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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 8x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1614-e-sl

2.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 7.5 "Automatic Context Saving"**, for more information.

2.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See section **Section 3.5 "Stack"** for more details.

2.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can now also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. There are also new instructions to support the FSRs. See **Section 3.6 "Indirect Addressing"** for more details.

2.4 Instruction Set

There are 49 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 34.0 "Instruction Set Summary"** for more details.

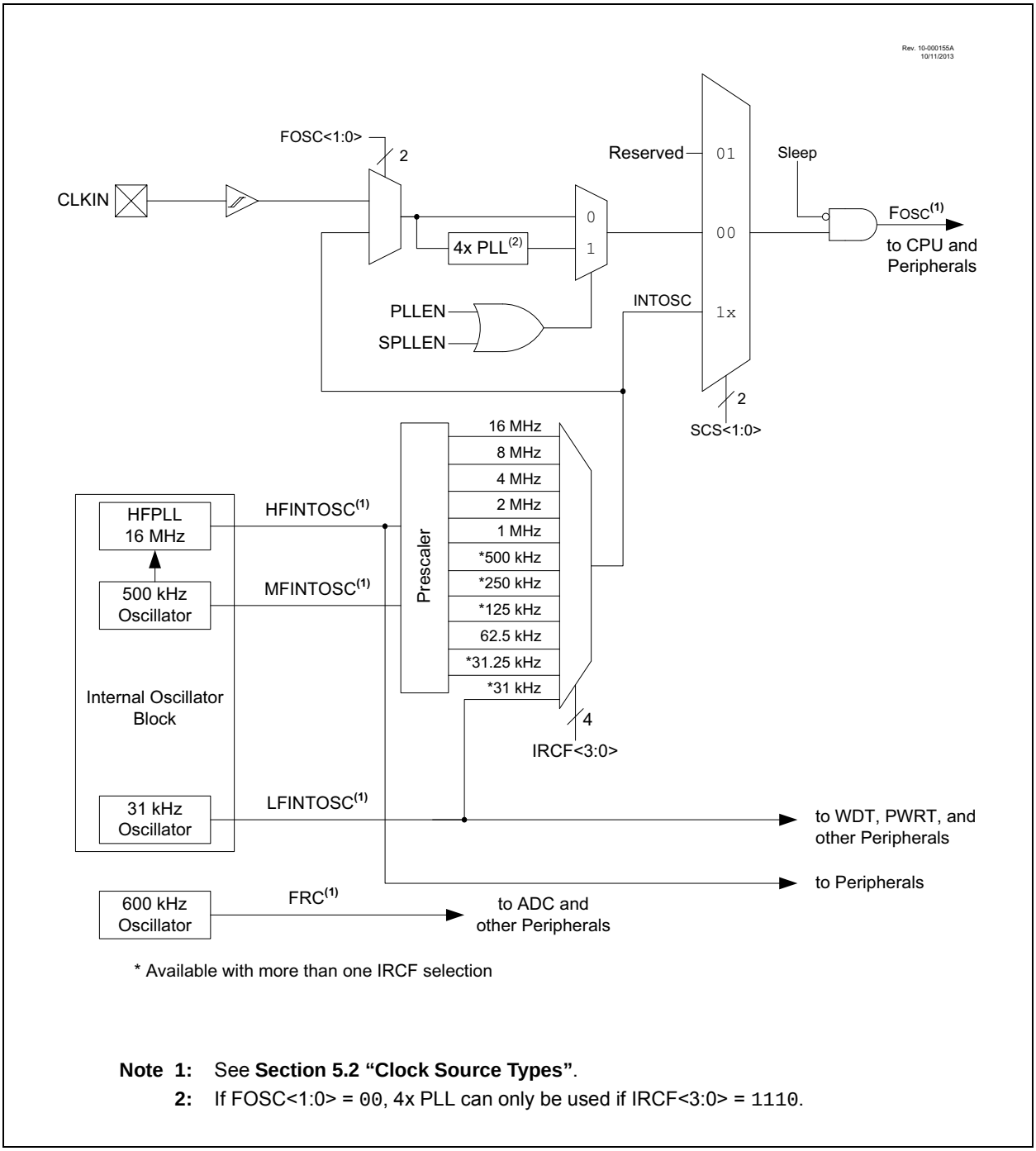
REGISTER 4-3: CONFIG3: CONFIGURATION WORD 3 (CONTINUED)

bit 4-0 **WDTCPS<4:0>:** WDT Configuration Period Select bits

WDTCPS <4:0>	WDTPS at POR				Software control of WDTPS	
	Value	Divider Ratio		Typical time out (FIN = 31 kHz)		
11111	01011	1:65536	2^{16}	2 s	Yes	Default fuse = 11111
10011 ... 11110	10011 ... 11110	1:32	2^5	1 ms	No	
10010	10010	1:8388608	2^{23}	256 s		
10001	10001	1:4194304	2^{22}	128 s		
10000	10000	1:2097152	2^{21}	64 s		
01111	01111	1:1048576	2^{20}	32 s		
01110	01110	1:524299	2^{19}	16 s		
01101	01101	1:262144	2^{18}	8 s		
01100	01100	1:131072	2^{17}	4 s		
01011	01011	1:65536	2^{16}	2 s		
01010	01010	1:32768	2^{15}	1 s		
01001	01001	1:16384	2^{14}	512 ms	No	
01000	01000	1:8192	2^{13}	256 ms		
00111	00111	1:4096	2^{12}	128 ms		
00110	00110	1:2048	2^{11}	64 ms		
00101	00101	1:1024	2^{10}	32 ms		
00100	00100	1:512	2^9	16 ms		
00011	00011	1:256	2^8	8 ms		
00010	00010	1:128	2^7	4 ms		
00001	00001	1:64	2^6	2 ms		
00000	00000	1:32	2^5	1 ms		

Note 1: A window delay of 12.5% is only available in Software Control mode via the WDTCON1 register.

