



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	40MHz
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)	2V ~ 5.5V
Data Converters	A/D 7x10b
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/pic18lf1320-i-p

2.7.1 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 2-2) controls several aspects of the system clock's operation, both in full-power operation and in power managed modes.

The System Clock Select bits, SCS1:SCS0, select the clock source that is used when the device is operating in power managed modes. The available clock sources are the primary clock (defined in Configuration Register 1H), the secondary clock (Timer1 oscillator) and the internal oscillator block. The clock selection has no effect until a **SLEEP** instruction is executed and the device enters a power managed mode of operation. The SCS bits are cleared on all forms of Reset.

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, the INTOSC source (8 MHz), or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). If the internal oscillator block is supplying the system clock, changing the states of these bits will have an immediate change on the internal oscillator's output.

The OSTS, IOFS and T1RUN bits indicate which clock source is currently providing the system clock. The OSTS indicates that the Oscillator Start-up Timer has timed out and the primary clock is providing the system clock in Primary Clock modes. The IOFS bit indicates

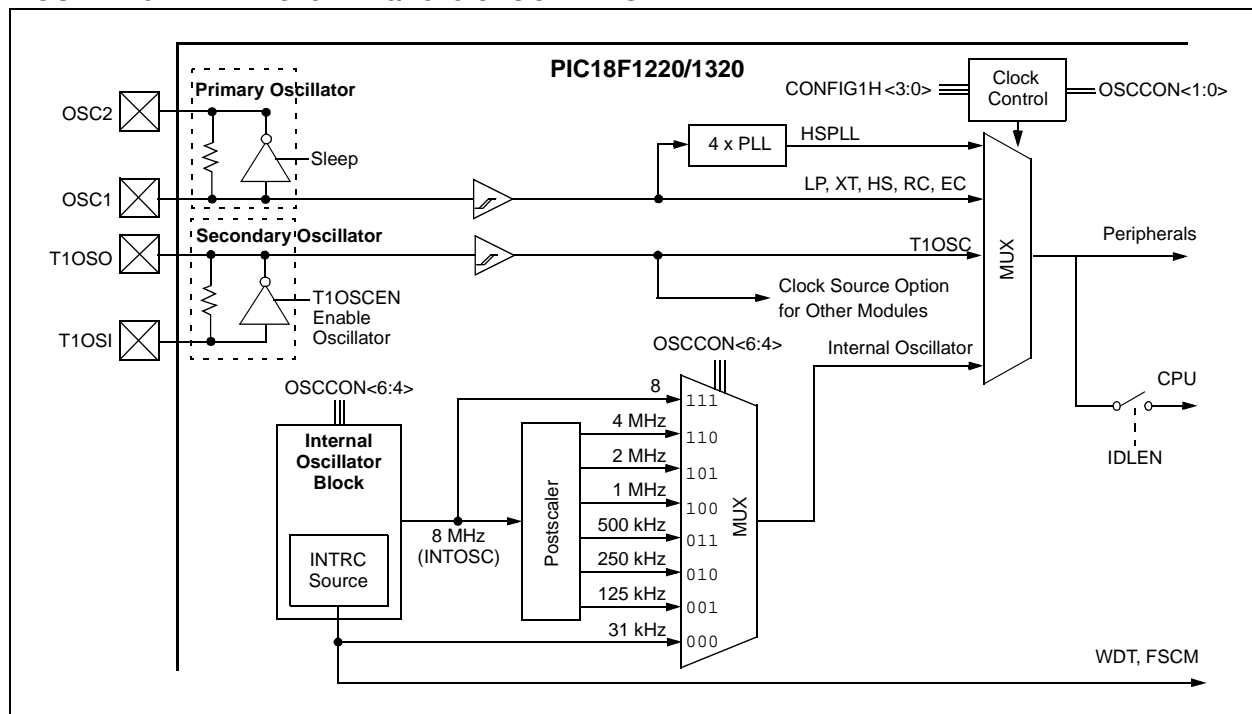
when the internal oscillator block has stabilized and is providing the system clock in RC Clock modes or during Two-Speed Start-ups. The T1RUN bit (T1CON<6>) indicates when the Timer1 oscillator is providing the system clock in Secondary Clock modes. In power managed modes, only one of these three bits will be set at any time. If none of these bits are set, the INTRC is providing the system clock, or the internal oscillator block has just started and is not yet stable.

The IDLEN bit controls the selective shutdown of the controller's CPU in power managed modes. The uses of these bits are discussed in more detail in **Section 3.0 "Power Managed Modes"**.

Note 1: The Timer1 oscillator must be enabled to select the secondary clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select a secondary clock source when executing a **SLEEP** instruction will be ignored.

2: It is recommended that the Timer1 oscillator be operating and stable before executing the **SLEEP** instruction or a very long delay may occur while the Timer1 oscillator starts.

FIGURE 2-8: PIC18F1220/1320 CLOCK DIAGRAM



5.2 Return Address Stack

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a `CALL` or `RCALL` instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a `RETURN`, `RETLW` or a `RETFIE` instruction. `PCLATU` and `PCLATH` are not affected by any of the `RETURN` or `CALL` instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, with the Stack Pointer initialized to `00000B` after all Resets. There is no RAM associated with Stack Pointer, `00000B`. This is only a Reset value. During a `CALL` type instruction, causing a push onto the stack, the Stack Pointer is first incremented and the RAM location pointed to by the Stack Pointer (`STKPTR`) register is written with the contents of the PC (already pointing to the instruction following the `CALL`). During a `RETURN` type instruction, causing a pop from the stack, the contents of the RAM location pointed to by the `STKPTR` are transferred to the PC and then the Stack Pointer is decremented.

The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the top-of-stack Special File Registers. Data can also be pushed to or popped from the stack using the top-of-stack SFRs. Status bits indicate if the stack is full, has overflowed or underflowed.

5.2.1 TOP-OF-STACK ACCESS

The top of the stack is readable and writable. Three register locations, `TOSU`, `TOSH` and `TOSL`, hold the contents of the stack location pointed to by the `STKPTR` register (Figure 5-3). This allows users to implement a software stack if necessary. After a `CALL`, `RCALL` or interrupt, the software can read the pushed value by reading the `TOSU`, `TOSH` and `TOSL` registers. These values can be placed on a user defined software stack. At return time, the software can replace the `TOSU`, `TOSH` and `TOSL` and do a return.

The user must disable the Global Interrupt Enable bits while accessing the stack to prevent inadvertent stack corruption.

5.2.2 RETURN STACK POINTER (STKPTR)

The `STKPTR` register (Register 5-1) contains the Stack Pointer value, the `STKFUL` (Stack Full) Status bit and the `STKUNF` (Stack Underflow) Status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. At Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the `STKFUL` bit is set. The `STKFUL` bit is cleared by software or by a POR.

