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	20MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f628a-i-ss

TABLE 3-2: PIC16F627A/628A/648A PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB4/PGM	RB4	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low-voltage programming input pin. When low-voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/T1OSO/T1CKI/PGC	RB6	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSO	—	XTAL	Timer1 oscillator output
	T1CKI	ST	—	Timer1 clock input
	PGC	ST	—	ICSP™ programming clock
RB7/T1OSI/PGD	RB7	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSI	XTAL	—	Timer1 oscillator input
	PGD	ST	CMOS	ICSP data I/O
VSS	VSS	Power	—	Ground reference for logic and I/O pins
VDD	VDD	Power	—	Positive supply for logic and I/O pins

Legend: O = Output, — = Not used, TTL = TTL Input, CMOS = CMOS Output, I = Input, OD = Open Drain Output, P = Power, ST = Schmitt Trigger Input, AN = Analog

4.2.2 SPECIAL FUNCTION REGISTERS

The SFRs are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (Table 4-3). These registers are static RAM.

The special registers can be classified into two sets (core and peripheral). The SFRs associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

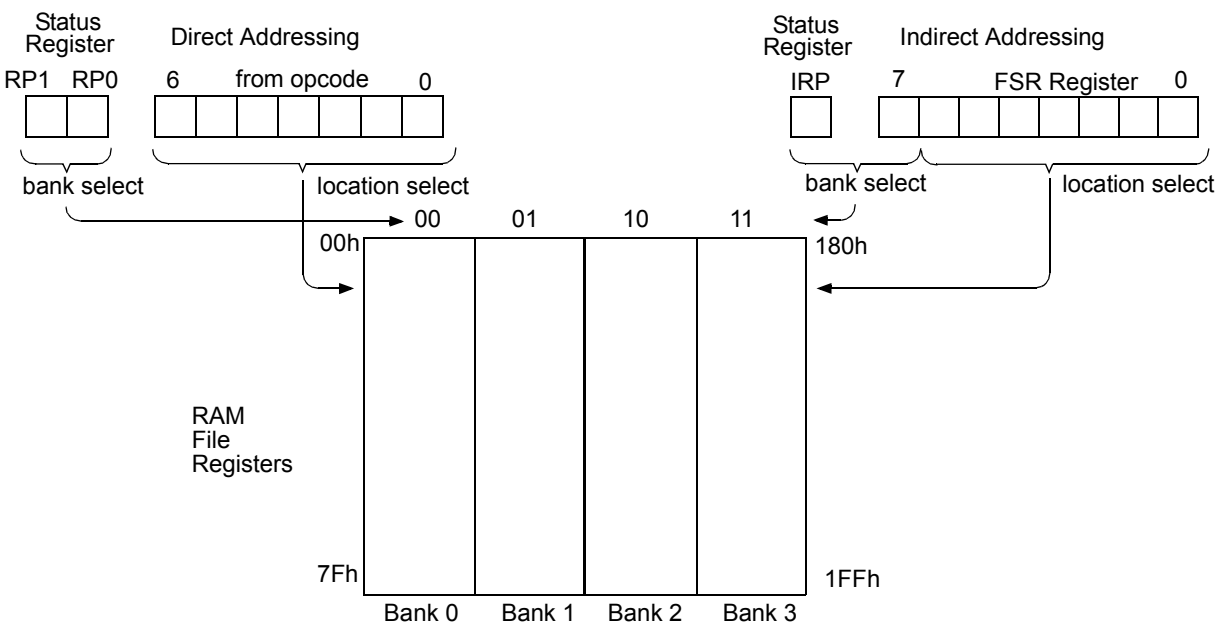
TABLE 4-3: SPECIAL REGISTERS SUMMARY BANK0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset(1)	Details on Page
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	30
01h	TMR0	Timer0 Module's Register								xxxx xxxx	47
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	30
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	24
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	30
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000	33
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	38
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	30
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	26
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	28
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	50
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	50
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	--00 0000	50
11h	TMR2	TMR2 Module's Register								0000 0000	54
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	54
13h	—	Unimplemented								—	—
14h	—	Unimplemented								—	—
15h	CCPR1L	Capture/Compare/PWM Register (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	57
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	74
19h	TXREG	USART Transmit Data Register								0000 0000	79
1Ah	RCREG	USART Receive Data Register								0000 0000	82
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	—	Unimplemented								—	—
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	63

Legend: - = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: For the initialization condition for registers tables, refer to Table 14-6 and Table 14-7.

FIGURE 4-5: DIRECT/INDIRECT ADDRESSING PIC16F627A/628A/648A



Note: For memory map detail see Figure 4-3, Figure 4-2 and Figure 4-1.

NOTES:

7.3 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (**Section 7.3.2 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

7.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high and low time requirements. Refer to Table 17-8 in the Electrical Specifications Section, timing parameters 45, 46 and 47.