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



Note 1: See **Section 5.2 “Clock Source Types”**.

2: If FOSC<1:0> = 00, 4x PLL can only be used if IRCF<3:0> = 1110.

5.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Words
- Internal Oscillator Block (INTOSC)

5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by value of the FOSC<1:0> bits in the Configuration Words.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

TABLE 5-1: OSCILLATOR SWITCHING DELAYS

Switch From	Switch To	Frequency	Oscillator Delay
Sleep	LFINTOSC(1) MFINTOSC(1) HFINTOSC(1)	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	Oscillator Warm-up Delay (Tiosc st)
Sleep/POR	EC(1)	DC – 32 MHz	2 cycles
LFINTOSC	EC(1)	DC – 32 MHz	1 cycle of each
Any clock source	MFINTOSC(1) HFINTOSC(1)	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs (approx.)
Any clock source	LFINTOSC(1)	31 kHz	1 cycle of each
PLL inactive	PLL active	16-32 MHz	2 ms (approx.)

Note 1: PLL inactive.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1, PIE2 and PIE3 registers)

The INTCON, PIR1, PIR2 and PIR3 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section7.5 "Automatic Context Saving"."**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

REGISTER 11-16: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **TSEL<3:0>**: Scanner Data Trigger Input Selection bits
1111-1010 = Reserved
1001 = SMT2_Match
1000 = SMT1_Match
0111 = TMR0_Overflow
0110 = TMR5_Overflow
0101 = TMR3_Overflow
0100 = TMR1_Overflow
0011 = TMR6_postscaled
0010 = TMR4_postscaled
0001 = TMR2_postscaled
0000 = LFINTOSC

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CRCACCH	ACC<15:8>								142
CRCACCL	ACC<7:0>								142
CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	141
CRCCON1	DLEN<3:0>				PLEN<3:0>				141
CRCDATH	DAT<15:8>								142
CRCDATL	DAT<7:0>								142
CRCSHIFTH	SHIFT<15:8>								143
CRCSHIFTL	SHIFT<7:0>								143
CRCXORH	XOR<15:8>								143
CRCXORL	XOR<7:1>							—	143
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	106
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	101
SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		144
SCANHADRH	HADR<15:8>								146
SCANHADRL	HADR<7:0>								146
SCANLADRH	LADR<15:8>								145
SCANLADRL	LADR<7:0>								145
SCANTRIG					TSEL<3:0>				147

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for the CRC module.
* Page provides register information.

12.6 Register Definitions: PORTC

REGISTER 12-17: PORTC: PORTC REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **RC<7:0>**: PORTC I/O Value bits(1, 2)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

Note 1: RC<7:6> on PIC16(L)F1618 only.
2: Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

REGISTER 12-18: TRISC: PORTC TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **TRISC<7:0>**: PORTC Tri-State Control bits(1)
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

Note 1: TRISC<7:6> on PIC16(L)F1618 only.

REGISTER 13-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-1 **Unimplemented:** Read as ‘0’

bit 0 **PPSLOCKED:** PPS Locked bit

1 = PPS is locked. PPS selections can not be changed.

0 = PPS is not locked. PPS selections can be changed.