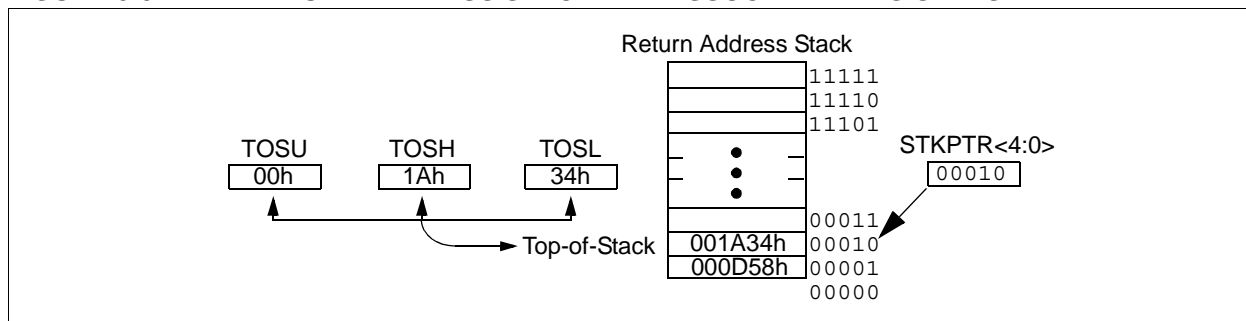
The action that takes place when the stack becomes full depends on the state of the `STVR` (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 19.1 “Configuration Bits”** for a description of the device Configuration bits.) If `STVR` is set (default), the 31st push will push the `(PC + 2)` value onto the stack, set the `STKFUL` bit and reset the device. The `STKFUL` bit will remain set and the Stack Pointer will be set to zero.

If `STVR` is cleared, the `STKFUL` bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and `STKPTR` will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and sets the `STKUNF` bit, while the Stack Pointer remains at zero. The `STKUNF` bit will remain set until cleared by software or a POR occurs.

Note: Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

FIGURE 5-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



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.

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in four registers, RES3:RES0.

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

```
RES3:RES0
= ARG1H:ARG1L • ARG2H:ARG2L
= (ARG1H • ARG2H • 216) +
  (ARG1H • ARG2L • 28) +
  (ARG1L • ARG2H • 28) +
  (ARG1L • ARG2L)
```

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
MOV FARG1L, W
MULWF ARG2L      ; ARG1L * ARG2L ->
                  ; PRODH:PRODL

MOV FPRODH, RES1
MOV FPRODL, RES0
;

MOV FARG1H, W
MULWF ARG2H      ; ARG1H * ARG2H ->
                  ; PRODH:PRODL

MOV FPRODH, RES3
MOV FPRODL, RES2
;

MOV FARG1L, W
MULWF ARG2H      ; ARG1L * ARG2H ->
                  ; PRODH:PRODL

MOV FPRODL, W
ADDWF RES1, F    ; Add cross
MOV FPRODH, W    ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F
;

MOV FARG1H, W
MULWF ARG2L      ; ARG1H * ARG2L ->
                  ; PRODH:PRODL

MOV FPRODL, W
ADDWF RES1, F    ; Add cross
MOV FPRODH, W    ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F
```

Example 8-4 shows the sequence to do a 16 x 16 signed multiply. Equation 8-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pairs' Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

```
RES3:RES0
= ARG1H:ARG1L • ARG2H:ARG2L
= (ARG1H • ARG2H • 216) +
  (ARG1H • ARG2L • 28) +
  (ARG1L • ARG2H • 28) +
  (ARG1L • ARG2L) +
  (-1 • ARG2H<7> • ARG1H:ARG1L • 216) +
  (-1 • ARG1H<7> • ARG2H:ARG2L • 216)
```

EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
MOV F ARG1L, W
MULWF ARG2L      ; ARG1L * ARG2L ->
                  ; PRODH:PRODL

MOV F PRODH, RES1
MOV F PRODL, RES0
;

MOV F ARG1H, W
MULWF ARG2H      ; ARG1H * ARG2H ->
                  ; PRODH:PRODL

MOV F PRODH, RES3
MOV F PRODL, RES2
;

MOV F ARG1L, W
MULWF ARG2H      ; ARG1L * ARG2H ->
                  ; PRODH:PRODL

MOV F PRODL, W
ADDWF RES1, F    ; Add cross
MOV F PRODH, W    ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F
;

MOV F ARG1H, W
MULWF ARG2L      ; ARG1H * ARG2L ->
                  ; PRODH:PRODL

MOV F PRODL, W
ADDWF RES1, F    ; Add cross
MOV F PRODH, W    ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F
;

BTFSS ARG2H, 7   ; ARG2H:ARG2L neg?
BRA SIGN_ARG1    ; no, check ARG1
MOV F ARG1L, W
SUBWF RES2
MOV F ARG1H, W
SUBWFB RES3
;

SIGN_ARG1
BTFSS ARG1H, 7   ; ARG1H:ARG1L neg?
BRA CONT_CODE    ; no, done
MOV F ARG2L, W
SUBWF RES2
MOV F ARG2H, W
SUBWFB RES3
;

CONT_CODE
:
```

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

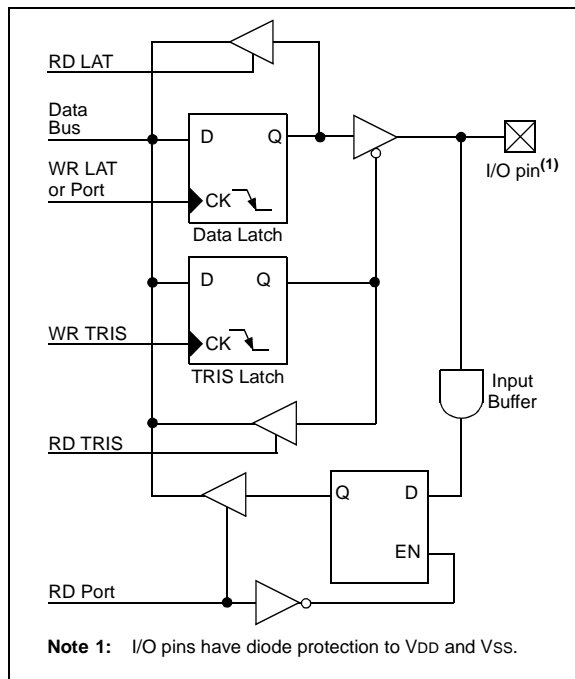
Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (output latch)

The Data Latch (LATA) register is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified model of a generic I/O port without the interfaces to other peripherals is shown in Figure 10-1.

FIGURE 10-1: GENERIC I/O PORT OPERATION



10.1 PORTA, TRISA and LATA Registers

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

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA4 pin is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin.

The sixth pin of PORTA ($\overline{\text{MCLR}}/\text{VPP}/\text{RA5}$) is an input only pin. Its operation is controlled by the MCLRE configuration bit in Configuration Register 3H (CONFIG3H<7>). When selected as a port pin (MCLRE = 0), it functions as a digital input only pin; as such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, RA5 also functions as the programming voltage input during programming.

Note: On a Power-on Reset, RA5 is enabled as a digital input only if Master Clear functionality is disabled.

Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 19.1 "Configuration Bits"** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as '0'.

The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the LVD input. The operation of pins RA3:RA0 as A/D converter inputs is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

Note: On a Power-on Reset, RA3:RA0 are configured as analog inputs and read as '0'. RA4 is always a digital pin.