7.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading the TMR1H or TMR1L register, while the timer is running from an external asynchronous clock, will produce a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Example 7-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

EXAMPLE 7-1: READING A 16-BIT FREE-RUNNING TIMER

```
; All interrupts are disabled
    MOVF    TMR1H, W    ;Read high byte
    MOVWF   TMPH        ;
    MOVF    TMR1L, W    ;Read low byte
    MOVWF   TMPL        ;
    MOVF    TMR1H, W    ;Read high byte
    SUBWF   TMPH, W     ;Sub 1st read with
                        ;2nd read
    BTFSC   STATUS,Z    ;Is result = 0
    GOTO    CONTINUE    ;Good 16-bit read
;
; TMR1L may have rolled over between the
; read of the high and low bytes. Reading
; the high and low bytes now will read a good
; value.
;
    MOVF    TMR1H, W    ;Read high byte
    MOVWF   TMPH        ;
    MOVF    TMR1L, W    ;Read low byte
    MOVWF   TMPL        ;
; Re-enable the Interrupts (if required)
CONTINUE                ;Continue with your
                        ;code
```

7.4 Timer1 Oscillator

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). It will continue to run during Sleep. It is primarily intended for a 32.768 kHz watch crystal. Table 7-1 shows the capacitor selection for the Timer1 oscillator.

The user must provide a software time delay to ensure proper oscillator start-up.

TABLE 7-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Freq	C1	C2
32.768 kHz	15 pF	15 pF

Note: These values are for design guidance only. Consult Application Note AN826 "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826) for further information on Crystal/Capacitor Selection.

7.5 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 module is configured in Compare mode to generate a "special event trigger" (CCP1M<3:0> = 1011), this signal will reset Timer1.

Note: The special event triggers from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL register pair effectively becomes the period register for Timer1.

7.6 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR or any other Reset except by the CCP1 special event triggers (see **Section 9.2.4 "Special Event Trigger"**).

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

7.7 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

TABLE 7-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer1 module.

NOTES:

12.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART) MODULE

The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as a Serial Communications Interface (SCI). The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISB<2:1> have to be set in order to configure pins RB2/TX/CK and RB1/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

Register 12-1 shows the Transmit Status and Control Register (TXSTA) and Register 12-2 shows the Receive Status and Control Register (RCSTA).

REGISTER 12-1: TXSTA – TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC**: Clock Source Select bit
Asynchronous mode
Don't care
Synchronous mode
1 = Master mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

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

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode
1 = High speed
0 = Low speed
Synchronous mode
Unused in this mode

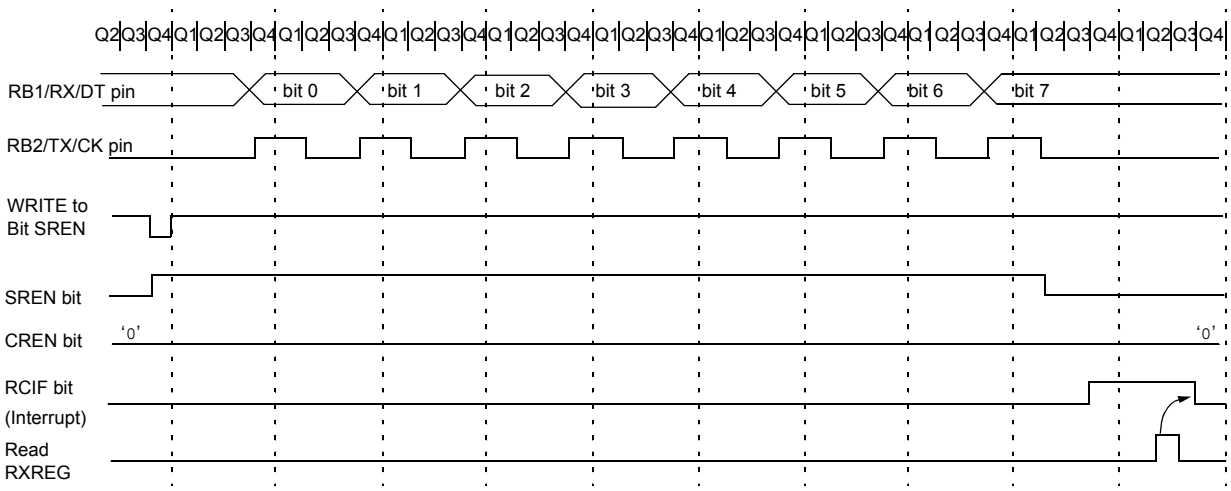
bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D**: 9th bit of transmit data. Can be parity bit.

Note 1: SREN/CREN overrides TXEN in SYNC mode.

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

FIGURE 12-10: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRG = 0.

12.5 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RB2/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

12.5.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the Sleep mode.

