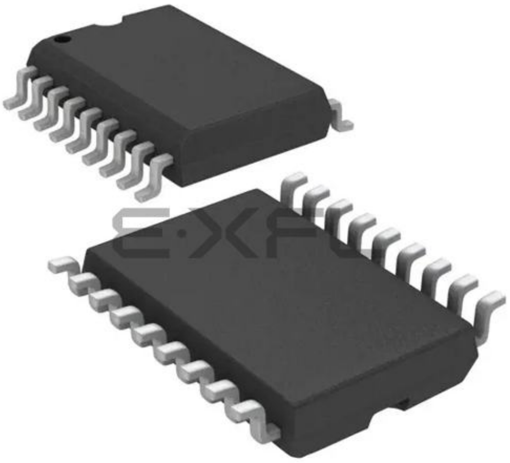
Microchip Technology - PIC18F1320-H/SO Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	25MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1320-h-so

Table of Contents

3.4.4 EXIT TO IDLE MODE

An exit from a power managed Run mode to its corresponding Idle mode is executed by setting the IDLEN bit and executing a SLEEP instruction. The CPU is halted at the beginning of the instruction following the SLEEP instruction. There are no changes to any of the clock source status bits (OSTS, IOFS or T1RUN). While the CPU is halted, the peripherals continue to be clocked from the previously selected clock source.

3.4.5 EXIT TO SLEEP MODE

An exit from a power managed Run mode to Sleep mode is executed by clearing the IDLEN and SCS1:SCS0 bits and executing a SLEEP instruction. The code is no different than the method used to invoke Sleep mode from the normal operating (full-power) mode.

The primary clock and internal oscillator block are disabled. The INTRC will continue to operate if the WDT is enabled. The Timer1 oscillator will continue to run, if enabled in the T1CON register (Register 12-1). All clock source Status bits are cleared (OSTS, IOFS and T1RUN).

3.5 Wake from Power Managed Modes

An exit from any of the power managed modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power managed modes. The clocking subsystem actions are discussed in each of the power managed modes (see Sections 3.2 through 3.4).

Note:	If application code is timing sensitive, it should wait for the OSTS bit to become set before continuing. Use the interval during the low-power exit sequence (before OSTS is set) to perform timing insensitive “housekeeping” tasks.

Device behavior during Low-Power mode exits is summarized in Table 3-3.

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit a power managed mode and resume full-power operation. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set. On all exits from Low-Power mode by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

5.3 Fast Register Stack

A “fast return” option is available for interrupts. A fast register stack is provided for the Status, WREG and BSR registers and is only one in depth. The stack is not readable or writable and is loaded with the current value of the corresponding register when the processor vectors for an interrupt. The values in the registers are then loaded back into the working registers, if the `RETFIE, FAST` instruction is used to return from the interrupt.

All interrupt sources will push values into the stack registers. If both low and high priority interrupts are enabled, the stack registers cannot be used reliably to return from low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten. Users must save the key registers in software during a low priority interrupt.

If interrupt priority is not used, all interrupts may use the fast register stack for returns from interrupt.

If no interrupts are used, the fast register stack can be used to restore the Status, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a `CALL LABEL, FAST` instruction must be executed to save the Status, WREG and BSR registers to the fast register stack. A `RETURN, FAST` instruction is then executed to restore these registers from the fast register stack.

Example 5-1 shows a source code example that uses the fast register stack during a subroutine call and return.

EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
        •
        •

SUB1    •
        •
    RETURN, FAST   ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```

5.4 PCL, PCLATH and PCLATU

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide. The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC<15:8> bits and is not directly readable or writable. Updates to the PCH register may be performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits and is not directly readable or writable. Updates to the PCU register may be performed through the PCLATU register.

The contents of PCLATH and PCLATU will be transferred to the program counter by any operation that writes PCL. Similarly, the upper two bytes of the program counter will be transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 5.8.1 “Computed GOTO”**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the LSB of PCL is fixed to a value of ‘`0`’. The PC increments by two to address sequential instructions in the program memory.

The `CALL`, `RCALL`, `GOTO` and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

REGISTER 6-1: EECON1: EEPROM CONTROL 1 REGISTER

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR[1]	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit

1 = Access program Flash memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit

1 = Accesses Configuration, User ID and Device ID Registers
0 = Accesses Flash Program or data EEPROM Memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit

1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation – TBLPTR<5:0> are ignored)
0 = Perform write only

bit 3 **WRERR:** EEPROM Error Flag bit[1]

1 = A write operation was prematurely terminated (any Reset during self-timed programming)
0 = The write operation completed normally

bit 2 **WREN:** Program/Erase Enable bit

1 = Allows program/erase cycles
0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit

1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle completed

bit 0 **RD:** Read Control bit

1 = Initiates a memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Read completed

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

```
           MOVLW   D'64                      ; number of bytes in erase block
           MOVWF   COUNTER
           MOVLW   BUFFER_ADDR_HIGH          ; point to buffer
           MOVWF   FSR0H
           MOVLW   BUFFER_ADDR_LOW
           MOVWF   FSR0L
           MOVLW   CODE_ADDR_UPPER           ; Load TBLPTR with the base
           MOVWF   TBLPTRU                   ; address of the memory block
           MOVLW   CODE_ADDR_HIGH
           MOVWF   TBLPTRH
           MOVLW   CODE_ADDR_LOW             ; 6 LSB = 0
           MOVWF   TBLPTRL
READ_BLOCK
           TBLRD*+                           ; read into TABLAT, and inc
           MOVF    TABLAT, W                 ; get data
           MOVWF   POSTINC0                  ; store data and increment FSR0
           DECFSZ  COUNTER                   ; done?
           GOTO    READ_BLOCK                ; repeat
MODIFY_WORD
           MOVLW   DATA_ADDR_HIGH            ; point to buffer
           MOVWF   FSR0H
           MOVLW   DATA_ADDR_LOW
           MOVWF   FSR0L
           MOVLW   NEW_DATA_LOW              ; update buffer word and increment FSR0
           MOVWF   POSTINC0
           MOVLW   NEW_DATA_HIGH             ; update buffer word
           MOVWF   INDF0
ERASE_BLOCK
           MOVLW   CODE_ADDR_UPPER           ; load TBLPTR with the base
           MOVWF   TBLPTRU                   ; address of the memory block
           MOVLW   CODE_ADDR_HIGH
           MOVWF   TBLPTRH
           MOVLW   CODE_ADDR_LOW             ; 6 LSB = 0
           MOVWF   TBLPTRL
           BCF     EECON1, CFGS              ; point to PROG/EEPROM memory
           BSF     EECON1, EEPGD             ; point to FLASH program memory
           BSF     EECON1, WREN              ; enable write to memory
           BSF     EECON1, FREE              ; enable Row Erase operation
           BCF     INTCON, GIE               ; disable interrupts
           MOVLW   55h                       ; Required sequence
           MOVWF   EECON2                    ; write 55H
           MOVLW   AAh
           MOVWF   EECON2                    ; write AAH
           BSF     EECON1, WR                ; start erase (CPU stall)
           NOP
           BSF     INTCON, GIE               ; re-enable interrupts
WRITE_BUFFER_BACK
           MOVLW   8                         ; number of write buffer groups of 8 bytes
           MOVWF   COUNTER_HI
           MOVLW   BUFFER_ADDR_HIGH          ; point to buffer
           MOVWF   FSR0H
           MOVLW   BUFFER_ADDR_LOW
           MOVWF   FSR0L
PROGRAM_LOOP
           MOVLW   8                         ; number of bytes in holding register
           MOVWF   COUNTER
```