FIGURE 22-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE

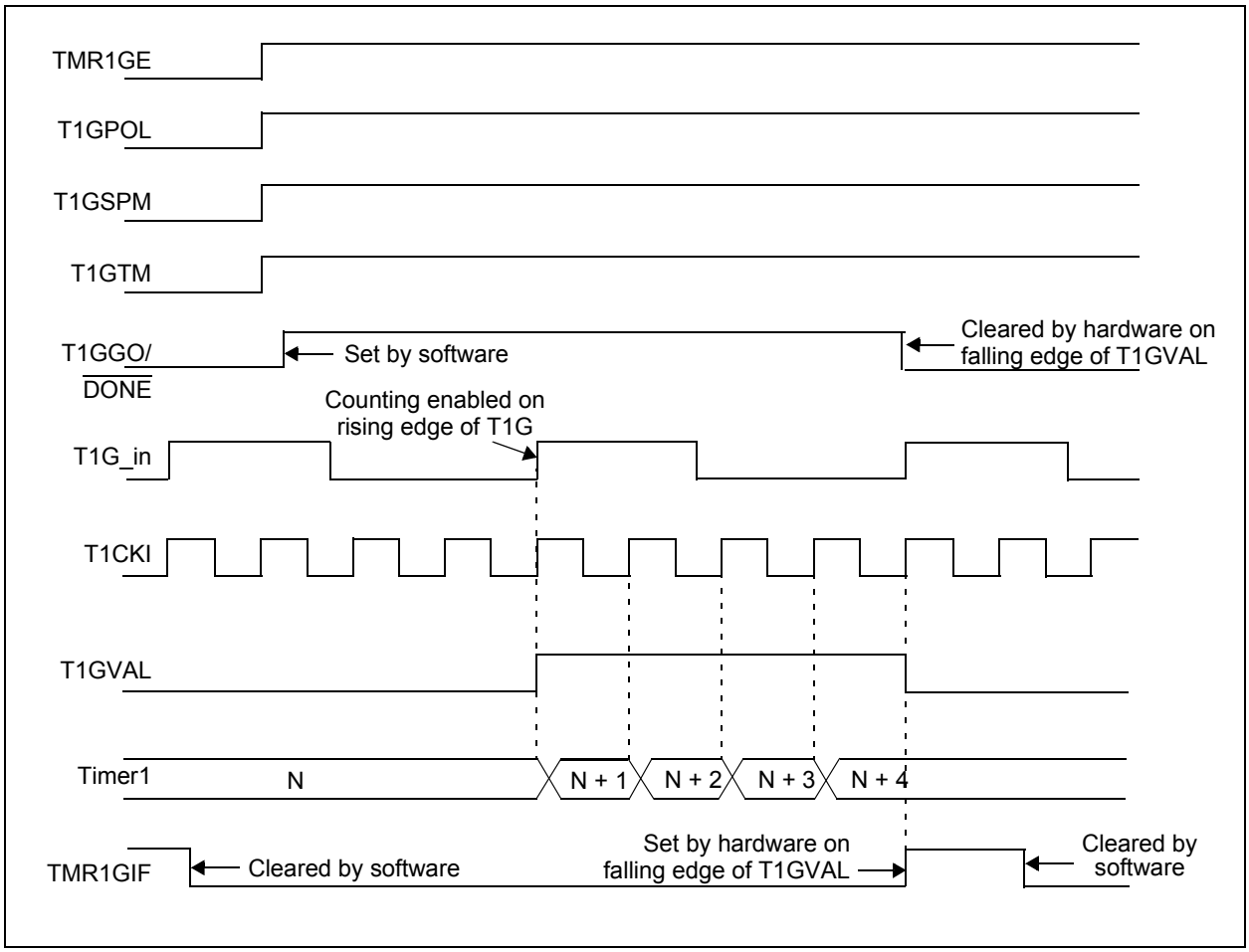


TABLE 22-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	152
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								227*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								227*
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Count								227*
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Count								227*
TMR5H	Holding Register for the Most Significant Byte of the 16-bit TMR5 Count								227*
TMR5L	Holding Register for the Least Significant Byte of the 16-bit TMR5 Count								227*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	151
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON	231
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		232
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		—	T3SYNC	—	TMR3ON	231
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>		232
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		—	T5SYNC	—	TMR5ON	231
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		232

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.
* Page provides register information.
Note 1: Unimplemented, read as '1'.

24.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 24-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 24-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 24-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

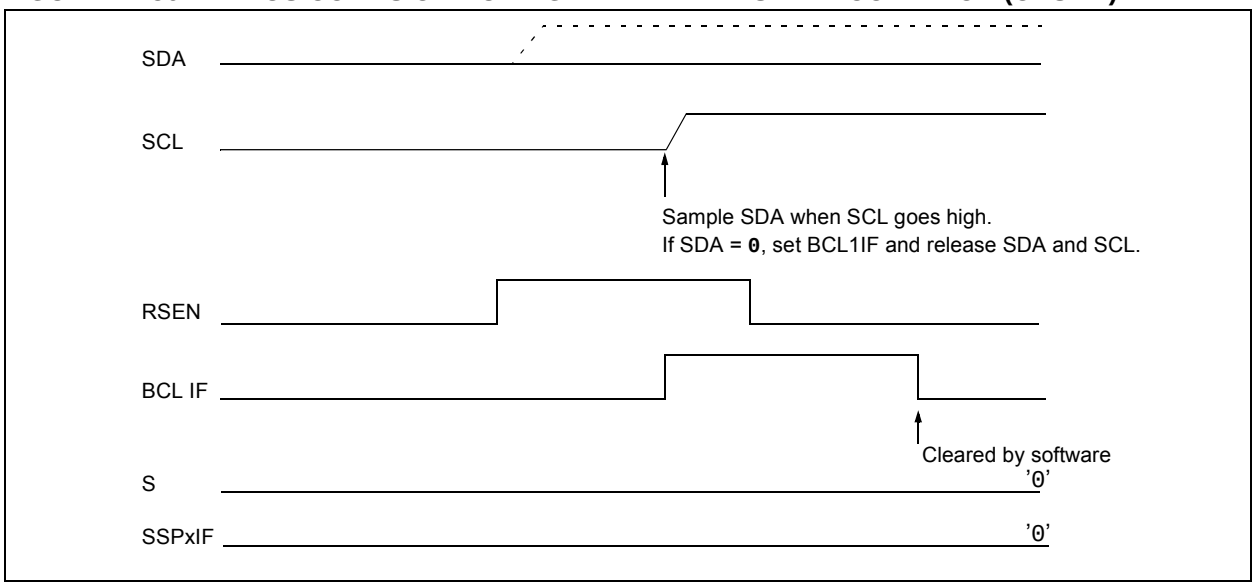
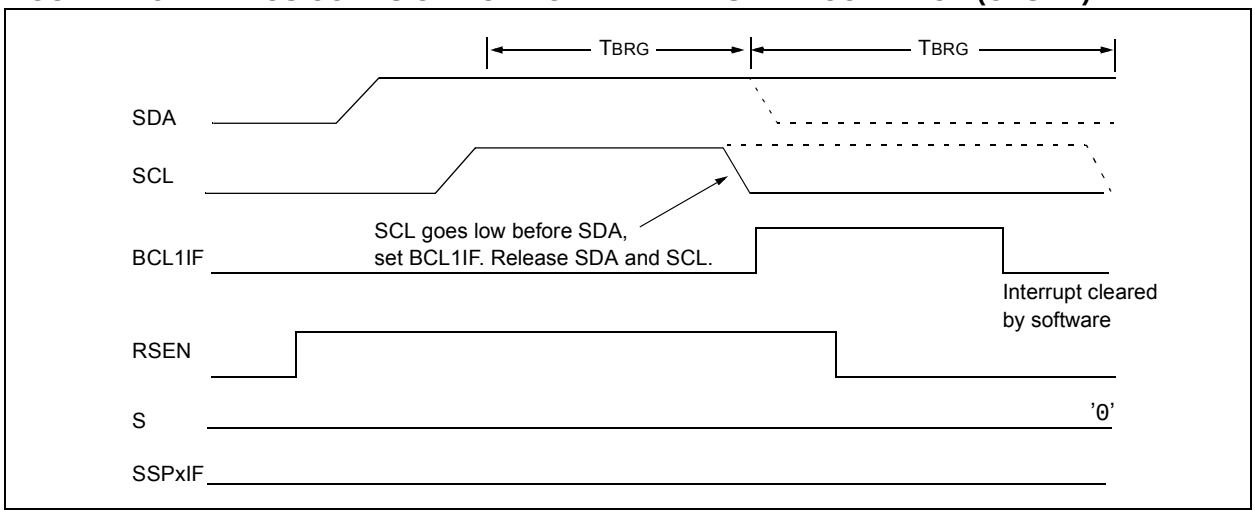