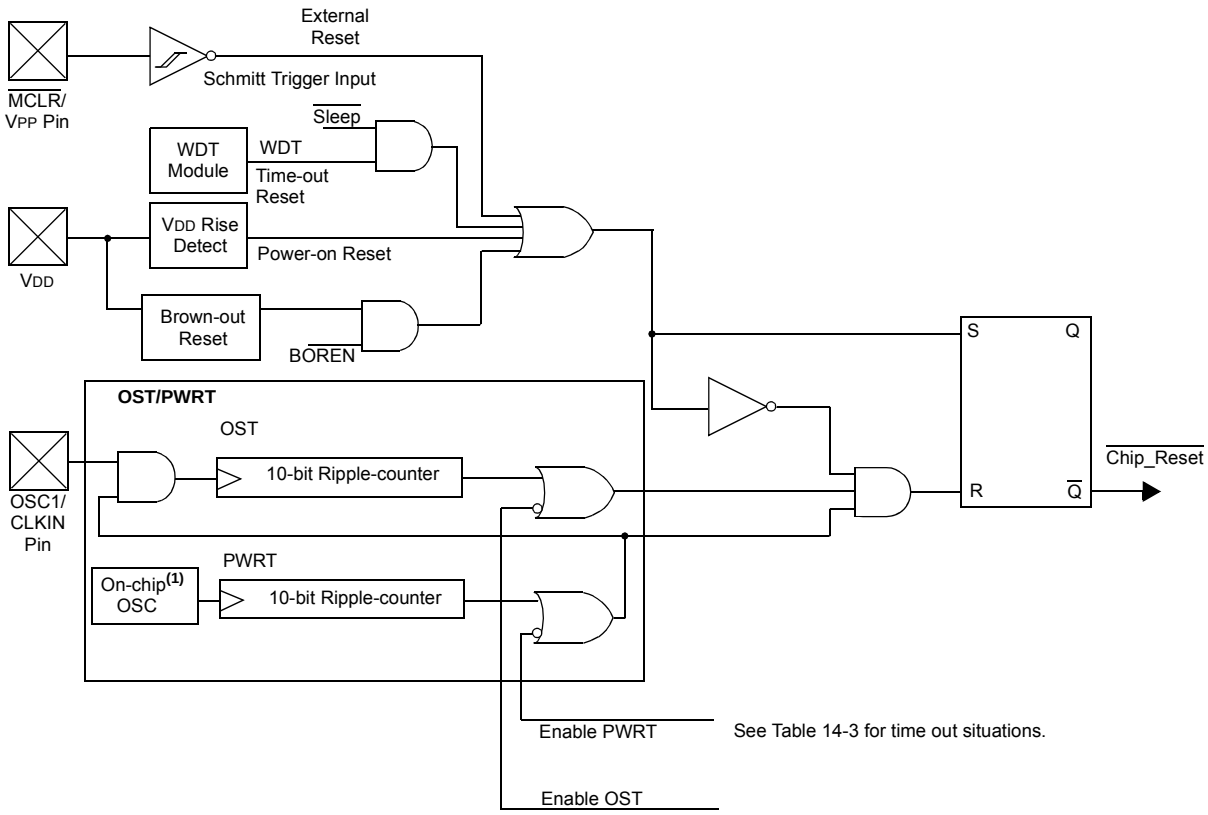
If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) Flag bit TXIF will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
e) If enable bit TXIE is set, the interrupt will wake the chip from Sleep and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Follow these steps when setting up a Synchronous Slave Transmission:

1. TRISB<1> and TRISB<2> should both be set to '1' to configure the RB1/RX/DT and RB2/TX/CK pins as inputs. Output drive, when required, is controlled by the peripheral circuitry.
2. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
3. Clear bits CREN and SREN.
4. If interrupts are desired, then set enable bit TXIE.
5. If 9-bit transmission is desired, then set bit TX9.
6. Enable the transmission by setting enable bit TXEN.
7. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
8. Start transmission by loading data to the TXREG register.

FIGURE 14-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: This is a separate oscillator from the INTOSC/RC oscillator.

14.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOR)

14.4.1 POWER-ON RESET (POR)

The on-chip POR holds the part in Reset until a VDD rise is detected (in the range of 1.2-1.7V). A maximum rise time for VDD is required. See **Section 17.0 "Electrical Specifications"** for details.

The POR circuit does not produce an internal Reset when VDD declines.

When the device starts normal operation (exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure proper operation. If these conditions are not met, the device must be held in Reset via $\overline{MCLR}$, BOR or PWRT until the operating conditions are met.

For additional information, refer to Application Note AN607 "*Power-up Trouble Shooting*" (DS00607).

14.4.2 POWER-UP TIMER (PWRT)

The PWRT provides a fixed 72 ms (nominal) time out on power-up (POR) or if enabled from a Brown-out Reset. The PWRT operates on an internal RC oscillator. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, $\overline{PWRTE}$ can disable (if set) or enable (if cleared or programmed) the PWRT. It is recommended that the PWRT be enabled when Brown-out Reset is enabled.

The power-up time delay will vary from chip-to-chip and due to VDD, temperature and process variation. See DC parameters Table 17-7 for details.

14.4.3 OSCILLATOR START-UP TIMER (OST)

The OST provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. Program execution will not start until the OST time out is complete. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from Sleep. See Table 17-7.