REGISTER 7-1: EECON1: EEPROM CONTROL 1 REGISTER

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR[1]	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Access program Flash memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Accesses Configuration, User ID and Device ID Registers
0 = Accesses Flash Program or data EEPROM Memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation – TBLPTR<5:0> are ignored)
0 = Perform write only

bit 3 **WRERR:** EEPROM Error Flag bit[1]
1 = A write operation was prematurely terminated (any Reset during self-timed programming)
0 = The write operation completed normally

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase cycles
0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle completed

bit 0 **RD:** Read Control bit
1 = Initiates a memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Read completed

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

9.4 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Priority registers (IPR1, IPR2). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

REGISTER 9-8: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

U-0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP
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 **Unimplemented:** Read as ‘0’

bit 6 **ADIP:** A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RCIP:** EUSART Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TXIP:** EUSART Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

9.5 RCON Register

The RCON register contains bits used to determine the cause of the last Reset or wake-up from a low-power mode. RCON also contains the bit that enables interrupt priorities (IPEN).

REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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 **IPEN:** Interrupt Priority Enable bit
`1` = Enable priority levels on interrupts
`0` = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6-5 **Unimplemented:** Read as ‘`0`’

bit 4 **$\overline{\text{RI}}$:** `RESET` Instruction Flag bit
For details of bit operation, see Register 5-3.

bit 3 **$\overline{\text{TO}}$:** Watchdog Time-out Flag bit
For details of bit operation, see Register 5-3.

bit 2 **$\overline{\text{PD}}$:** Power-down Detection Flag bit
For details of bit operation, see Register 5-3.

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
For details of bit operation, see Register 5-3.

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
For details of bit operation, see Register 5-3.

16.4 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the RB1/AN5/TX/CK/INT1 and RB4/AN6/RX/DT/KBI0 I/O pins to CK (clock) and DT (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK line. Clock polarity is selected with the SCKP bit (BAUDCTL<5>); setting SCKP sets the Idle state on CK as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

16.4.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 16-2. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available).

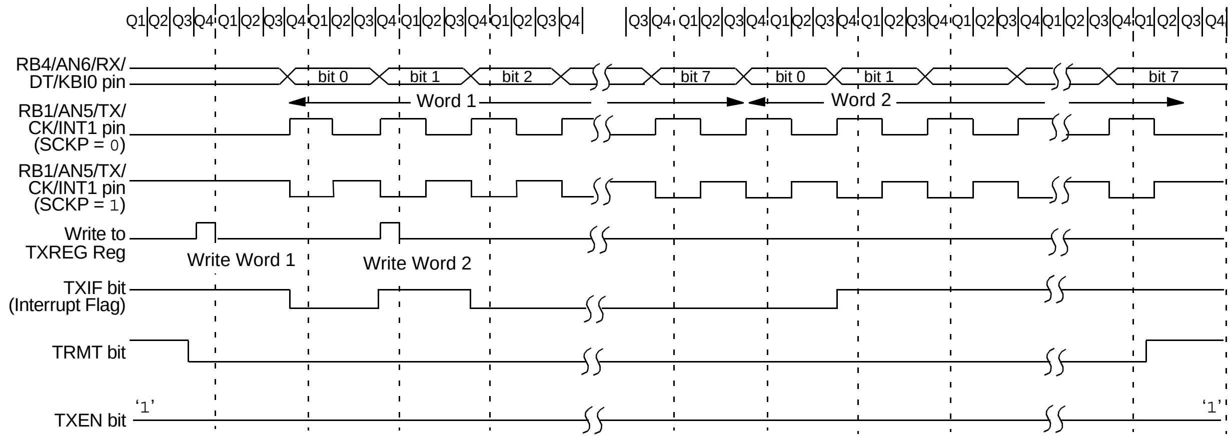
Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit, TXIF (PIR1<4>), is set. The interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit, TXIF, will be set, regardless of the state of enable bit, TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register.

While flag bit, TXIF, indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit, which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 16-10: SYNCHRONOUS TRANSMISSION


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
RB4/AN6/RX/DT/KBI0 pin
bit 0
bit 1
bit 2
bit 7
bit 0
bit 1
bit 7
Word 1
Word 2
RB1/AN5/TX/CK/INT1 pin (SCKP = 0)
RB1/AN5/TX/CK/INT1 pin (SCKP = 1)
Write to TXREG Reg
Write Word 1
Write Word 2
TXIF bit (Interrupt Flag)
TRMT bit
TXEN bit
'1'
'1'

Note: Sync Master mode, SPBRG = 0, continuous transmission of two 8-bit words.

TABLE 19-2: SUMMARY OF WATCHDOG TIMER REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS2	WDTPS0	WDTEN
RCON	IPEN	—	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
WDTCON	—	—	—	—	—	—	—	SWDTEN

Legend: Shaded cells are not used by the Watchdog Timer.

19.3 Two-Speed Start-up

The Two-Speed Start-up feature helps to minimize the latency period from oscillator start-up to code execution by allowing the microcontroller to use the INTRC oscillator as a clock source until the primary clock source is available. It is enabled by setting the IESO bit in Configuration Register 1H (CONFIG1H<7>).

Two-Speed Start-up is available only if the primary oscillator mode is LP, XT, HS or HSPLL (crystal-based modes). Other sources do not require an OST start-up delay; for these, Two-Speed Start-up is disabled.