FIGURE 24-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



REGISTER 26-4: CCPRxH: CCPx HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPR<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0

MODE = Capture Mode
CCPRxH<7:0>: MSB of captured TMR1 value
MODE = Compare Mode
CCPRxH<7:0>: MSB compared to TMR1 value
MODE = PWM Mode && FMT = 0
CCPRxH<7:2>: Not used
CCPRxH<1:0>: CCPW<9:8> — Pulse width Most Significant two bits
MODE = PWM Mode && FMT = 1
CCPRxH<7:0>: CCPW<9:2> — Pulse width Most Significant eight bits

REGISTER 26-5: CCPxCAP: CCPx CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

111 = Reserved. No channel connected.
110 = Reserved. No channel connected.
101 = LC2_out
100 = LC1_out
011 = IOC_interrupt
010 = C2_OUT_sync
001 = C1_OUT_sync
000 = CCPx pin

30.6.4 HIGH AND LOW MEASURE MODE

This mode measures the high and low pulse time of the SMTSIGx relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the SMTSIGx input, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See Figure 30-8 and Figure 30-9.

FIGURE 31-4: TIMING DIAGRAM FOR SINGLE PULSE MODE

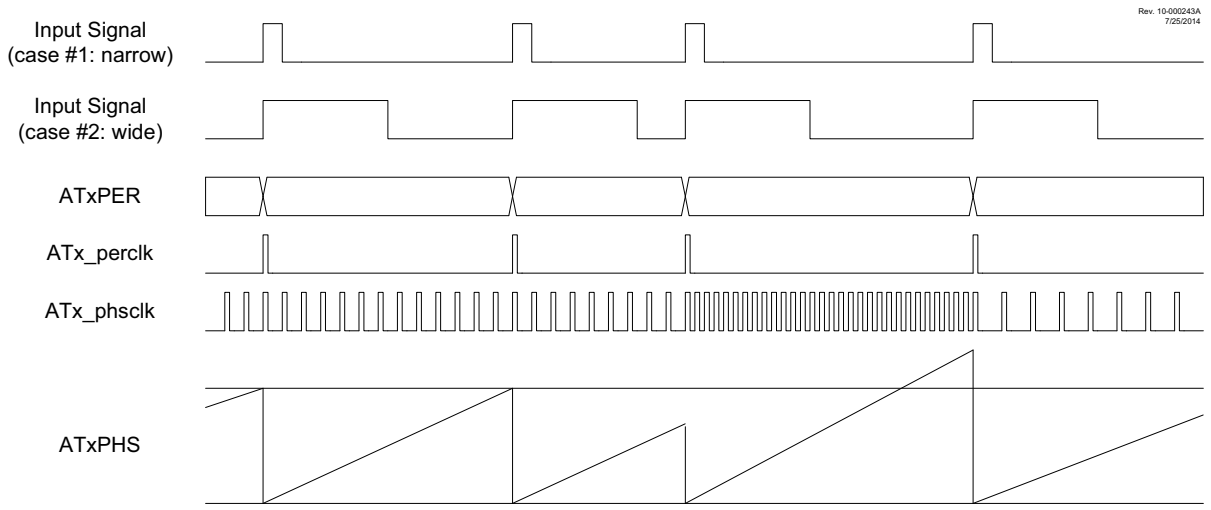
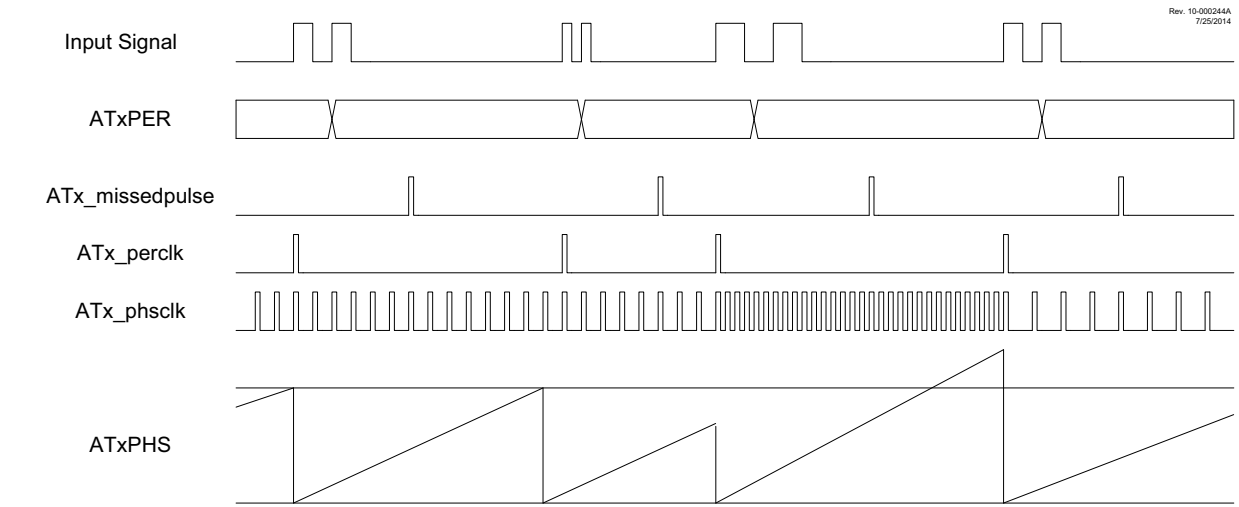


FIGURE 31-5: TIMING DIAGRAM FOR MULTI-PULSE MODE



REGISTER 32-20: PIDxZ2U: PID Z2 UPPER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	Z216
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-1 **Unimplemented:** Read as ‘0’

bit 0 **Z216:** Bit 16 of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-21: PIDxZ2H: PID Z2 HIGH REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
Z2<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-0 **Z2<15:8>:** Bits <15:8> of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-22: PIDxZ2L: PID Z2 LOW REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
Z2<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-0 **Z2<7:0>:** Bits <7:0> of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