The RA4/T0CKI pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA    ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF    LATA      ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0x7F     ; Configure A/D
MOVWF   ADCON1   ; for digital inputs
MOVLW   0xD0     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISA    ; Set RA<3:0> as outputs
                ; RA<7:4> as inputs
```

FIGURE 10-6: MCLR/VPP/RA5 PIN BLOCK DIAGRAM

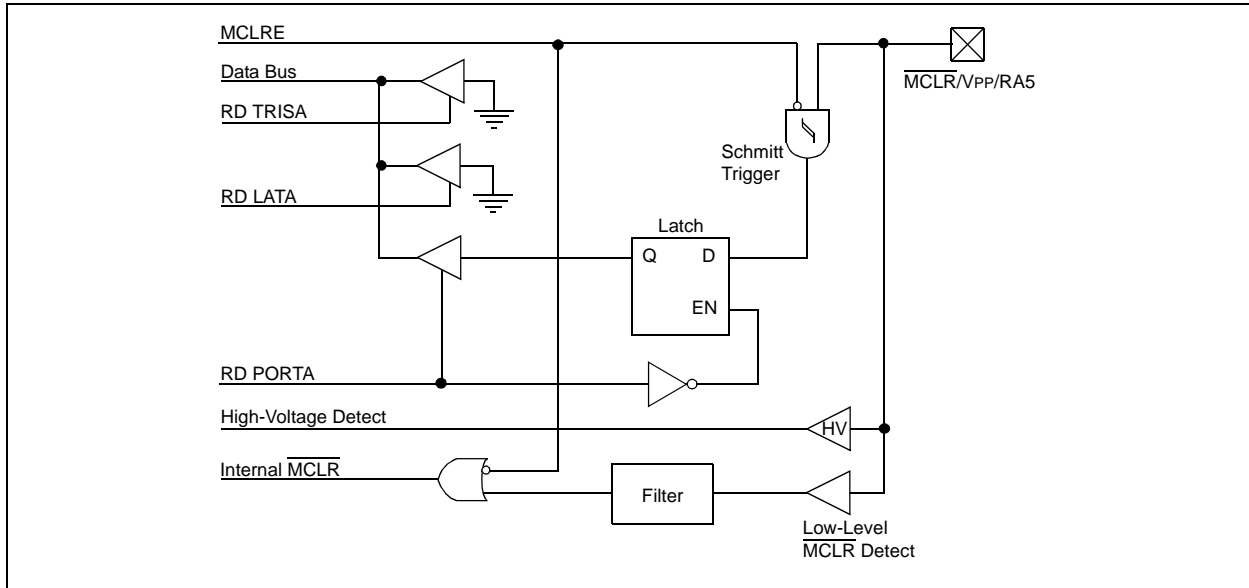


TABLE 10-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	ST	Input/output port pin or analog input.
RA1/AN1/LVDIN	bit 1	ST	Input/output port pin, analog input or Low-Voltage Detect input.
RA2/AN2/VREF-	bit 2	ST	Input/output port pin, analog input or VREF-.
RA3/AN3/VREF+	bit 3	ST	Input/output port pin, analog input or VREF+.
RA4/T0CKI	bit 4	ST	Input/output port pin or external clock input for Timer0. Output is open-drain type.
MCLR/VPP/RA5	bit 5	ST	Master Clear input or programming voltage input (if MCLR is enabled); input only port pin or programming voltage input (if MCLR is disabled).
OSC2/CLKO/RA6	bit 6	ST	OSC2, clock output or I/O pin.
OSC1/CLKI/RA7	bit 7	ST	OSC1, clock input or I/O pin.

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

TABLE 10-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

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
PORTA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5 ⁽²⁾	RA4	RA3	RA2	RA1	RA0	xx0x 0000	uu0u 0000
LATA	LATA7 ⁽¹⁾	LATA6 ⁽¹⁾	—	LATA Data Output Register					xx-x xxxx	uu-u uuuu
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	—	PORTA Data Direction Register					11-1 1111	11-1 1111
ADCON1	—	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	-000 0000	-000 0000

Legend: x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

Note 1: RA7:RA6 and their associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.

2: RA5 is an input only if MCLR is disabled.

PIC18F1220/1320

14.2 Timer1 Oscillator

The Timer1 oscillator may be used as the clock source for Timer3. The Timer1 oscillator is enabled by setting the T1OSCEN (T1CON<3>) bit. The oscillator is a low-power oscillator rated for 32 kHz crystals. See **Section 12.2 “Timer1 Oscillator”** for further details.

14.3 Timer3 Interrupt

The TMR3 register pair (TMR3H:TMR3L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR3 interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit, TMR3IF (PIR2<1>). This interrupt can be enabled/disabled by setting/clearing TMR3 Interrupt Enable bit, TMR3IE (PIE2<1>).

14.4 Resetting Timer3 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = 1011), this signal will reset Timer3. See **Section 15.4.4 “Special Event Trigger”** for more information.

Note: The special event triggers from the CCP module will not set interrupt flag bit, TMR3IF (PIR1<0>).

Timer3 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer3 is running in Asynchronous Counter mode, this Reset operation may not work. In the event that a write to Timer3 coincides with a special event trigger from CCP1, the write will take precedence. In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer3.

TABLE 14-1: REGISTERS ASSOCIATED WITH TIMER3 AS A TIMER/COUNTER