When enabled, Resets and wake-ups from Sleep mode cause the device to configure itself to run from the internal oscillator block as the clock source, following the time-out of the Power-up Timer after a Power-on Reset is enabled. This allows almost immediate code execution while the primary oscillator starts and the OST is running. Once the OST times out, the device automatically switches to PRI_RUN mode.

Because the OSCCON register is cleared on Reset events, the INTOSC (or postscaler) clock source is not initially available after a Reset event; the INTRC clock is used directly at its base frequency. To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IFRC2:IFRC0, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting IFRC2:IFRC0 prior to entering Sleep mode.

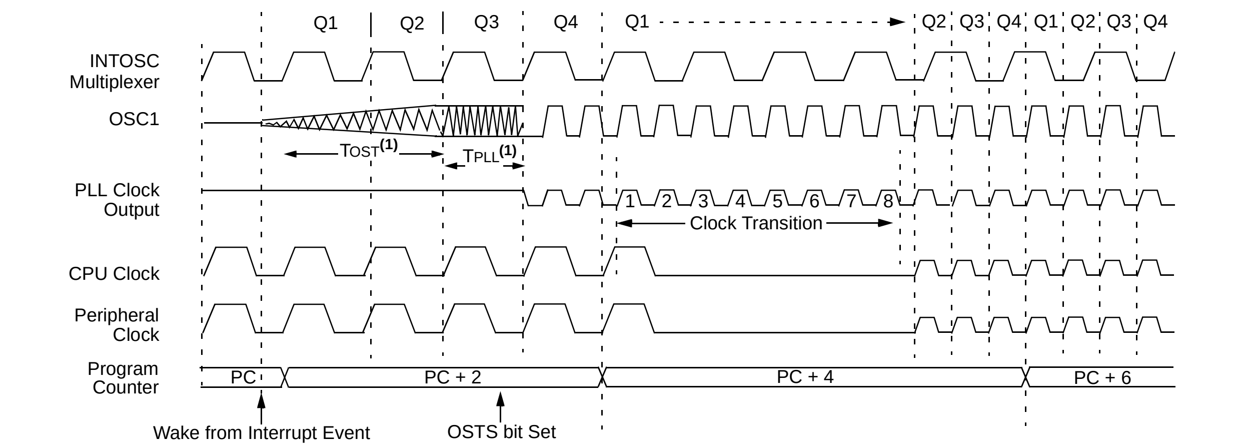
In all other power managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

19.3.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTRC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power managed modes, including serial SLEEP instructions (refer to **Section 3.1.3 "Multiple Sleep Commands"**). In practice, this means that user code can change the SCS1:SCS0 bit settings and issue SLEEP commands before the OST times out. This would allow an application to briefly wake-up, perform routine "housekeeping" tasks and return to Sleep before the device starts to operate from the primary oscillator.

User code can also check if the primary clock source is currently providing the system clocking by checking the status of the OSTS bit (OSCCON<3>). If the bit is set, the primary oscillator is providing the system clock. Otherwise, the internal oscillator block is providing the clock during wake-up from Reset or Sleep mode.

FIGURE 19-2: TIMING TRANSITION FOR TWO-SPEED START-UP (INTOSC TO HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

19.4.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset, or by entering a power managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INTOSC multiplexer provides the system clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock system source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power managed mode is entered.

Entering a power managed mode by loading the OSCCON register and executing a SLEEP instruction will clear the Fail-Safe condition. When the Fail-Safe condition is cleared, the clock monitor will resume monitoring the peripheral clock.

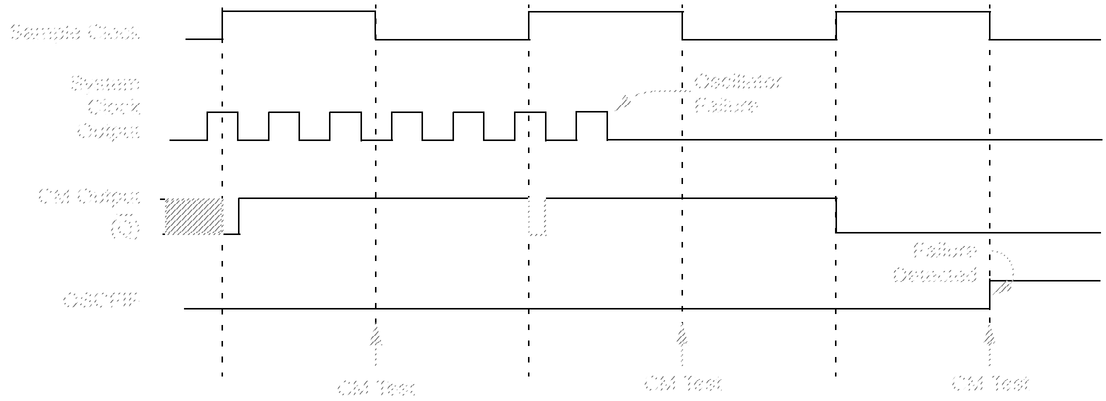
19.4.3 FSCM INTERRUPTS IN POWER MANAGED MODES

As previously mentioned, entering a power managed mode clears the Fail-Safe condition. By entering a power managed mode, the clock multiplexer selects the clock source selected by the OSCCON register. Fail-Safe monitoring of the power managed clock source resumes in the power managed mode.

If an oscillator failure occurs during power managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, the device will not exit the power managed mode on oscillator failure. Instead, the device will continue to operate as before, but clocked by the INTOSC multiplexer. While in Idle mode, subsequent interrupts will cause the CPU to begin executing instructions while being clocked by the INTOSC multiplexer. The device will not transition to a different clock source until the Fail-Safe condition is cleared.

