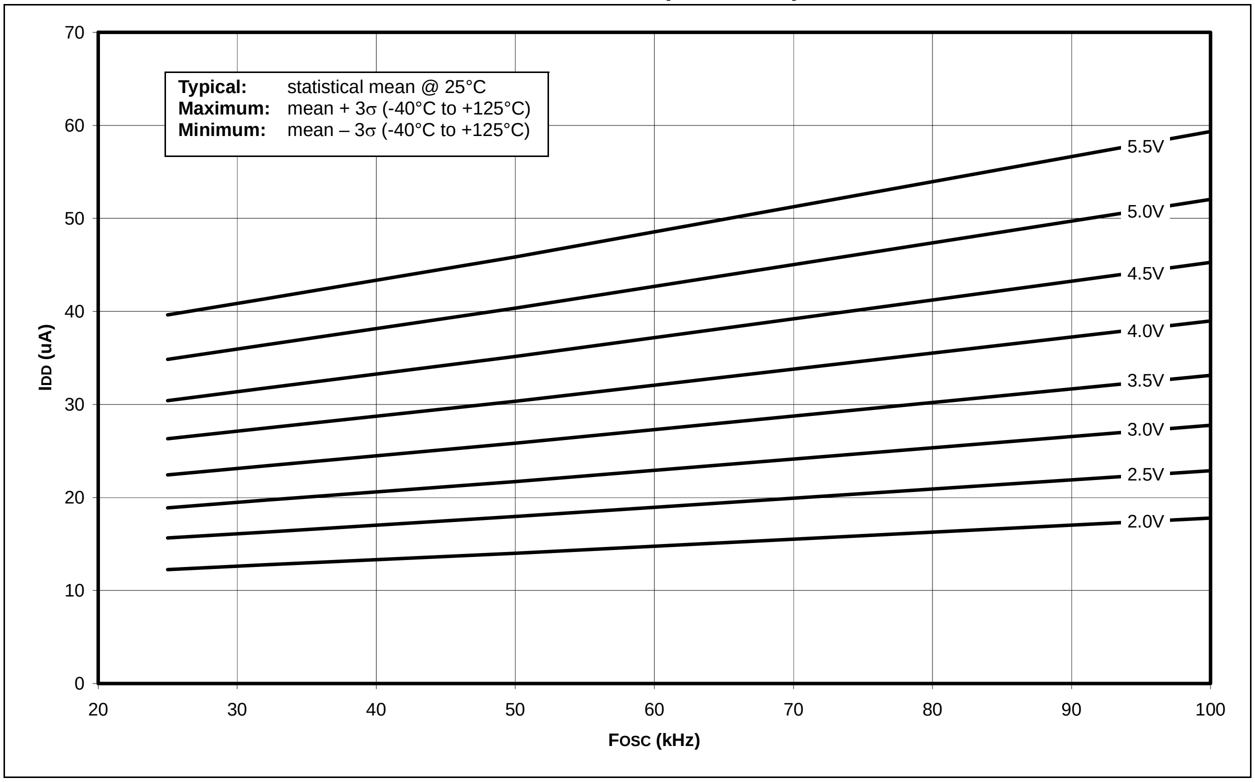


FIGURE 16-6: MAXIMUM IDD vs. FOSC OVER VDD (LP MODE)