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	TMR3IF	—	0--0 -00-	0--0 -00-
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	TMR3IE	—	0--0 -00-	0--0 -00-
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	TMR3IP	—	1--1 -11-	1--1 -11-
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \overline{C}	TMR1CS	TMR1ON	0000 0000	u0uu uuuu
T3CON	RD16	—	T3CKPS1	T3CKPS0	T3CCP1	T3SYN \overline{C}	TMR3CS	TMR3ON	0-00 0000	u-uu uuuu

Legend: x = unknown, u = unchanged, – = unimplemented, read as ‘0’. Shaded cells are not used by the Timer3 module.

17.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has seven inputs for the PIC18F1220/1320 devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

A new feature for the A/D converter is the addition of programmable acquisition time. This feature allows the user to select a new channel for conversion and to set the GO/DONE bit immediately. When the GO/DONE bit is set, the selected channel is sampled for the programmed acquisition time before a conversion is actually started. This removes the firmware overhead that may have been required to allow for an acquisition (sampling) period (see Register 17-3 and **Section 17.3 “Selecting and Configuring Automatic Acquisition Time”**).

The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 17-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 17-2, configures the functions of the port pins. The ADCON2 register, shown in Register 17-3, configures the A/D clock source, programmed acquisition time and justification.

PIC18F1220/1320

17.7 A/D Conversions

Figure 17-3 shows the operation of the A/D converter after the GO bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Low-Power Sleep mode before the conversion begins.

Figure 17-4 shows the operation of the A/D converter after the GO bit has been set and the ACQT2:ACQT0 bits are set to '010' and selecting a 4 TAD acquisition time before the conversion starts.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers).

After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

Note: The GO/DONE bit should **NOT** be set in the same instruction that turns on the A/D.

FIGURE 17-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

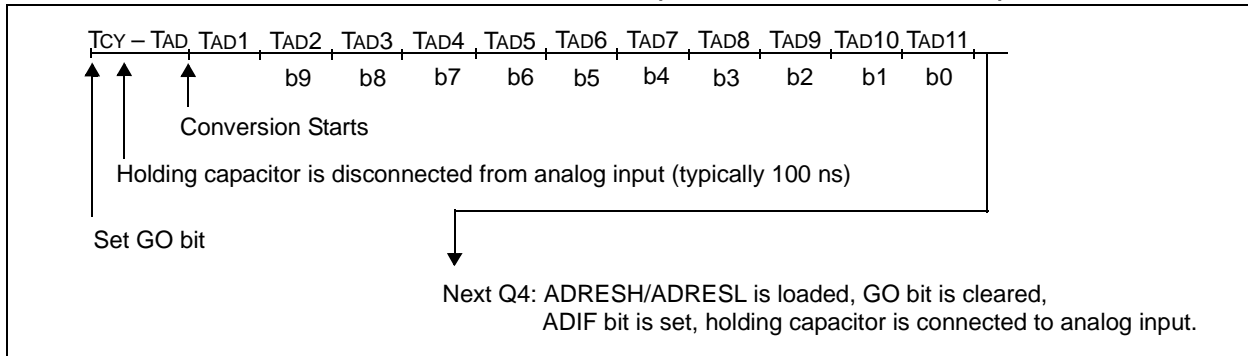
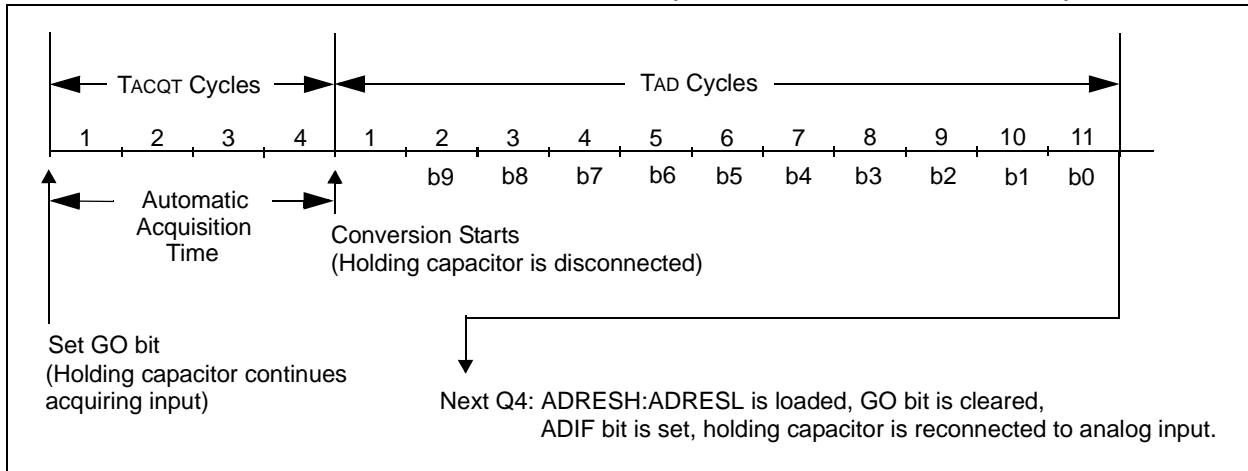


FIGURE 17-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)

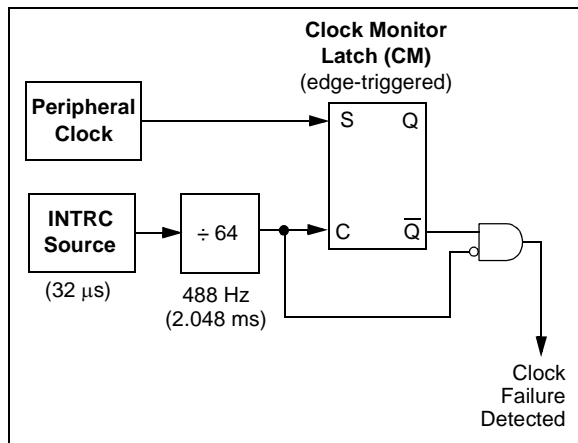


19.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation, in the event of an external oscillator failure, by automatically switching the system clock to the internal oscillator block. The FSCM function is enabled by setting the Fail-Safe Clock Monitor Enable bit, FSCM (CONFIG1H<6>).

When FSCM is enabled, the INTRC oscillator runs at all times to monitor clocks to peripherals and provide an instant backup clock in the event of a clock failure. Clock monitoring (shown in Figure 19-3) is accomplished by creating a sample clock signal, which is the INTRC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral system clock and the sample clock are presented as inputs to the Clock Monitor latch (CM). The CM is set on the falling edge of the system clock source, but cleared on the rising edge of the sample clock.