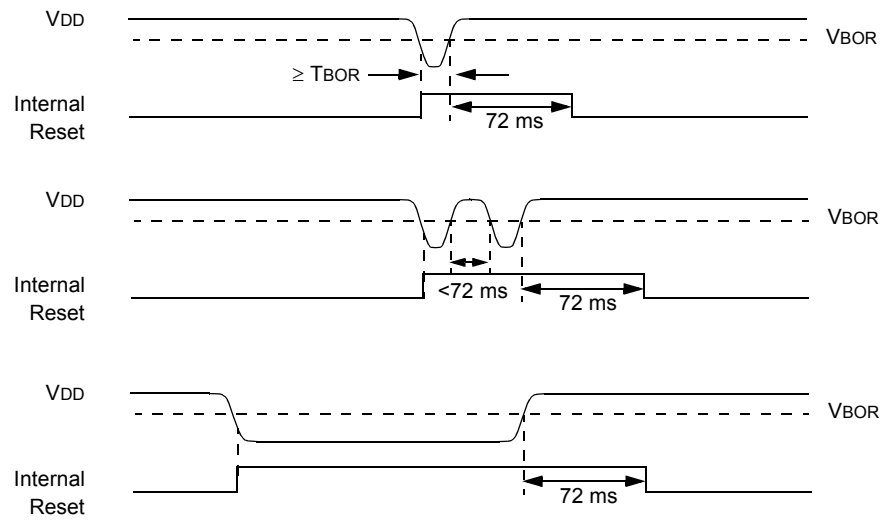
14.4.4 BROWN-OUT RESET (BOR)

The PIC16F627A/628A/648A have on-chip BOR circuitry. A configuration bit, BOREN, can disable (if clear/programmed) or enable (if set) the BOR circuitry. If VDD falls below VBOR for longer than TBOR, the brown-out situation will reset the chip. A Reset is not assured if VDD falls below VBOR for shorter than TBOR. VBOR and TBOR are defined in Table 17-2 and Table 17-7, respectively.

On any Reset (Power-on, Brown-out, Watchdog, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 14-7). The Power-up Timer will now be invoked, if enabled, and will keep the chip in Reset an additional 72 ms.

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-Up Timer will execute a 72 ms Reset. Figure 14-7 shows typical brown-out situations.

FIGURE 14-7: BROWN-OUT SITUATIONS WITH PWRT ENABLED



Note: 72 ms delay only if $\overline{PWRTE}$ bit is programmed to '0'.