FIGURE 19-4: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

NEGF **Negate f**

Syntax: [*label*] NEGF f [,a]

Operands: $0 \leq f \leq 255$
$a \in [0,1]$

Operation: $(\overline{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

0110	110a	ffff	ffff

Description: Location ‘f’ is negated using two’s complement. The result is placed in the data memory location ‘f’. If ‘a’ is ‘0’, the Access Bank will be selected, overriding the BSR value. If ‘a’ = 1, then the bank will be selected as per the BSR value.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: `NEGF REG, 1`

Before Instruction
```
REG    =   0011 1010 [0x3A]
```
After Instruction
```
REG    =   1100 0110 [0xC6]
```

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000 1111	0000 xxxx	0000 xxxx	0000 xxxx

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

TBLWT Table Write

Syntax: [*label*] TBLWT (*; *+; *-; +*)

Operands: None

Operation:
if TBLWT*,
(TABLAT) → Holding Register;
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register;
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register;
(TBLPTR) – 1 → TBLPTR;
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register;

Status Affected: None

Encoding:

0000	0000	0000	11nn nn = 0* = 1*+ = 2*- = 3+*

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the eight holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 "Flash Program Memory"** for additional details on programming Flash memory.)
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0:Least Significant Byte of Program Memory Word

TBLPTR[0] = 1:Most Significant Byte of Program Memory Word

The `TBLWT` instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

TBLWT Table Write (Continued)

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

Example 1: `TBLWT *+;`

Before Instruction

TABLAT	=	0x55
TBLPTR	=	0x00A356
HOLDING REGISTER (0x00A356)	=	0xFF

After Instructions (table write completion)

TABLAT	=	0x55
TBLPTR	=	0x00A357
HOLDING REGISTER (0x00A356)	=	0x55

Example 2: `TBLWT +*;`

Before Instruction

TABLAT	=	0x34
TBLPTR	=	0x01389A
HOLDING REGISTER (0x01389A)	=	0xFF
HOLDING REGISTER (0x01389B)	=	0xFF

After Instruction (table write completion)

TABLAT	=	0x34
TBLPTR	=	0x01389B
HOLDING REGISTER (0x01389A)	=	0xFF
HOLDING REGISTER (0x01389B)	=	0x34

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial) (Continued)

PIC18LF1220/1320 (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F1220/1320 (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (IDD)[2,3]						
	PIC18LF1220/1320	9.2	15	µA	-10°C	VDD = 2.0V	FOSC = 32 kHz[4] (**SEC_IDLE** mode, Timer1 as clock)
		9.6	15	µA	+25°C		
		12.7	18	µA	+70°C		
	PIC18LF1220/1320	22	30	µA	-10°C	VDD = 3.0V	
		21	30	µA	+25°C		
		20	35	µA	+70°C		
	All devices	50	80	µA	-10°C	VDD = 5.0V	
		45	80	µA	+25°C		
		45	80	µA	+70°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 VDD or VSS 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 IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

22.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 22-6: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)

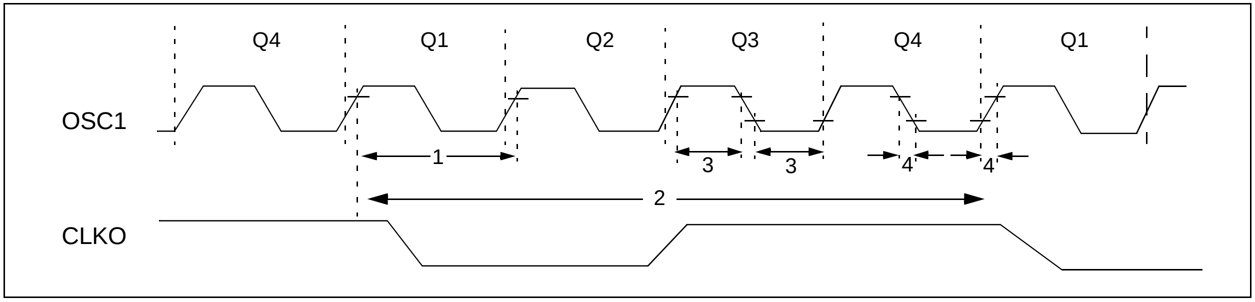


TABLE 22-4: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
1A	FOSC	External CLKI Frequency[1]	DC	40	MHz	EC, ECIO (LF and Industrial)
			DC	25	MHz	EC, ECIO (Extended)
		Oscillator Frequency[1]	DC	4	MHz	RC oscillator
			DC	1	MHz	XT oscillator
			DC	25	MHz	HS oscillator
			1	10	MHz	HS + PLL oscillator
			DC	33	kHz	LP Oscillator mode
1	TOSC	External CLKI Period[1]	25	—	ns	EC, ECIO (LF and Industrial)
			40	—	ns	EC, ECIO (Extended)
		Oscillator Period[1]	250	—	ns	RC oscillator
			1000	—	ns	XT oscillator
			25	—	ns	HS oscillator
			100	1000	ns	HS + PLL oscillator
			30	—	μs	LP oscillator
2	TCY	Instruction Cycle Time[1]	100	—	ns	TCY = 4/FOSC
3	TosL, TosH	External Clock in (OSC1) High or Low Time	30	—	ns	XT oscillator
			2.5	—	μs	LP oscillator
			10	—	ns	HS oscillator
4	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	20	ns	XT oscillator
			—	50	ns	LP oscillator
			—	7.5	ns	HS oscillator

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions, with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