FIGURE 19-3: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 19-4). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- the system clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the Fail-Safe condition); and
- the WDT is reset.

Since the postscaler frequency from the internal oscillator block may not be sufficiently stable, it may be desirable to select another clock configuration and enter an alternate power managed mode (see **Section 19.3.1 “Special Considerations for Using Two-Speed Start-up”** and **Section 3.1.3 “Multiple Sleep Commands”** for more details). This can be done to attempt a partial recovery, or execute a controlled shutdown.

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.

Adjustments to the internal oscillator block, using the OSTUNE register, also affect the period of the FSCM by the same factor. This can usually be neglected, as the clock frequency being monitored is generally much higher than the sample clock frequency.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

19.4.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF2:IRCF0 bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

DAW	Decimal Adjust W Register				
Syntax:	[<i>label</i>] DAW				
Operands:	None				
Operation:	If [W<3:0> > 9] or [DC = 1] then (W<3:0>) + 6 → W<3:0>; else (W<3:0>) → W<3:0>; If [W<7:4> > 9] or [C = 1] then (W<7:4>) + 6 → W<7:4>; else (W<7:4>) → W<7:4>;				
Status Affected:	C, DC				
Encoding:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>0111</td></tr></table>	0000	0000	0000	0111
0000	0000	0000	0111		
Description:	DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result. The Carry bit may be set by DAW regardless of its setting prior to the DAW instruction.				
Words:	1				
Cycles:	1				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register W	Process Data	Write W

Example 1: DAW

Before Instruction

W = 0xA5
C = 0
DC = 0

After Instruction

W = 0x05
C = 1
DC = 0

Example 2:

Before Instruction

W = 0xCE
C = 0
DC = 0

After Instruction

W = 0x34
C = 1
DC = 0

DECF		Decrement f								
Syntax:	[<i>label</i>] DECF f [,d [,a]]									
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]									
Operation:	(f) − 1 → dest									
Status Affected:	C, DC, N, OV, Z									
Encoding:	<table border="1"><tr><td>0000</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>						0000	01da	ffff	ffff
0000	01da	ffff	ffff							
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). 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 (default).									
Words:	1									
Cycles:	1									
Q Cycle Activity:										
	Q1	Q2	Q3	Q4						
	Decode	Read register 'f'	Process Data	Write to destination						

Example: DECF CNT

Before Instruction

CNT = 0x01
Z = 0

After Instruction

CNT = 0x00
Z = 1

IORLW		Inclusive OR literal with W						
Syntax:	[<i>label</i>] IORLW k							
Operands:	$0 \leq k \leq 255$							
Operation:	(W) .OR. $k \rightarrow W$							
Status Affected:	N, Z							
Encoding:	<table border="1"><tr><td>0000</td><td>1001</td><td>kkkk</td><td>kkkk</td></tr></table>				0000	1001	kkkk	kkkk
0000	1001	kkkk	kkkk					
Description:	The contents of W are OR'ed with the 8-bit literal 'k'. The result is placed in W.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'k'	Process Data	Write to W				