FIGURE 14-8: TIME OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE

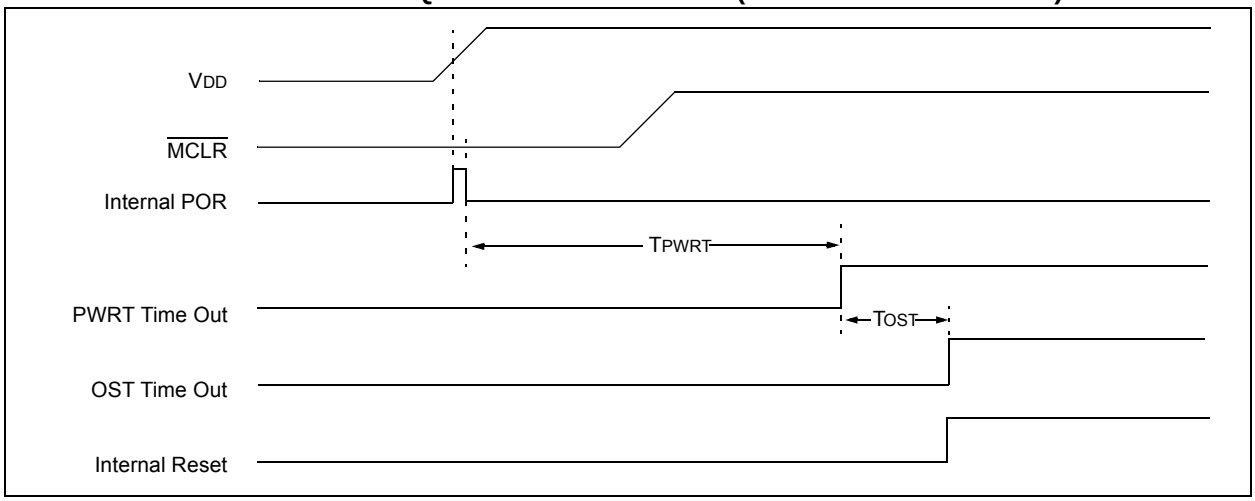


FIGURE 14-9: TIME OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

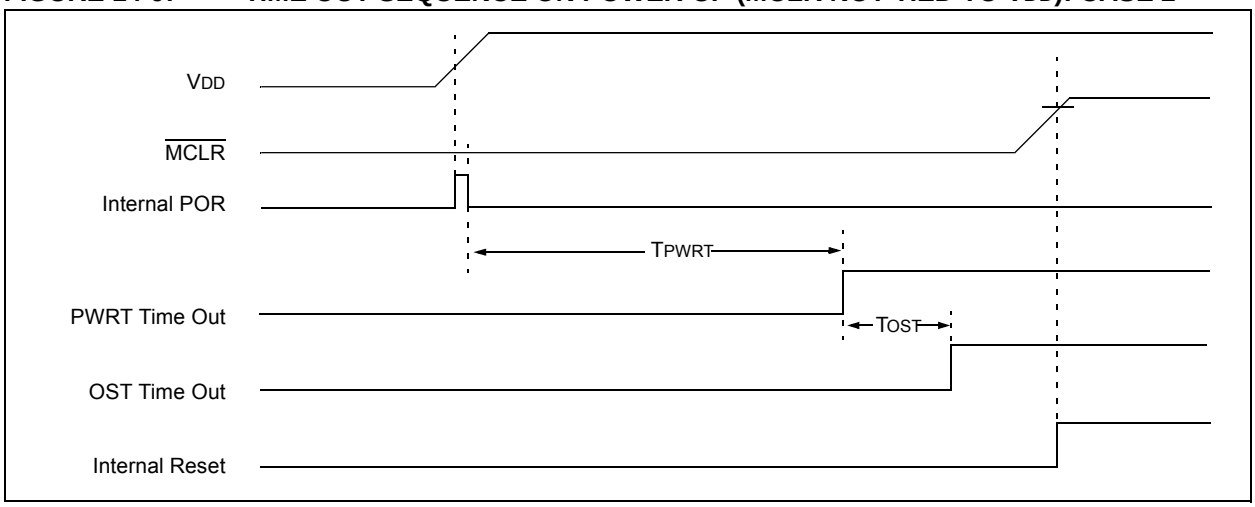
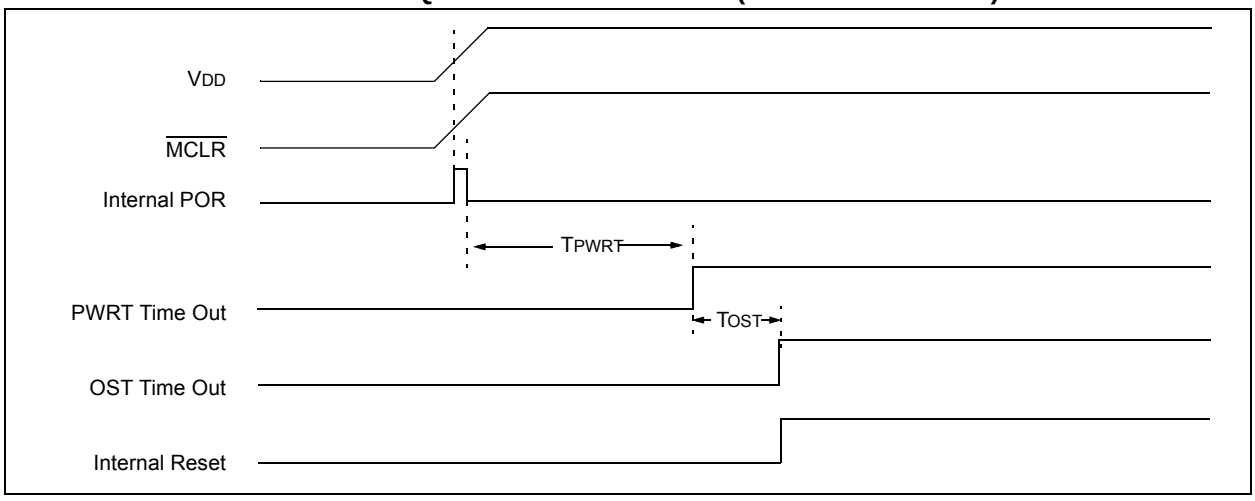


FIGURE 14-10: TIME OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)



BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: $0 \le f \le 127$
$0 \le b < 7$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSS   REG1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1> = 0,
PC = address FALSE
if FLAG<1> = 1,
PC = address TRUE

CALL **Call Subroutine**

Syntax: [*label*] CALL k

Operands: $0 \le k \le 2047$

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL    THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 127$

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF    REG1
```

Before Instruction
REG1 = 0x5A
After Instruction
REG1 = 0x00
Z = 1

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1, 1
```
Before Instruction
REG1 = 3
W = 2
C = ?
After Instruction
REG1 = 1
W = 2
C = 1; result is positive
DC = 1
Z = 0

Example 2: Before Instruction
REG1 = 2
W = 2
C = ?
After Instruction
REG1 = 0
W = 2
C = 1; result is zero
Z = DC = 1

Example 3: Before Instruction
REG1 = 1
W = 2
C = ?
After Instruction
REG1 = 0xFF
W = 2
C = 0; result is negative
Z = DC = 0

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<3:0>) → (dest<7:4>),
(f<7:4>) → (dest<3:0>)

Status Affected: None

Encoding:

00	1110	dfff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W register. If 'd' is '1', the result is placed in register 'f'.

Words: 1

Cycles: 1

Example
```
SWAPF   REG1, 0
```
Before Instruction
REG1 = 0xA5
After Instruction
REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: $5 \le f \le 7$

Operation: (W) → TRIS register f;

Status Affected: None

Encoding:

00	0000	0110	0fff

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC® MCU products, do not use this instruction.