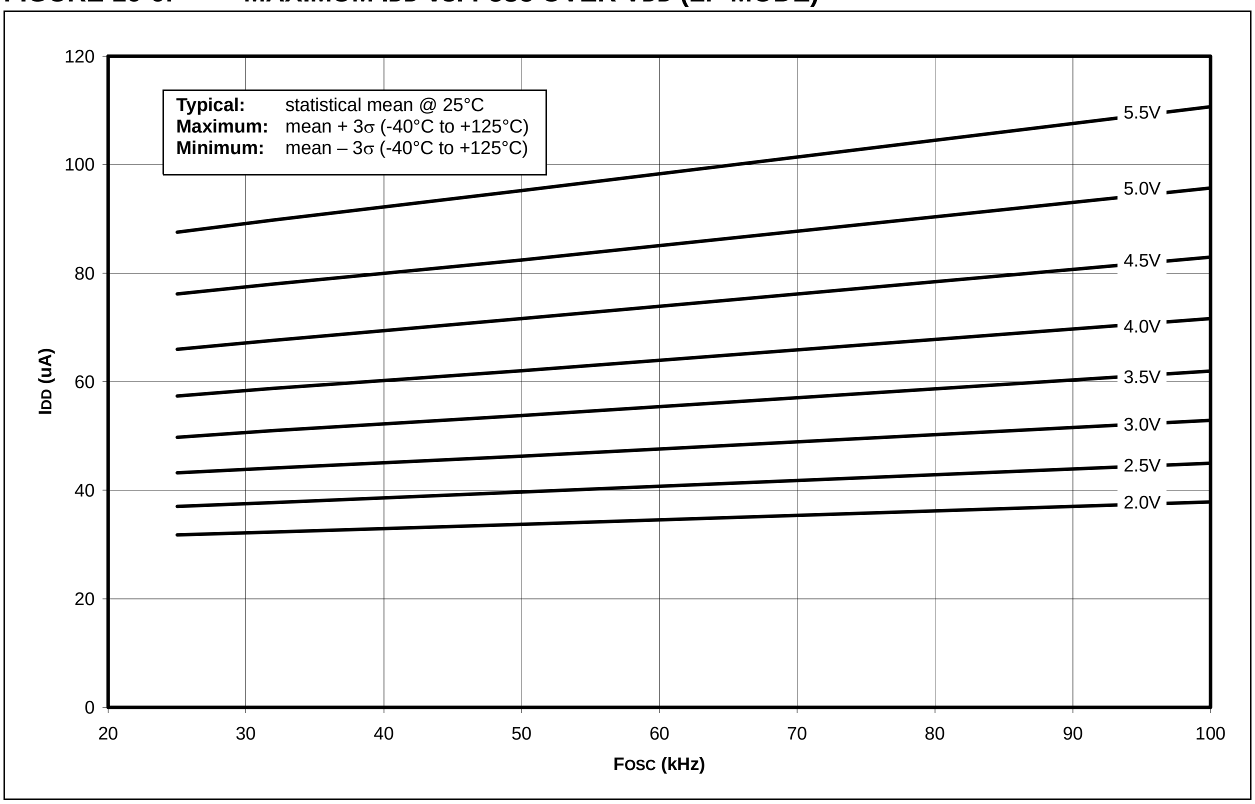


FIGURE 16-13: ΔIPD TIMER1 OSCILLATOR, -10°C TO +70°C (SLEEP MODE, TMR1 COUNTER DISABLED)