Example: IORLW 0x35

Before Instruction

W = 0x9A

After Instruction

W = 0xBF

IORWF		Inclusive OR W with f						
Syntax:	[<i>label</i>] IORWF f [,d [,a]]							
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]							
Operation:	(W) .OR. (f) → dest							
Status Affected:	N, Z							
Encoding:	<table border="1"><tr><td>0001</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>				0001	00da	ffff	ffff
0001	00da	ffff	ffff					
Description:	Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). 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 (default).							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process Data	Write to destination				

Example: IORWF RESULT, W

Before Instruction

RESULT = 0x13

W = 0x91

After Instruction

RESULT = 0x13

W = 0x93

PIC18F1220/1320

RCALL Relative Call

Syntax: [*label*] RCALL n

Operands: $-1024 \leq n \leq 1023$

Operation: (PC) + 2 → TOS,
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1101	1nnn	nnnn	nnnn
------	------	------	------

Description: Subroutine call with a jump up to 1K from the current location. First, return address (PC + 2) is pushed onto the stack. Then, add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a 2-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n' Push PC to stack	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: HERE RCALL Jump

Before Instruction

PC = Address (HERE)

After Instruction

PC = Address (Jump)

TOS = Address (HERE + 2)

RESET Reset

Syntax: [*label*] RESET

Operands: None

Operation: Reset all registers and flags that are affected by a MCLR Reset.

Status Affected: All

Encoding:

0000	0000	1111	1111
------	------	------	------

Description: This instruction provides a way to execute a MCLR Reset in software.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Start Reset	No operation	No operation

Example: RESET

After Instruction

Registers = Reset Value
Flags* = Reset Value

PIC18F1220/1320

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

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		
	Power-Down Current (IPD) ⁽¹⁾						
	PIC18LF1220/1320	0.1	0.5	μA	-40°C	VDD = 2.0V, (Sleep mode)	
		0.1	0.5	μA	+25°C		
		0.2	1.9	μA	+85°C		
	PIC18LF1220/1320	0.1	0.5	μA	-40°C	VDD = 3.0V, (Sleep mode)	
		0.1	0.5	μA	+ 25°C		
		0.3	1.9	μA	+85°C		
	All devices	0.1	2.0	μA	-40°C	VDD = 5.0V, (Sleep mode)	
		0.1	2.0	μA	+25°C		
		0.4	6.5	μA	+85°C		
	Extended devices	11.2	50	μA	+125°C		
	Supply Current (IDD) ^(2,3)						
	PIC18LF1220/1320	8	40	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (RC_RUN mode, Internal oscillator source)
		9	40	μA	+25°C		
		11	40	μA	+85°C		
	PIC18LF1220/1320	25	68	μA	-40°C	VDD = 3.0V	
25		68	μA	+25°C			
20		68	μA	+85°C			
All devices	55	80	μA	-40°C	VDD = 5.0V		
	55	80	μA	+25°C			
	50	80	μA	+85°C			
Extended devices	50	80	μA	+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} ;
MCLR = V_{DD} ; 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 $I_r = V_{DD}/2R_{EXT}$ (mA) with REXT in k Ω .
- 4:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.

TABLE 22-5: PLL CLOCK TIMING SPECIFICATIONS, HS/HSPLL MODE (V_{DD} = 4.2V TO 5.5V)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	10	MHz	HS and HSPLL mode only