16.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

16.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

16.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

16.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

16.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

17.3 DC Characteristics: PIC16F627A/628A/648A (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ Ta ≤ +125°C for extended					
Param No.	Device Characteristics	Min†	Typ	Max	Units	Conditions	
						VDD	Note
Supply Voltage (VDD)							
D001	—	3.0	—	5.5	V	—	
Power-down Base Current (IPD)							
D020E	—	—	0.01	4	µA	3.0	WDT, BOR, Comparators, VREF and T1OSC: disabled
		—	0.02	8	µA	5.0	
Peripheral Module Current (ΔIMOD)(1)							
D021E	—	—	2	9	µA	3.0	WDT Current
		—	9	20	µA	5.0	
D022E	—	—	29	52	µA	4.5	BOR Current
		—	30	55	µA	5.0	
D023E	—	—	22	37	µA	3.0	Comparator Current (Both comparators enabled)
		—	44	68	µA	5.0	
D024E	—	—	50	75	µA	3.0	VREF Current
		—	83	110	µA	5.0	
D025E	—	—	1.3	4	µA	3.0	T1OSC Current
		—	1.8	6	µA	5.0	
Supply Current (IDD)							
D010E	—	—	15	28	µA	3.0	FOSC = 32 kHz LP Oscillator Mode
		—	28	54	µA	5.0	
D011E	—	—	175	340	µA	3.0	FOSC = 1 MHz XT Oscillator Mode
		—	320	520	µA	5.0	
D012E	—	—	450	650	µA	3.0	FOSC = 4 MHz XT Oscillator Mode
		—	0.710	1.1	mA	5.0	
D012AE	—	—	565	785	µA	3.0	FOSC = 4 MHz INTOSC
		—	0.895	1.3	mA	5.0	
D013E	—	—	2.5	2.9	mA	4.5	FOSC = 20 MHz HS Oscillator Mode
		—	2.75	3.5	mA	5.0	

Note 1: The "Δ" current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement. Max values should be used when calculating total current consumption.

**TABLE 17-1: DC Characteristics: PIC16F627A/628A/648A (Industrial, Extended)
PIC16LF627A/628A/648A (Industrial)**

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and -40°C ≤ TA ≤ +125°C for extended Operating voltage VDD range as described in DC specification Table 17-2 and Table 17-3				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Data EEPROM Memory					
D120	ED	Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ 85°C
D120A	ED	Endurance	10K	100K		E/W	85°C ≤ TA ≤ 125°C
D121	VDRW	VDD for read/write	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D122	TDEW	Erase/Write cycle time	—	4	8*	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(1)	1M	10M	—	E/W	-40°C to +85°C
		Program Flash Memory					
D130	EP	Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ 85°C
D130A	EP	Endurance	1000	10K	—	E/W	85°C ≤ TA ≤ 125°C
D131	VPR	VDD for read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VIE	VDD for Block erase	4.5	—	5.5	V	
D132A	VPEW	VDD for write	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D133	TIE	Block Erase cycle time	—	4	8*	ms	VDD > 4.5V
D133A	TPEW	Write cycle time	—	2	4*	ms	
D134	TRETP	Characteristic Retention	40	—	—	year	Provided no other specifications are violated

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Refer to **Section 13.7 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

FIGURE 17-7: BROWN-OUT RESET TIMING

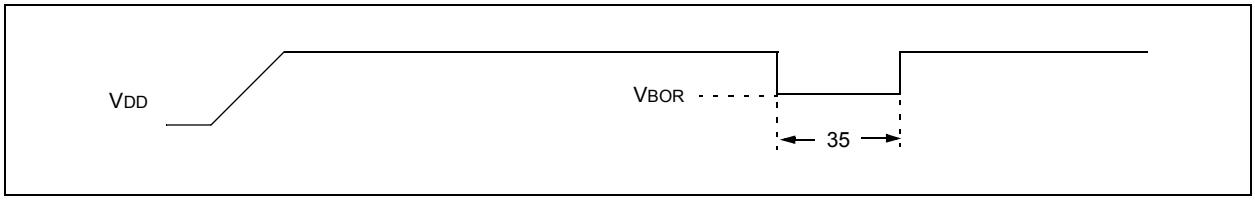


TABLE 17-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	$\overline{MCLR}$ Pulse Width (low)	2000	—	—	ns	VDD = 5V, -40°C to +85°C
31	TWDT	Watchdog Timer Time out Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +85°C
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33	TPWRT	Power-up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O High-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0*	µs	
35	TBOR	Brown-out Reset pulse width	100*	—	—	µs	VDD ≤ VBOR (D005)

Legend: TBD = To Be Determined.