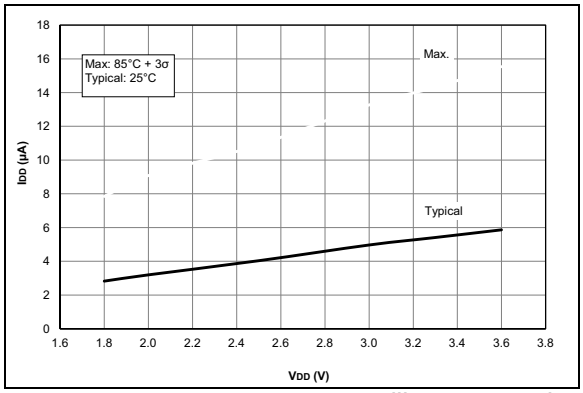


FIGURE 36-1: *IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC16LF1614/8 Only.*

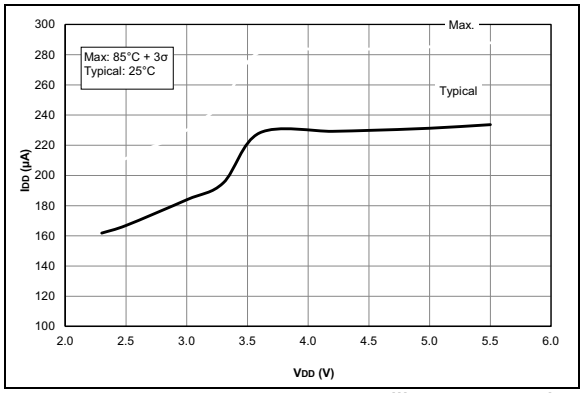


FIGURE 36-4: *IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC16F1614/8 Only.*

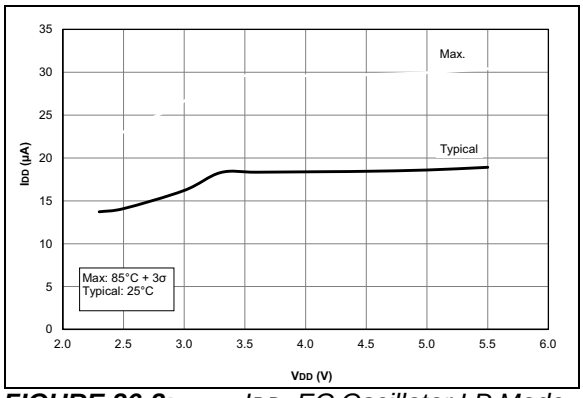


FIGURE 36-2: *IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC16F1614/8 Only.*

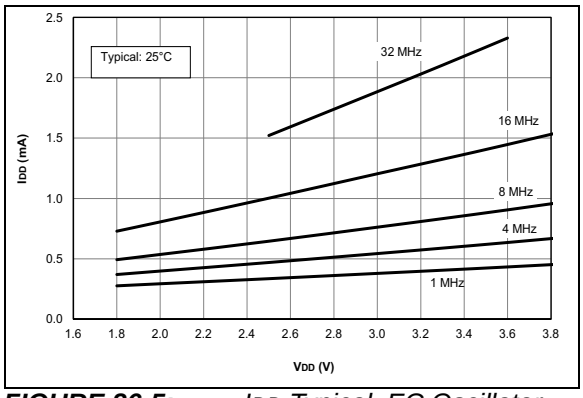


FIGURE 36-5: *IDD Typical, EC Oscillator MP Mode, PIC16LF1614/8 Only.*

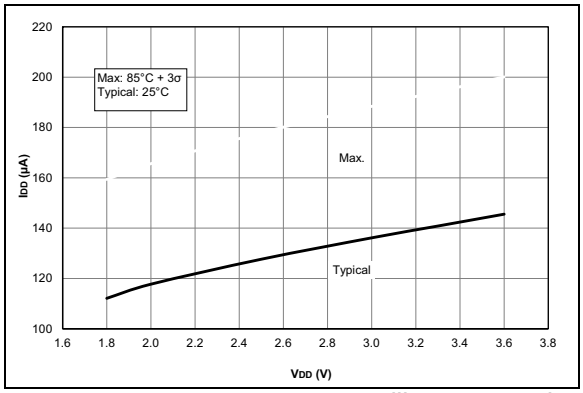


FIGURE 36-3: *IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC16LF1614/8 Only.*

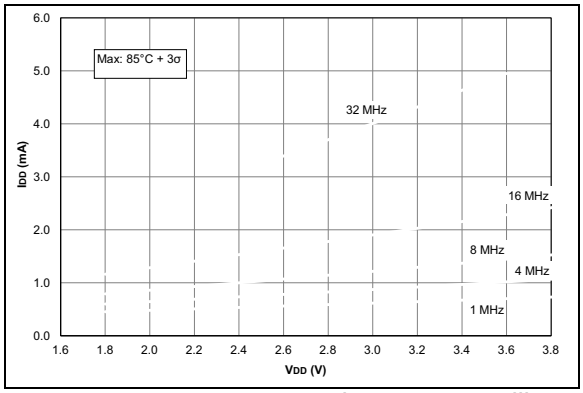


FIGURE 36-6: *IDD Maximum, EC Oscillator MP Mode, PIC16LF1614/8 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