FIGURE 23-11: TYPICAL IDD vs. FOSC OVER VDD PRI_IDLE, EC MODE, +25°C

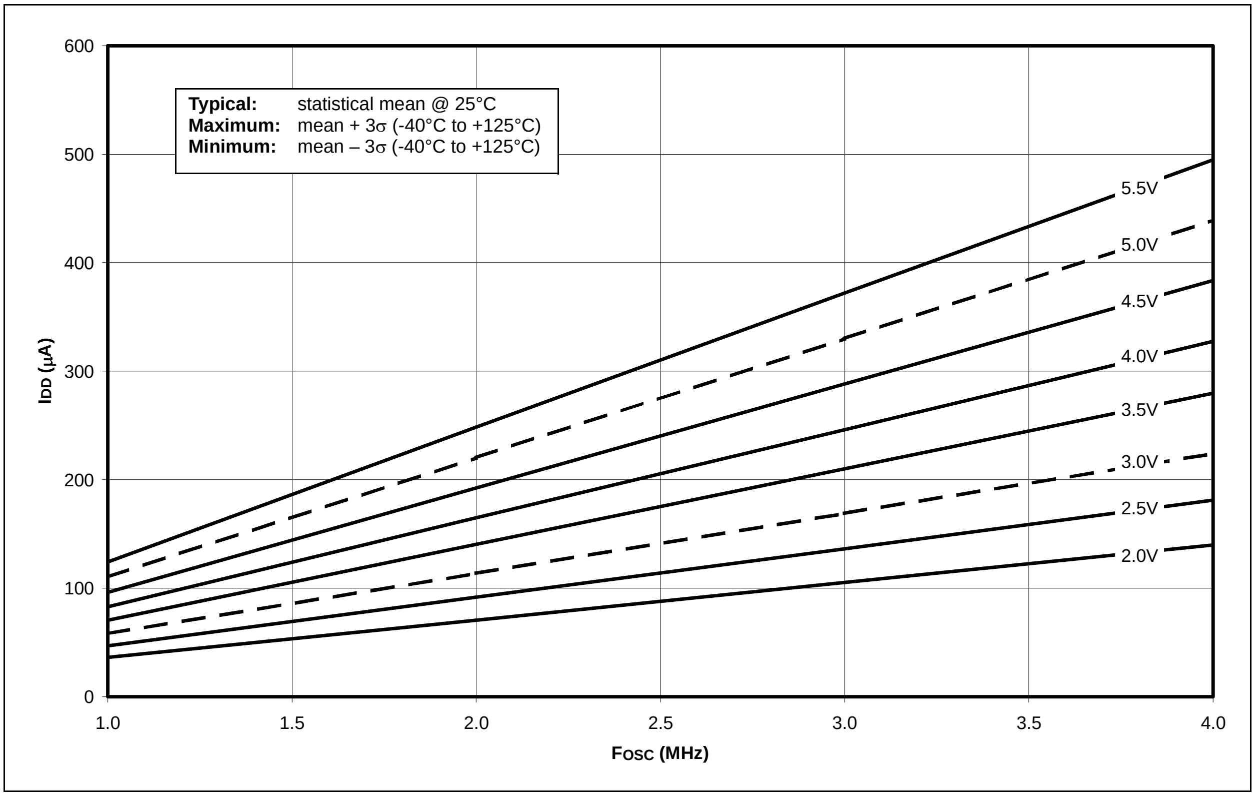


FIGURE 23-12: MAXIMUM IDD vs. FOSC OVER VDD PRI_IDLE, EC MODE, -40°C TO +125°C

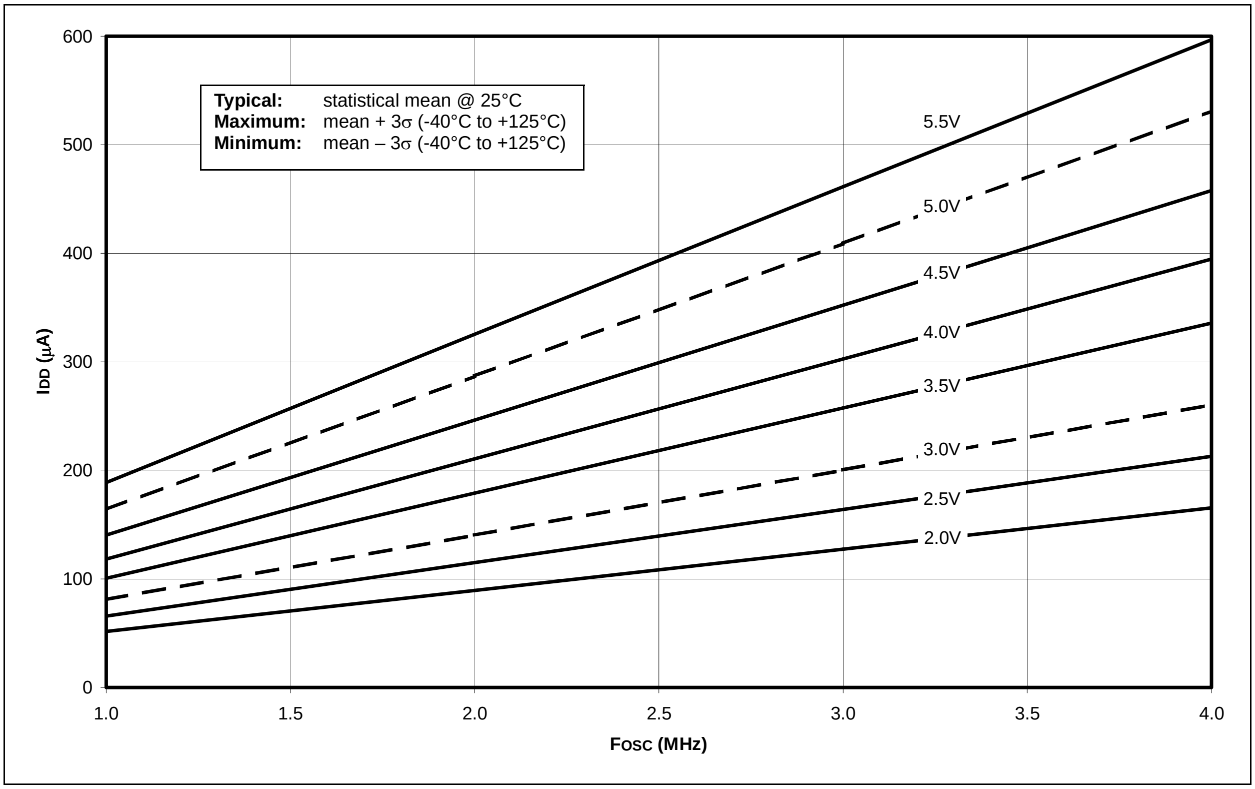


FIGURE 23-15: TYPICAL IPD vs. VDD (+25°C), 125 kHz