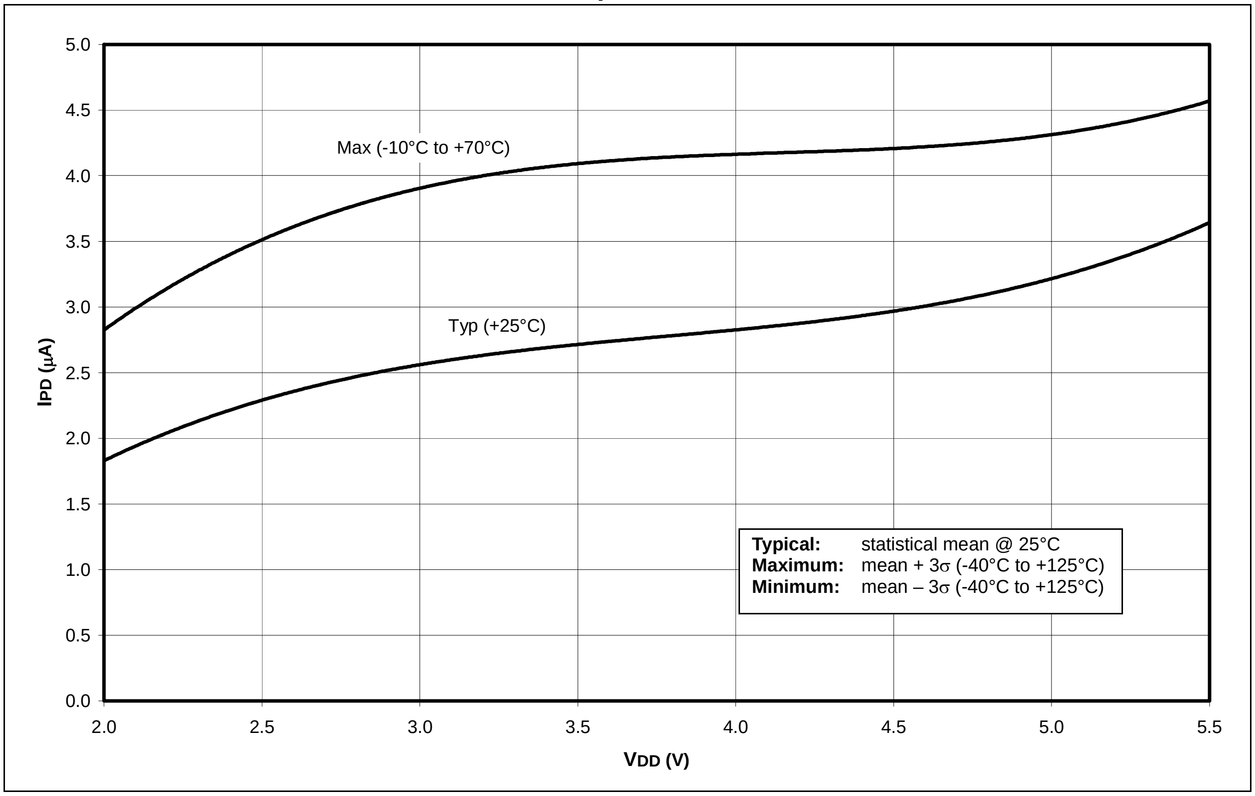
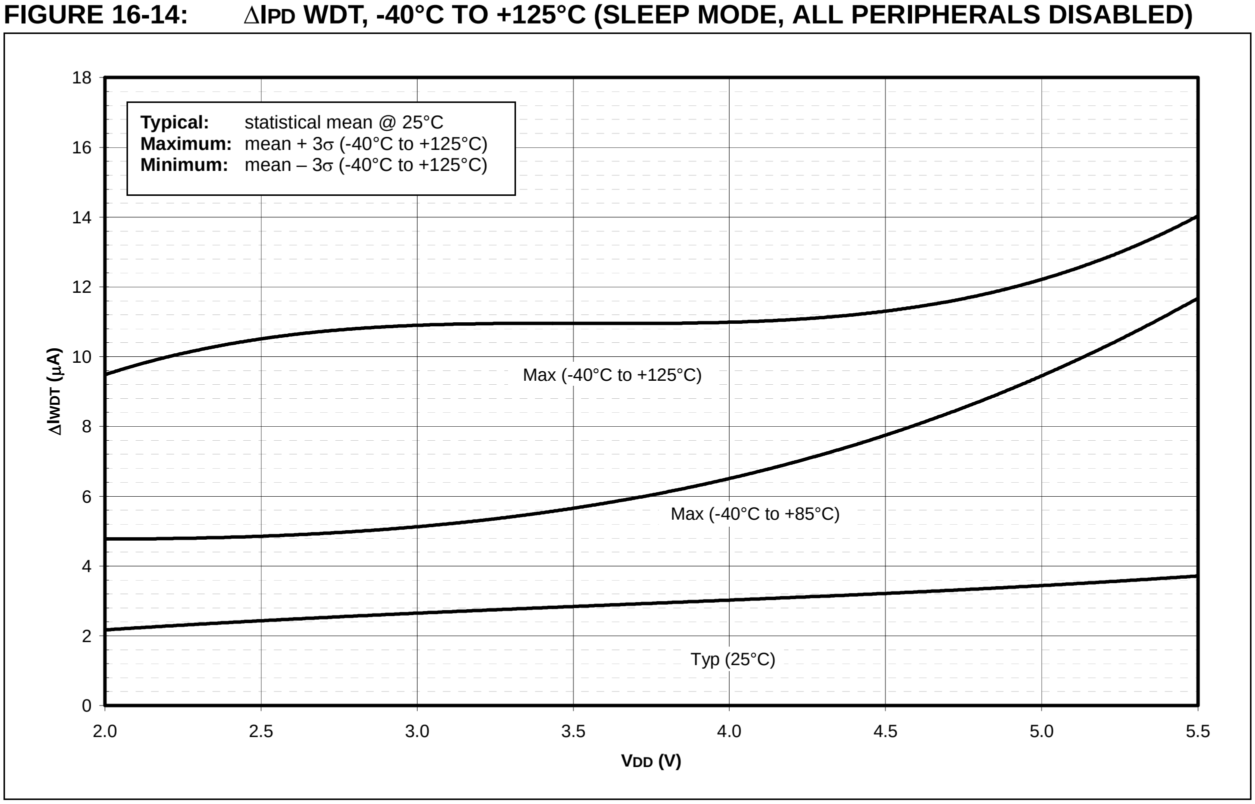
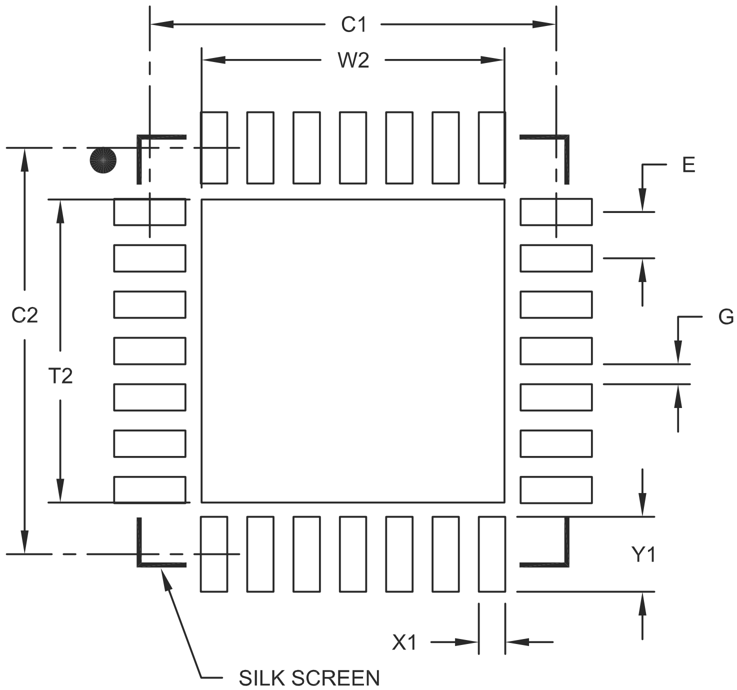


FIGURE 16-14: ΔIPD WDT, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)



28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