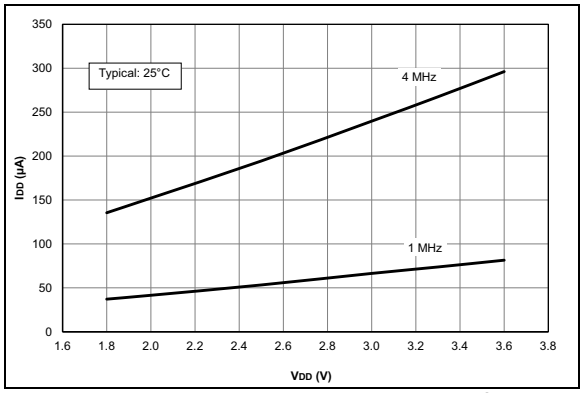


FIGURE 36-13: *IDD, LFINTOSC Mode, Fosc = 31 kHz, PIC16LF1614/8 Only.*

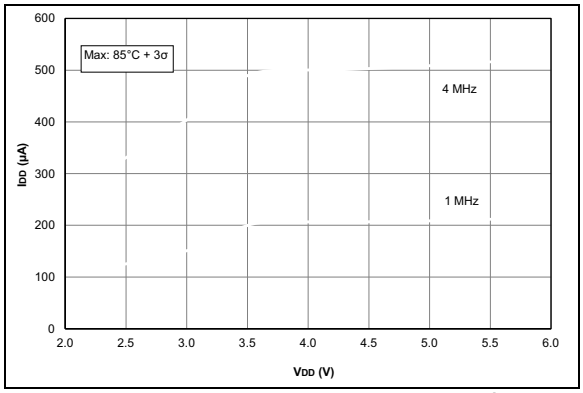


FIGURE 36-16: *IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16F1614/8 Only.*

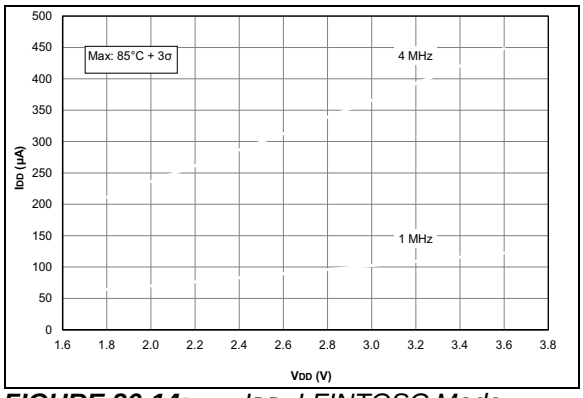


FIGURE 36-14: *IDD, LFINTOSC Mode, Fosc = 31 kHz, PIC16F1614/8 Only.*

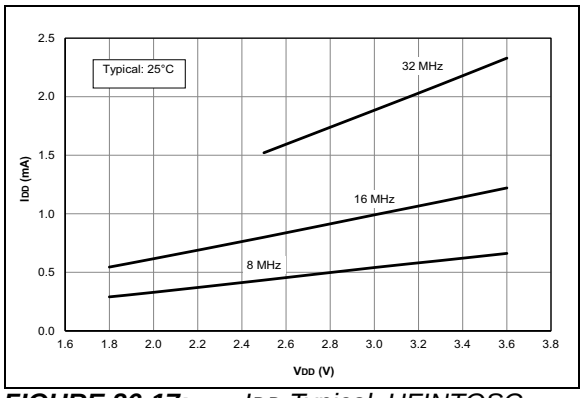


FIGURE 36-17: *IDD Typical, HFINTOSC Mode, PIC16LF1614/8 Only.*

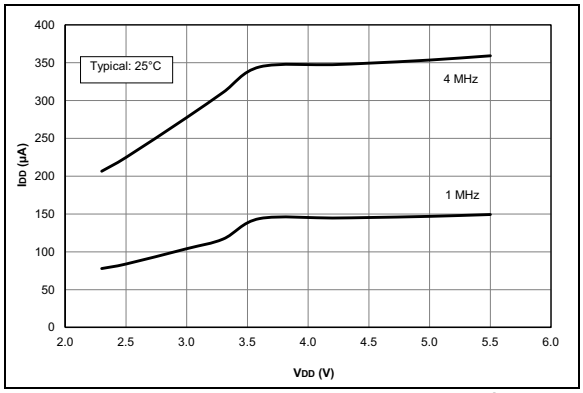


FIGURE 36-15: *IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16LF1614/8 Only.*

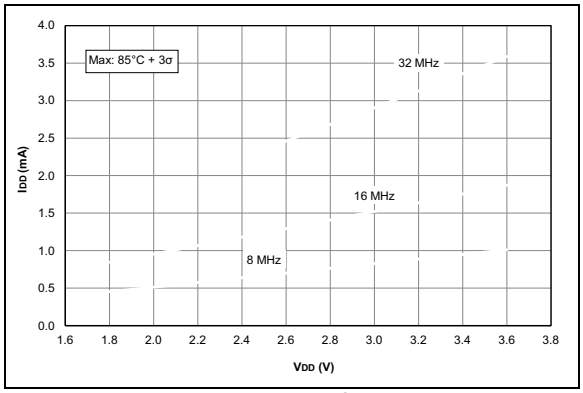


FIGURE 36-18: *IDD Maximum, HFINTOSC Mode, PIC16LF1614/8 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