F11	FSYS	On-Chip VCO System Frequency	16	—	40	MHz	HSPLL mode only
F12	TPLL	PLL Start-up Time (Lock Time)	—	—	2	ms	HSPLL mode only
F13	ΔCLK	CLKO Stability (Jitter)	-2	—	+2	%	HSPLL mode only

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

**TABLE 22-6: INTERNAL RC ACCURACY: PIC18F1220/1320 (INDUSTRIAL)
PIC18LF1220/1320 (INDUSTRIAL)**

PIC18LF1220/1320 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial						
PIC18F1220/1320 (Industrial)		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	Min.	Typ.	Max.	Units	Conditions		
	INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz ⁽¹⁾							
	PIC18LF1220/1320	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V	
		-5	—	5	%	-10°C to +85°C	VDD = 2.7-3.3V	
		-10	—	10	%	-40°C to +85°C	VDD = 2.7-3.3V	
	PIC18F1220/ 1320PIC18F1220/1320	-2	+/-1	2	%	+25°C	VDD = 4.5-5.5V	
		-5	—	5	%	-10°C to +85°C	VDD = 4.5-5.5V	
		-10	—	10	%	-40°C to +85°C	VDD = 4.5-5.5V	
	INTRC Accuracy @ Freq = 31 kHz ⁽²⁾							
	PIC18LF1220/1320	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V	
	PIC18F1220/ 1320PIC18F1220/1320	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V	

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

Note 1: Frequency calibrated at 25°C. OSCUNE register can be used to compensate for temperature and V_{DD} drift.

2: INTRC frequency after calibration.

3: Change of INTRC frequency as V_{DD} changes.

TABLE 22-12: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
125	TdtV2ckl	SYNC RCV (MASTER & SLAVE) Data Hold before CK↓ (DT hold time)	10	—	ns	
126	TckL2dtl	Data Hold after CK↓ (DT hold time)	15	—	ns	

**TABLE 22-13: A/D CONVERTER CHARACTERISTICS: PIC18F1220/1320 (INDUSTRIAL)
PIC18LF1220/1320 (INDUSTRIAL)**

Param No.	Symbol	Characteristic	Min.	Typ.	Max.	Units	Conditions
A01	NR	Resolution	—	—	10	bit	$\Delta V_{REF} \geq 3.0V$
A03	EIL	Integral Linearity Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	Differential Linearity Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A06	EOFF	Offset Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	Gain Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A10	—	Monotonicity	guaranteed ⁽²⁾			—	
A20	ΔV_{REF}	Reference Voltage Range ($V_{REFH} - V_{REFL}$)	3	—	$AV_{DD} - AV_{SS}$	V	For 10-bit resolution
A21	V_{REFH}	Reference Voltage High	$AV_{SS} + 3.0V$	—	$AV_{DD} + 0.3V$	V	For 10-bit resolution
A22	V_{REFL}	Reference Voltage Low	$AV_{SS} - 0.3V$	—	$AV_{DD} - 3.0V$	V	For 10-bit resolution
A25	V_{AIN}	Analog Input Voltage	V_{REFL}	—	V_{REFH}	V	
A28	AV_{DD}	Analog Supply Voltage	$V_{DD} - 0.3$	—	$V_{DD} + 0.3$	V	
A29	AV_{SS}	Analog Supply Voltage	$V_{SS} - 0.3$	—	$V_{SS} + 0.3$	V	
A30	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	2.5	k Ω	
A40	I _{AD}	A/D Conversion Current (V_{DD})	PIC18F1X20 —	180	—	μA	Average current consumption when A/D is on (Note 1)
		PIC18LF1X20 —	—	90	—	μA	
A50	I _{REF}	V _{REF} Input Current (Note 3)	—	—	± 5	μA	During V_{AIN} acquisition. During A/D conversion cycle.
			—	—	± 150	μA	

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current specification includes any such leakage from the A/D module.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: V_{REFH} current is from RA3/AN3/ V_{REF+} pin or AV_{DD} , whichever is selected as the V_{REFH} source.