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

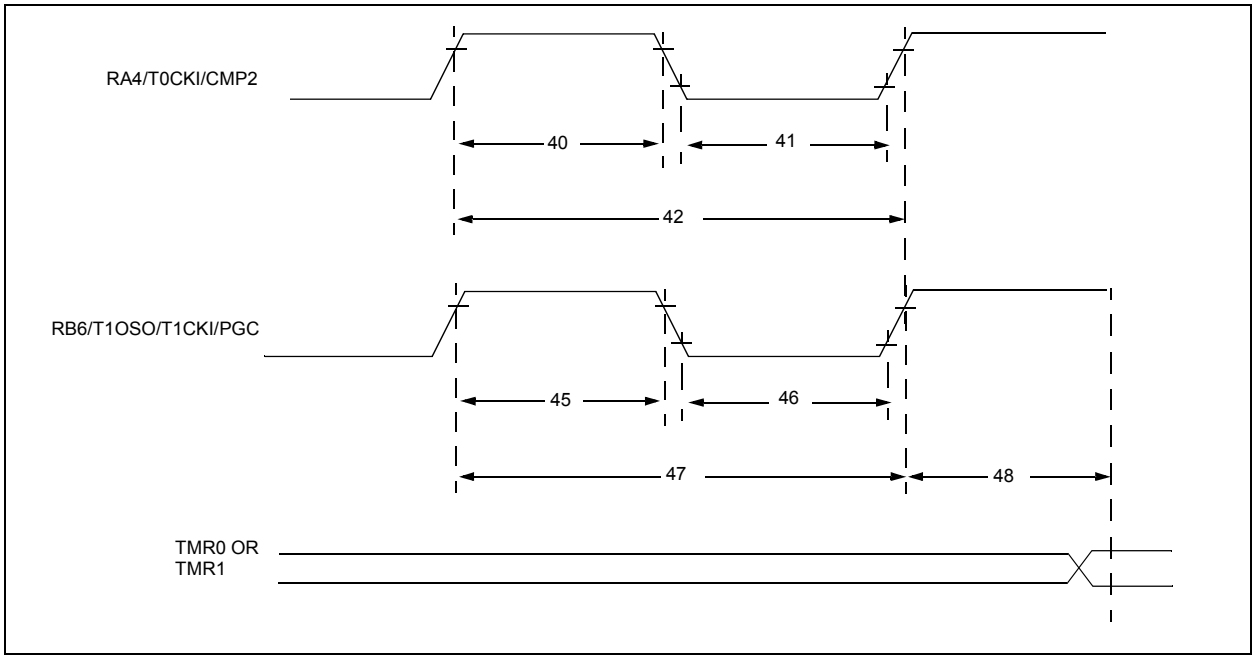


FIGURE 18-14: INTERNAL OSCILLATOR IDD vs. VDD – 4 MHz MODE

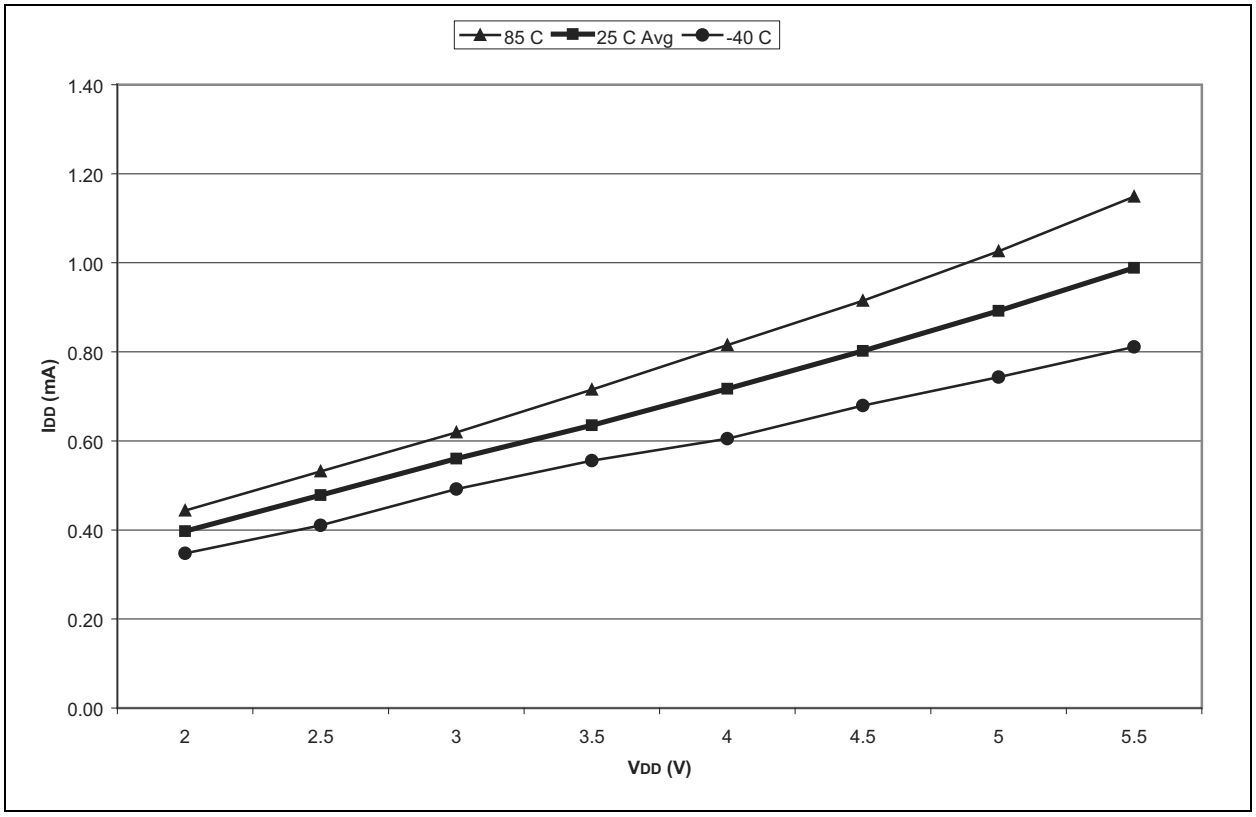
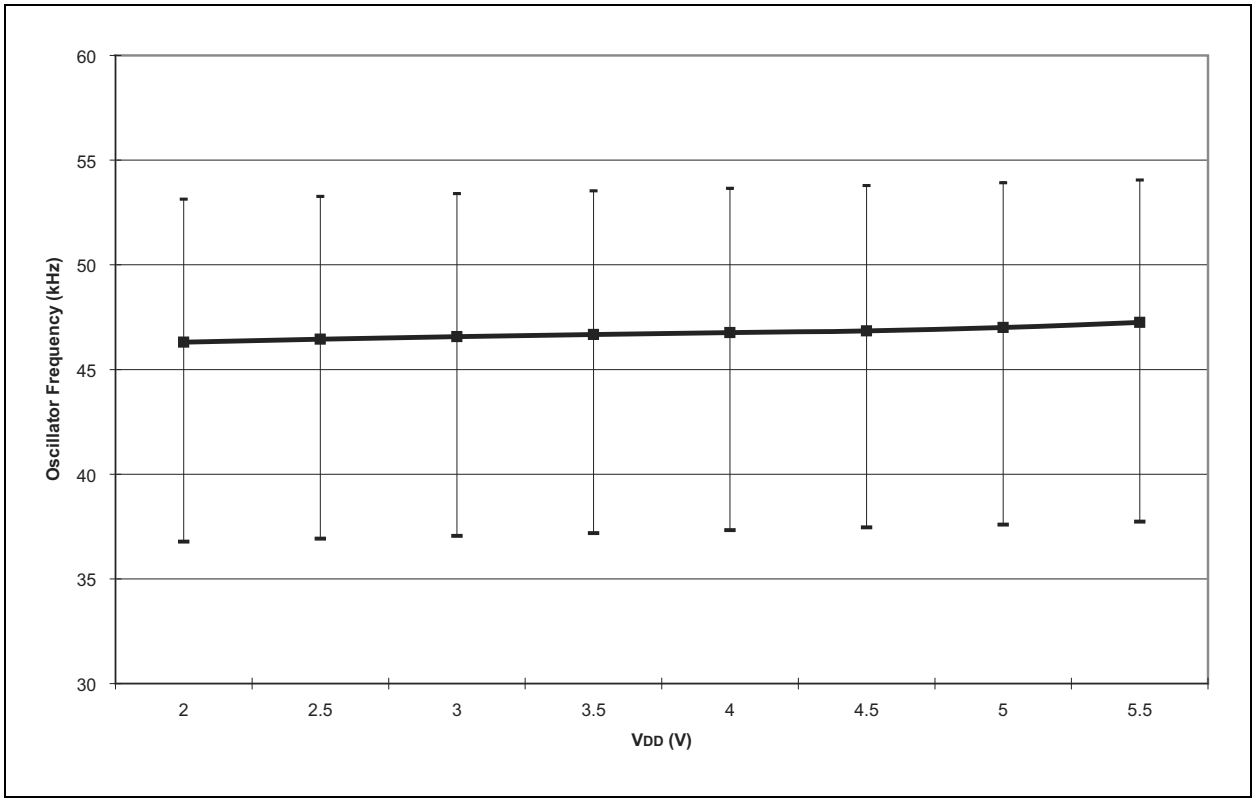


FIGURE 18-15: TYPICAL INTERNAL OSCILLATOR FREQUENCY vs. VDD AT 25°C – SLOW MODE



PRODUCT IDENTIFICATION SYSTEM

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

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device:	PIC16F627A/628A/648A:Standard VDD range 3.0V to 5.5V PIC16F627A/628A/648AT:VDD range 3.0V to 5.5V (Tape and Reel) PIC16LF627A/628A/648A:VDD range 2.0V to 5.5V PIC16LF627A/628A/648AT:VDD range 2.0V to 5.5V (Tape and Reel)
Temperature Range:	I = -40°C to +85°C E = -40°C to+125°C
Package:	P = PDIP SO = SOIC (Gull Wing, 7.50 mm body) SS = SSOP (5.30 mm ML = QFN (28 Lead)

Examples:

a) PIC16F627A - E/P 301 = Extended Temp., PDIP package, 20 MHz, normal VDD limits, QTP pattern #301.

b) PIC16LF627A - I/SO = Industrial Temp., SOIC package, 20 MHz, extended VDD limits.