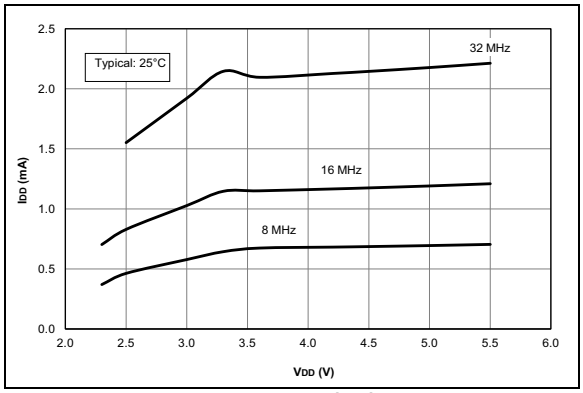


FIGURE 36-19: *IDD Typical, HFINTOSC Mode, PIC16F1614/8 Only.*

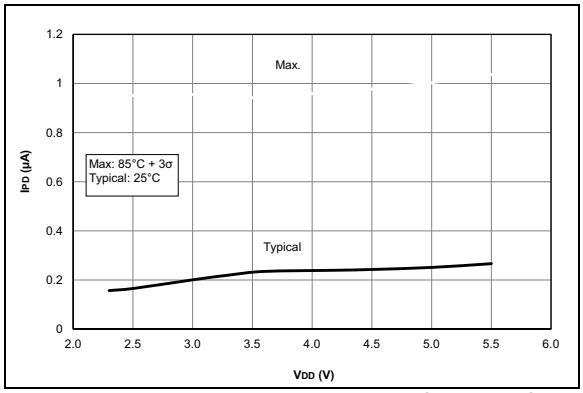


FIGURE 36-22: *IPD Base, LP Sleep Mode (VREGPM = 1), PIC16F1614/8 Only.*

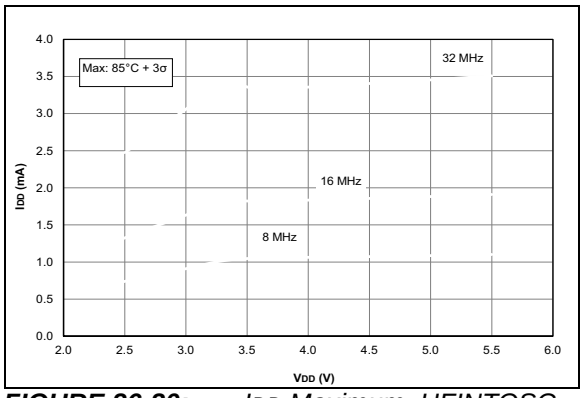


FIGURE 36-20: *IDD Maximum, HFINTOSC Mode, PIC16F1614/8 Only.*

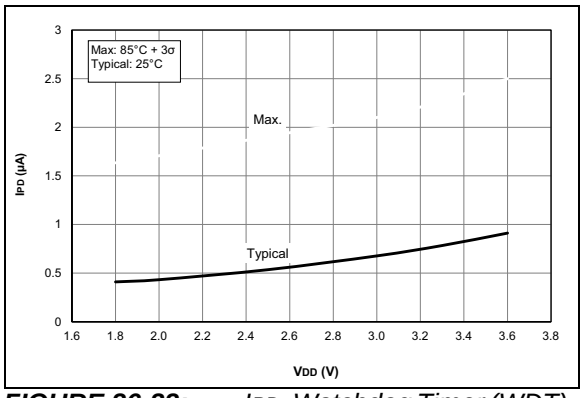


FIGURE 36-23: *IPD, Watchdog Timer (WDT), PIC16LF1614/8 Only.*

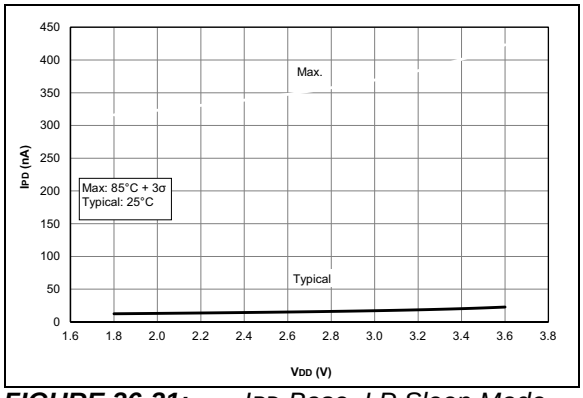


FIGURE 36-21: *IPD Base, LP Sleep Mode, PIC16LF1614/8 Only.*

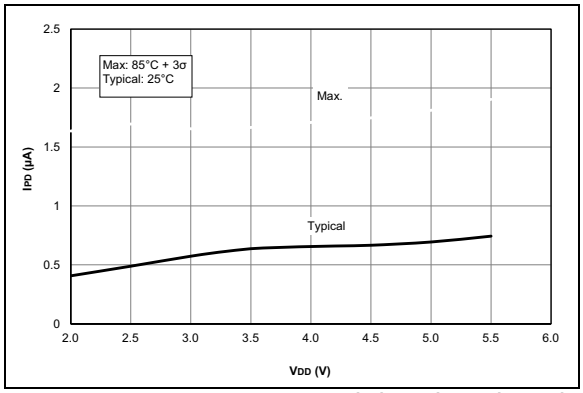


FIGURE 36-24: *IPD, Watchdog Timer (WDT), PIC16F1614/8 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

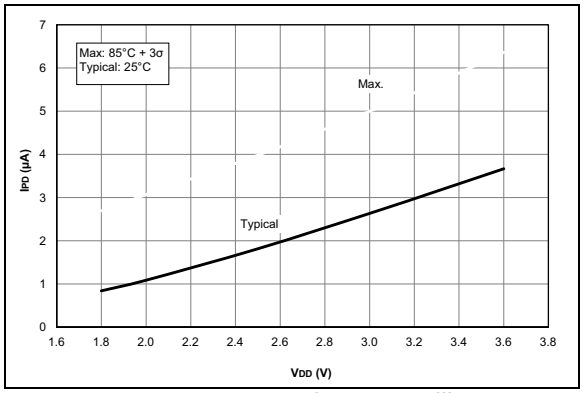


FIGURE 36-31: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16LF1614/8 Only.*