V_{REFL} current is from RA2/AN2/ V_{REF-} pin or AV_{SS} , whichever is selected as the V_{REFL} source.

FIGURE 23-3: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} PRI_RUN, EC MODE, -40°C TO $+125^{\circ}\text{C}$

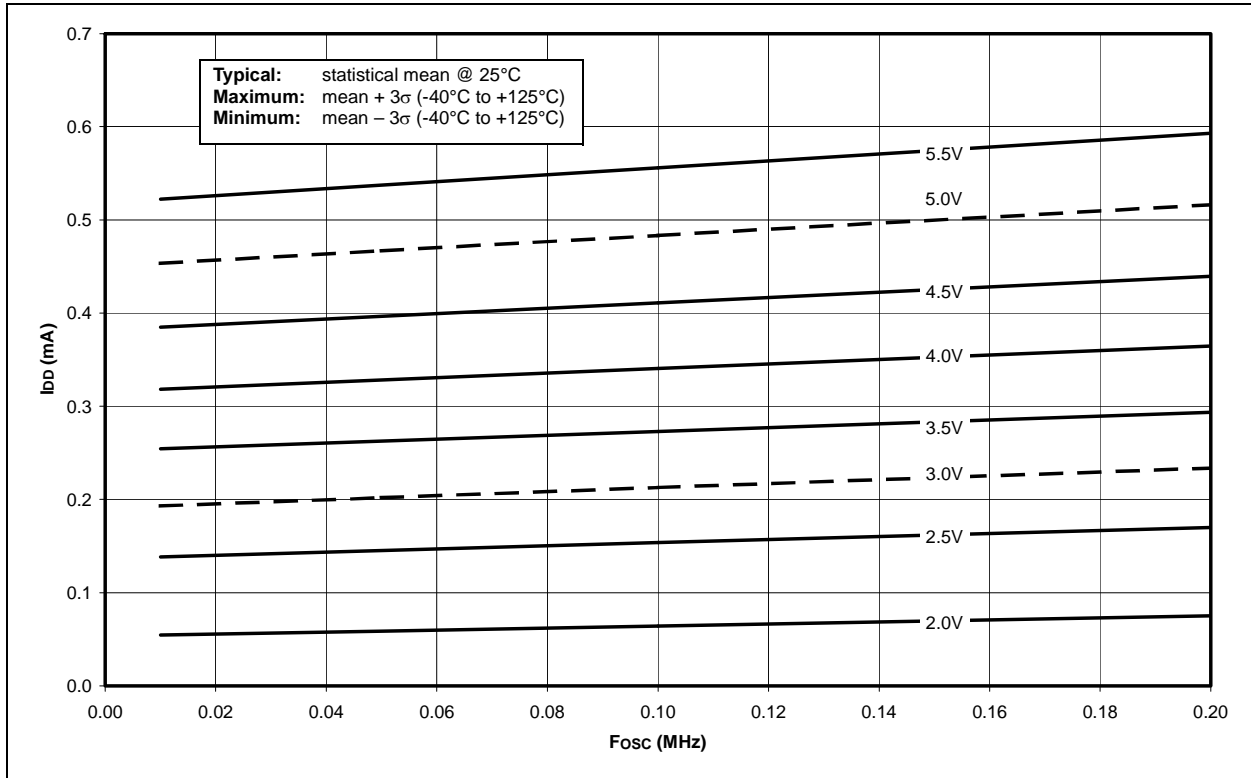
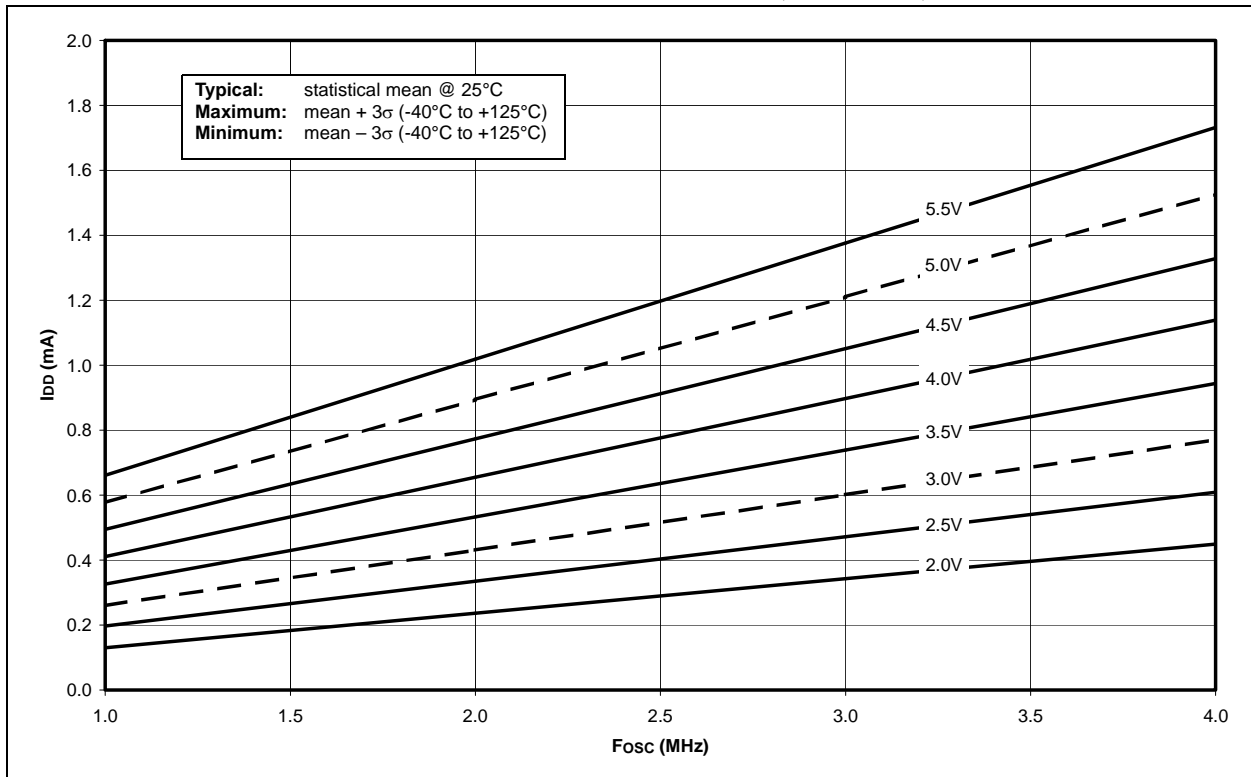


FIGURE 23-4: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} PRI_RUN, EC MODE, $+25^{\circ}\text{C}$



PIC18F1220/1320

FIGURE 23-5: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} PRI_RUN, EC MODE, -40°C TO $+125^{\circ}\text{C}$

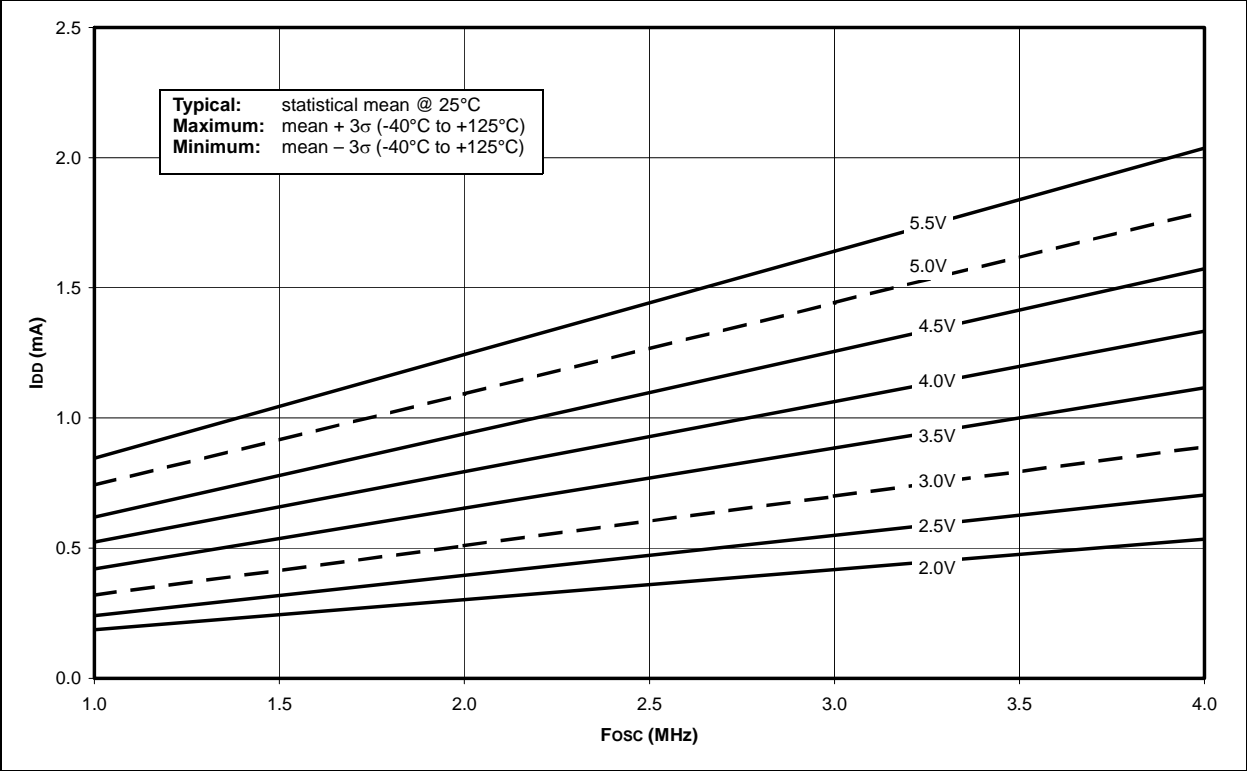
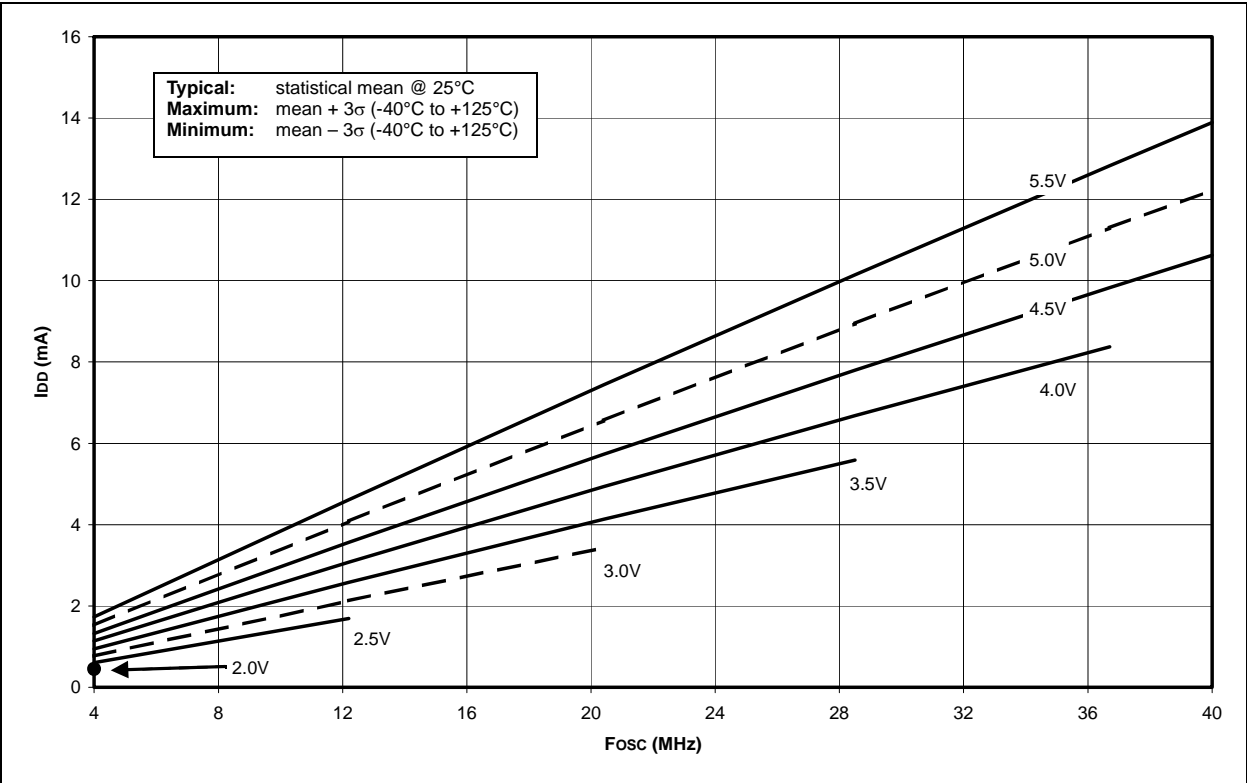


FIGURE 23-6: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} PRI_RUN, EC MODE, $+25^{\circ}\text{C}$



PIC18F1220/1320

FIGURE 23-13: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} PRI_IDLE, EC MODE, +25°C

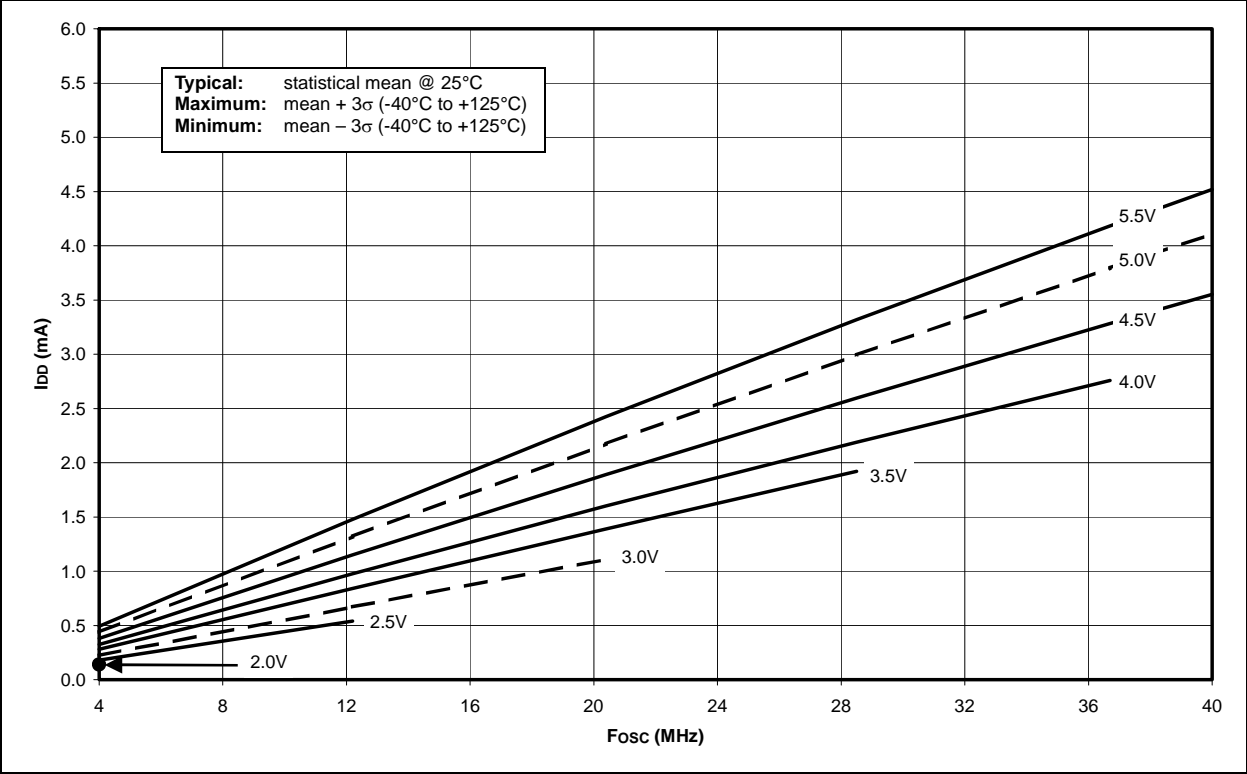
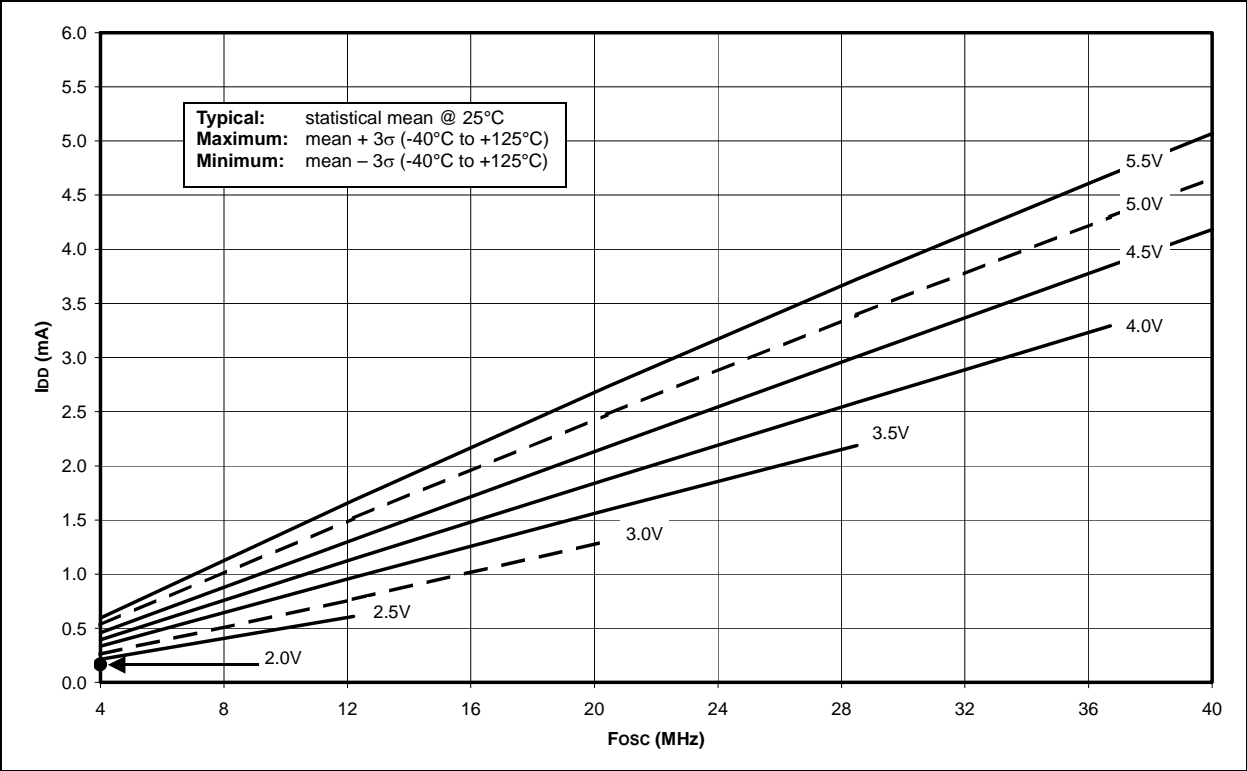


FIGURE 23-14: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} PRI_IDLE, EC MODE, -40°C TO +125°C



Worldwide Sales and Service

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX

Tel: 512-257-3370

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland

Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Novi, MI
Tel: 248-848-4000

Houston, TX

Tel: 281-894-5983
Indianapolis

Noblesville, IN

Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY

Tel: 631-435-6000

San Jose, CA

Tel: 408-735-9110

Canada - Toronto

Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong

Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan

Tel: 86-769-8702-9880

China - Hangzhou

Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR

Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen

Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen

Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai

Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore

Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune

Tel: 91-20-3019-1500

Japan - Osaka

Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo

Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu

Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul

Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang

Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila

Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore

Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu

Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung

Tel: 886-7-213-7828

Taiwan - Taipei

Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok

Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen

Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf

Tel: 49-2129-3766400

Germany - Munich

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim

Tel: 49-7231-424750

Italy - Milan

Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice

Tel: 39-049-7625286

Netherlands - Drunen

Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw

Tel: 48-22-3325737

Spain - Madrid

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm

Tel: 46-8-5090-4654

UK - Wokingham

Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15