TO 8 MHz RC_RUN MODE, ALL PERIPHERALS DISABLED

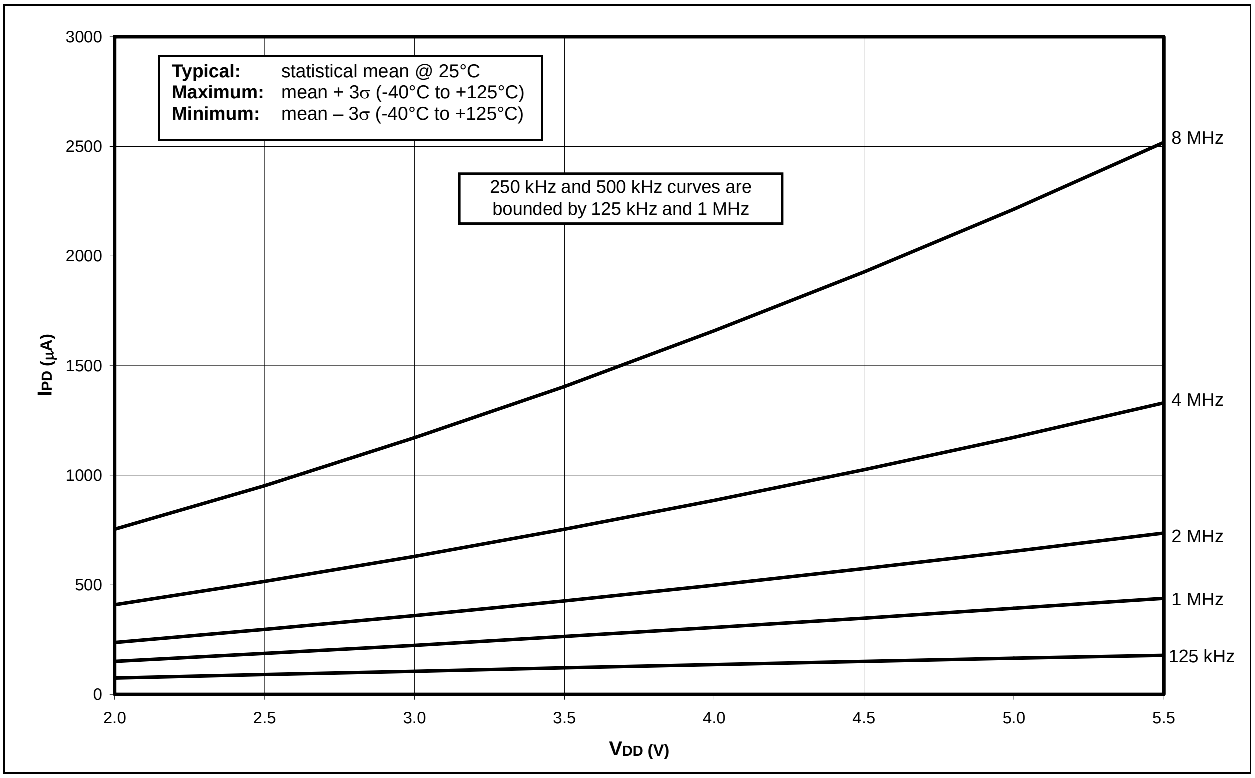


FIGURE 23-16: MAXIMUM IPD vs. VDD (-40°C TO +125°C), 125 kHz TO 8 MHz RC_RUN MODE, ALL PERIPHERALS DISABLED

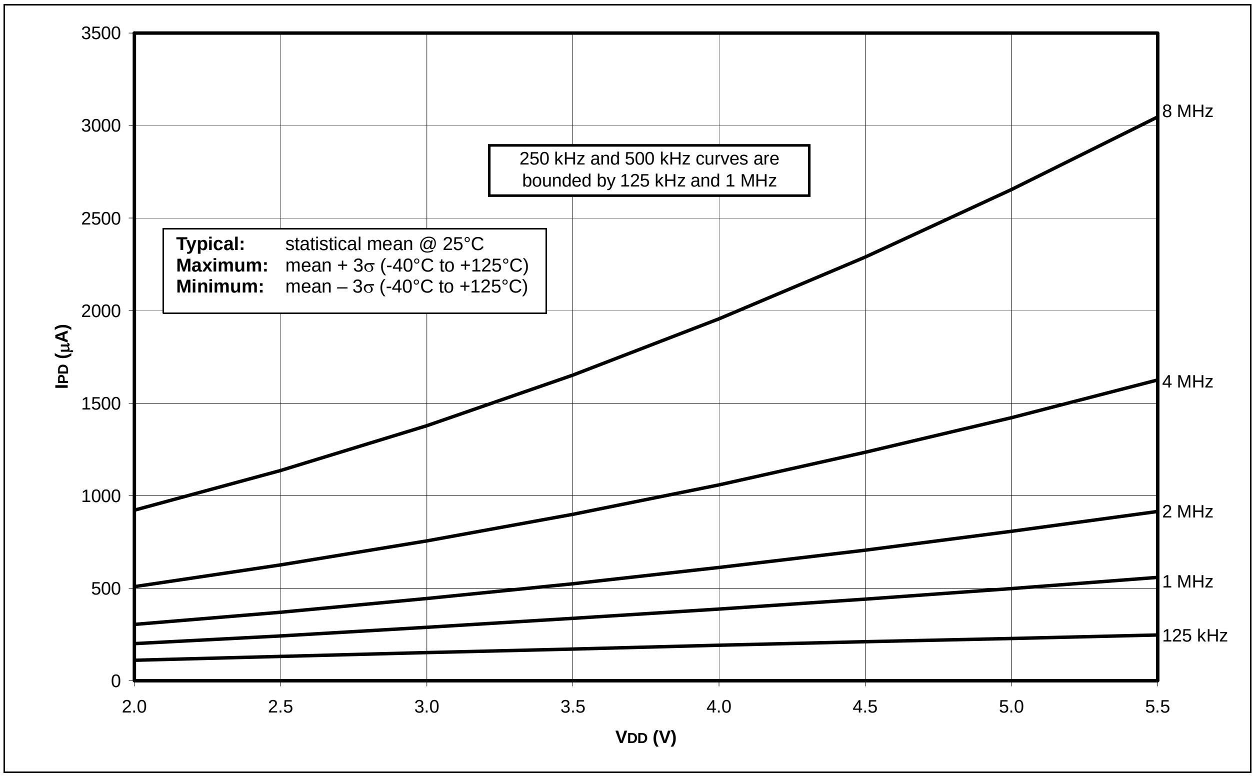
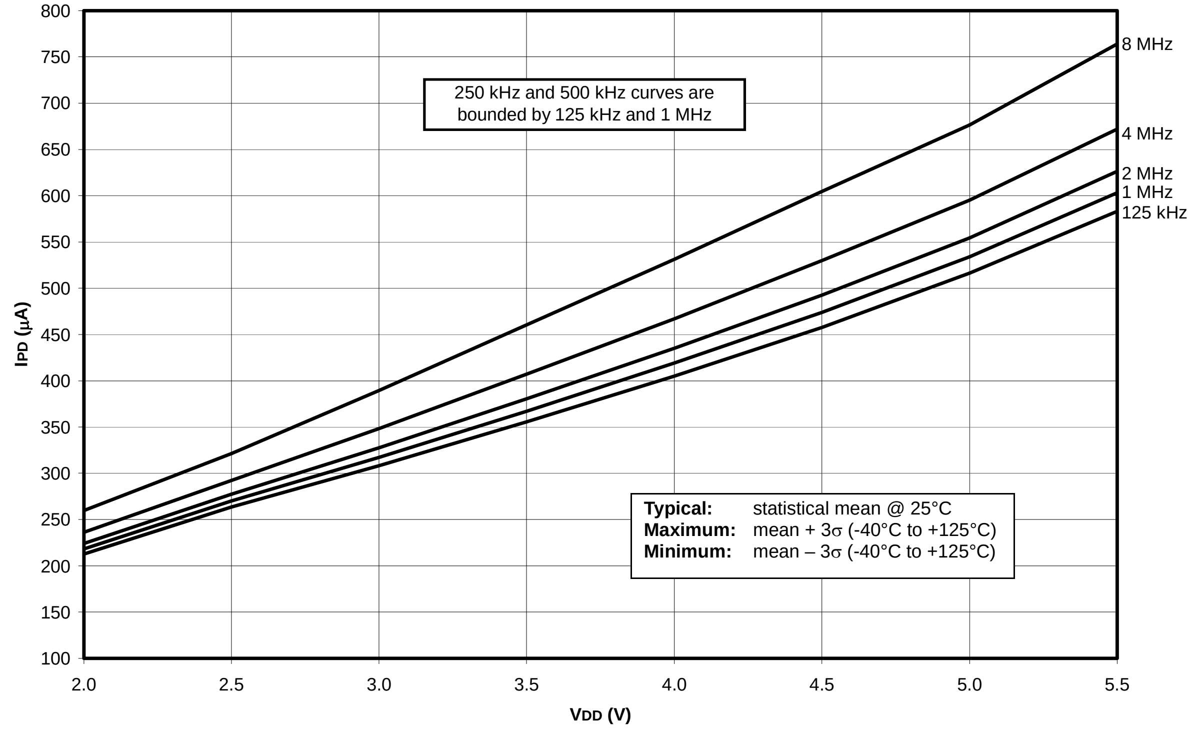