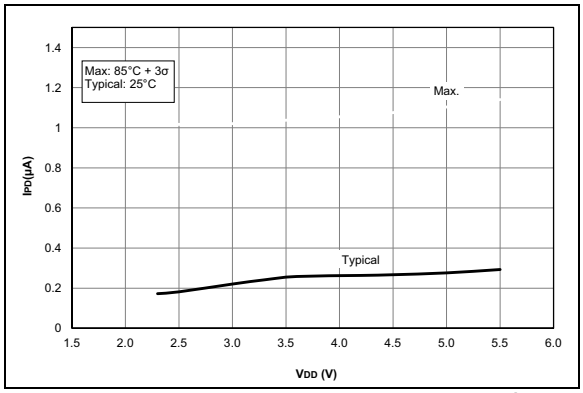


FIGURE 36-34: *IPD, ADC Non-Converting, PIC16F1614/8 Only.*

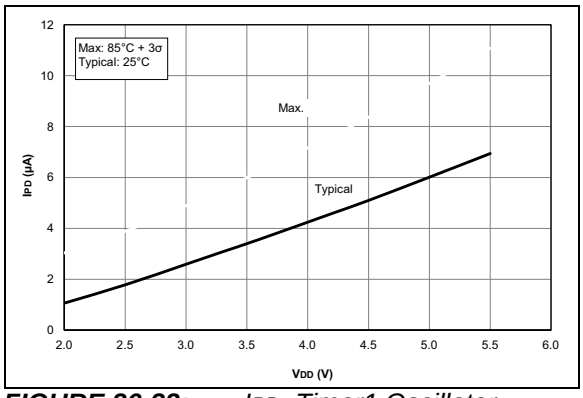


FIGURE 36-32: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16F1614/8 Only.*

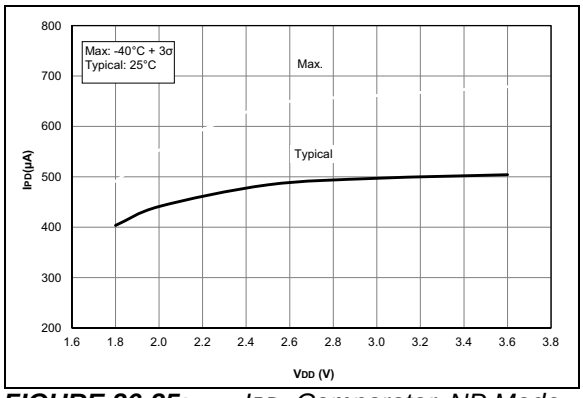


FIGURE 36-35: *IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1614/8 Only.*

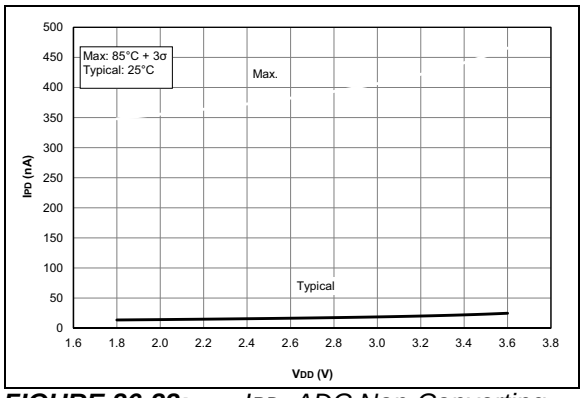


FIGURE 36-33: *IPD, ADC Non-Converting, PIC16LF1614/8 Only.*

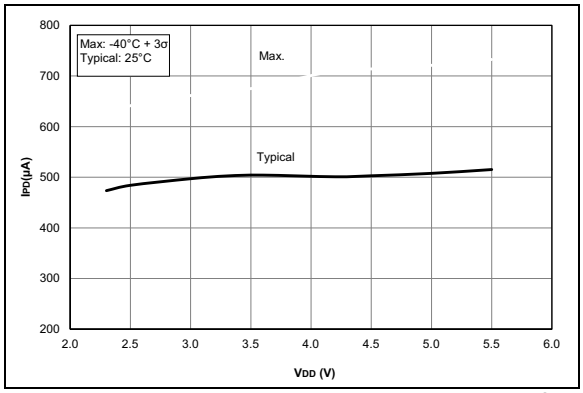
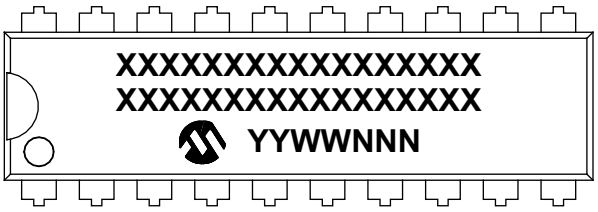


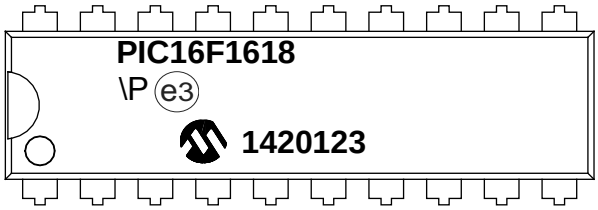
FIGURE 36-36: *IPD, Comparator, NP Mode (CxSP = 1), PIC16F1614/8 Only.*

38.1 Package Marking Information (Continued)

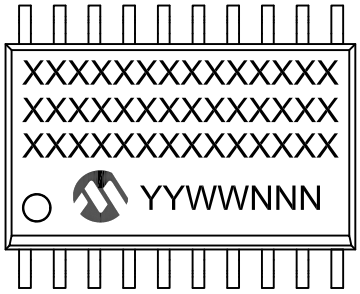
20-Lead PDIP (300 mil)



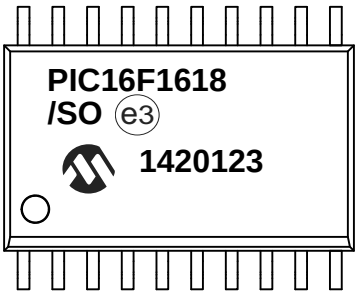
Example



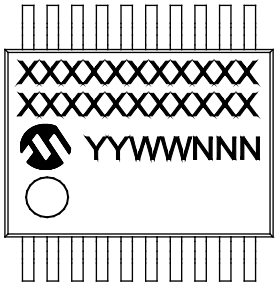
20-Lead SOIC (7.50 mm)



Example



20-Lead SSOP (5.30 mm)



Example