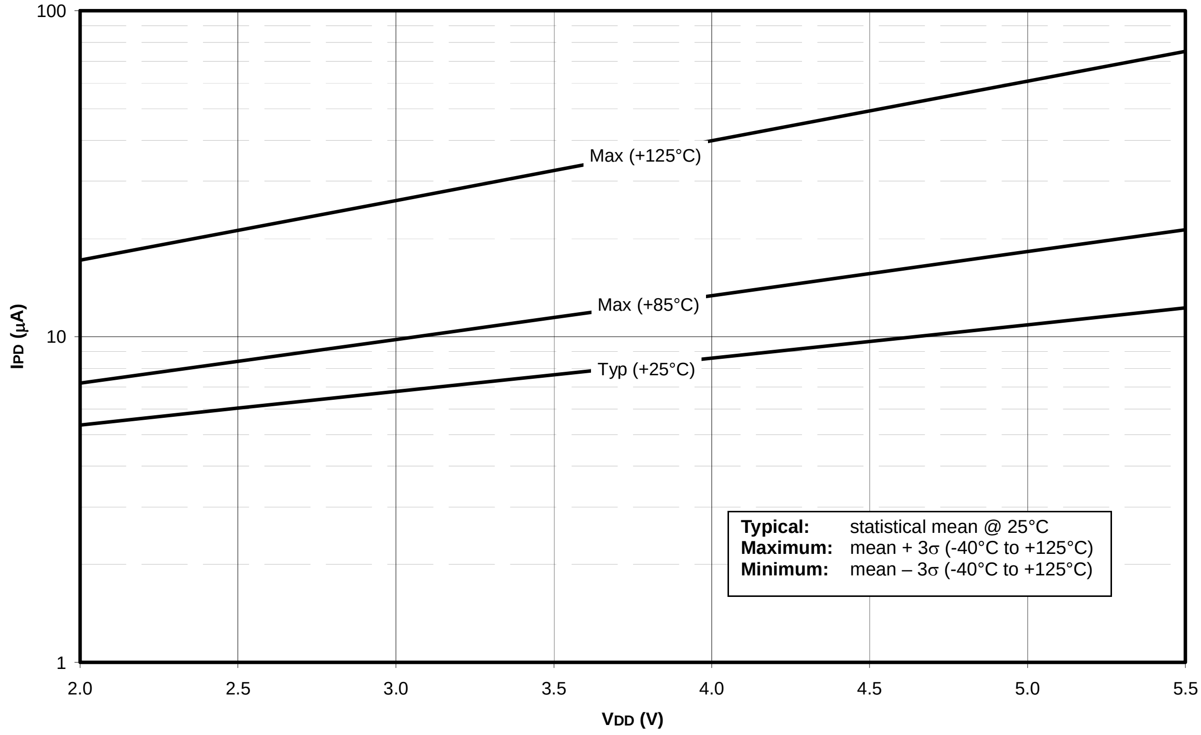
FIGURE 23-19: MAXIMUM IPD vs. VDD (-40°C TO +125°C), 125 kHz TO 8 MHz RC_IDLE MODE, ALL PERIPHERALS DISABLED



250 kHz and 500 kHz curves are bounded by 125 kHz and 1 MHz

Typical: statistical mean @ 25°C
Maximum: mean + 3σ (-40°C to +125°C)
Minimum: mean – 3σ (-40°C to +125°C)

FIGURE 23-20: TYPICAL AND MAXIMUM IPD vs. VDD (-40°C TO +125°C), 31.25 kHz RC_IDLE MODE, ALL PERIPHERALS DISABLED



Typical: statistical mean @ 25°C
Maximum: mean + 3σ (-40°C to +125°C)
Minimum: mean – 3σ (-40°C to +125°C)

FIGURE 23-31: ΔIPD LVD vs. VDD SLEEP MODE, LVDL3:LVDL0 = 0001 (2V)

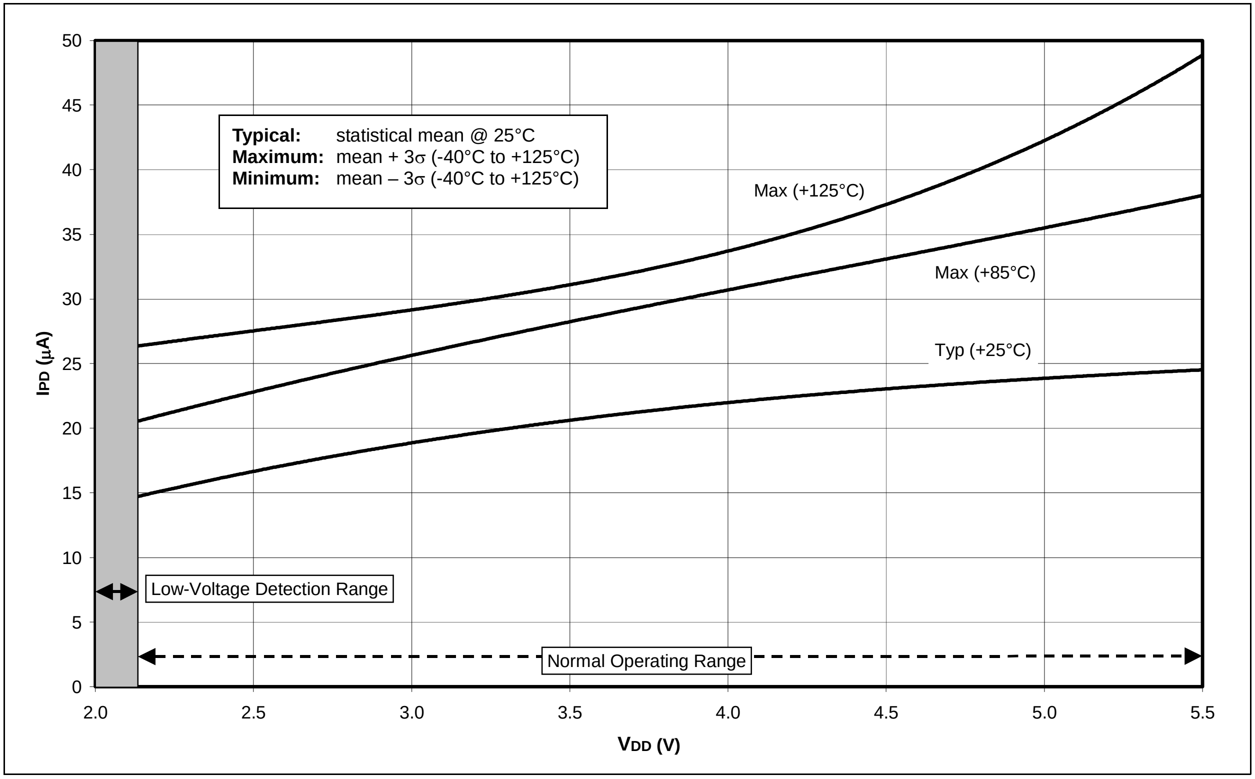
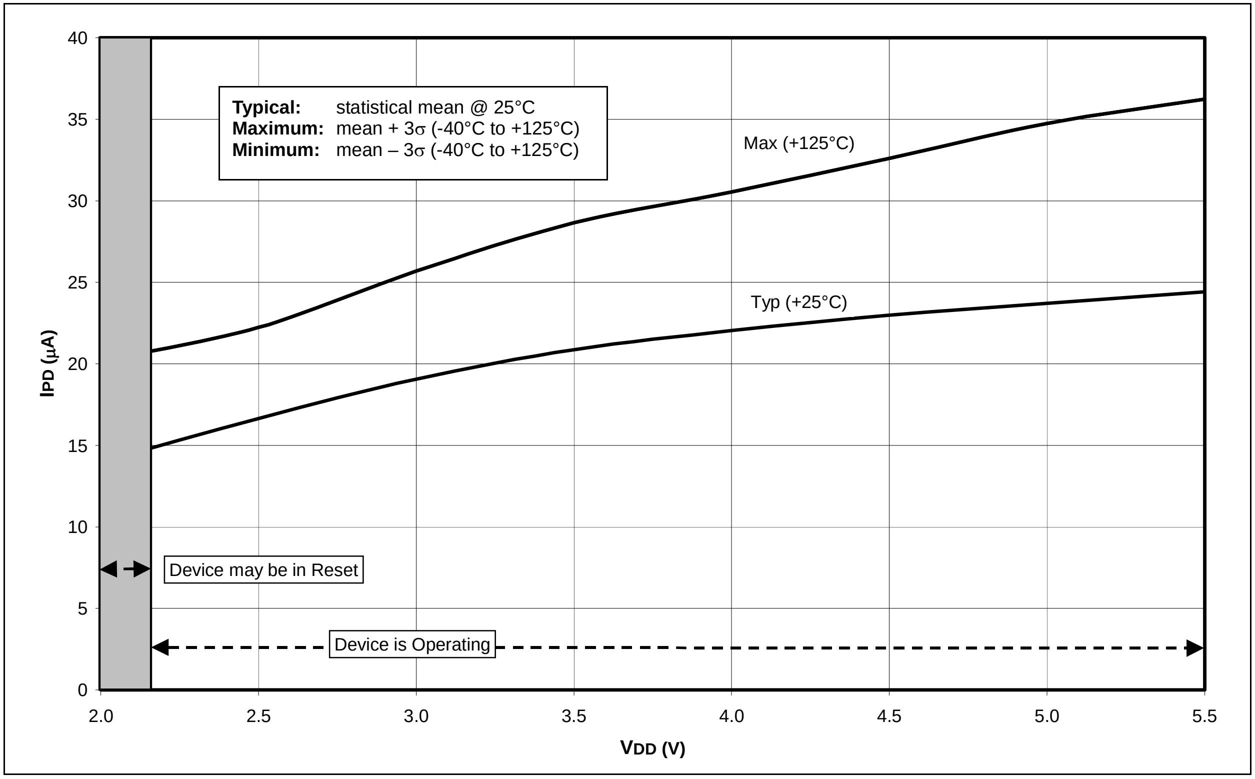


FIGURE 23-32: ΔIPD BOR vs. VDD, -40°C TO +125°C SLEEP MODE, BORV1:BORV0 = 11 (2V)



APPENDIX A: REVISION HISTORY

Revision A (August 2002)

Original data sheet for PIC18F1220/1320 devices.

Revision B (November 2002)

This revision includes significant changes to **Section 2.0**, **Section 3.0** and **Section 19.0**, as well as updates to the Electrical Specifications in **Section 22.0** and includes minor corrections to the data sheet text.

Revision C (May 2004)

This revision includes updates to the Electrical Specifications in **Section 22.0**, the DC and AC Characteristics Graphs and Tables in **Section 23.0** and includes minor corrections to the data sheet text.

Revision D (October 2006)

This revision includes updates to the packaging diagrams.

Revision E (January 2007)

This revision includes updates to the packaging diagrams.

Revision F (February 2007)

This revision includes updates to the packaging diagrams.

Revision G (April 2015)

Added Section 22.5: High Temperature Operation in the Electrical Specifications section.

APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1.

TABLE B-1: DEVICE DIFFERENCES

Features	PIC18F1220	PIC18F1320
Program Memory (Bytes)	4096	8192
Program Memory (Instructions)	2048	4096
Interrupt Sources	15	15
I/O Ports	Ports A, B	Ports A, B
Enhanced Capture/Compare/PWM Modules	1	1
10-bit Analog-to-Digital Module	7 input channels	7 input channels
Packages	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN