



Welcome to [E-XFL.COM](https://www.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	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic16f648at-e-ss

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, PIC³² logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F627A/628A/648A family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F627A/628A/648A uses a Harvard architecture in which program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional Von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single-word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single-cycle (200 ns @ 20 MHz) except for program branches.

Table 3-1 lists device memory sizes (Flash, Data and EEPROM).

TABLE 3-1: DEVICE MEMORY LIST

Device	Memory		
	Flash Program	RAM Data	EEPROM Data
PIC16F627A	1024 x 14	224 x 8	128 x 8
PIC16F628A	2048 x 14	224 x 8	128 x 8
PIC16F648A	4096 x 14	256 x 8	256 x 8
PIC16LF627A	1024 x 14	224 x 8	128 x 8
PIC16LF628A	2048 x 14	224 x 8	128 x 8
PIC16LF648A	4096 x 14	256 x 8	256 x 8

The PIC16F627A/628A/648A can directly or indirectly address its register files or data memory. All Special Function Registers (SFR), including the program counter, are mapped in the data memory. The PIC16F627A/628A/648A have an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation, on any register, using any addressing mode. This symmetrical nature and lack of 'special optimal situations' makes programming with the PIC16F627A/628A/648A simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F627A/628A/648A devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

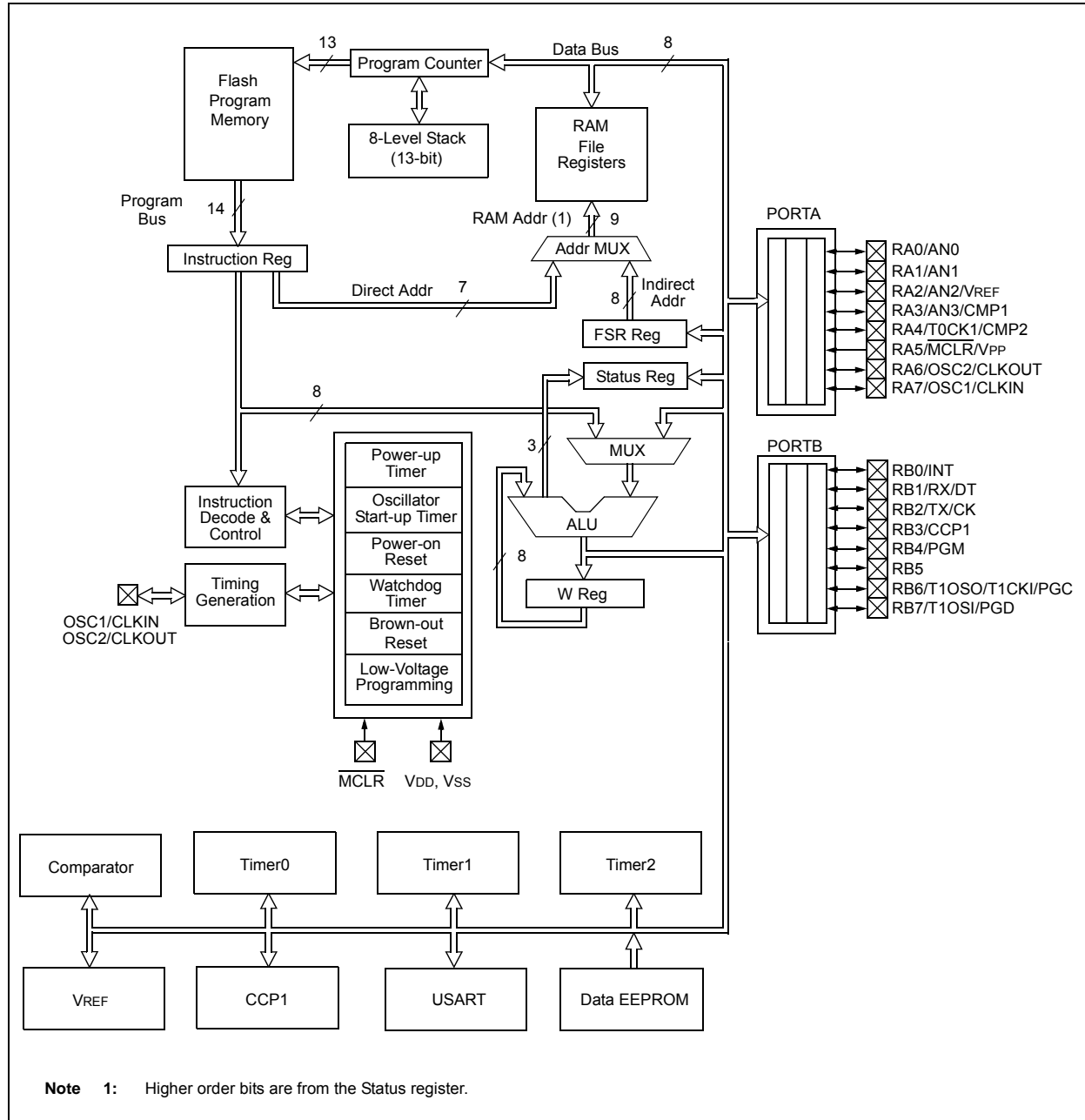
Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the Status Register. The C and DC bits operate as Borrow and Digit Borrow out bits, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.


A simplified block diagram is shown in Figure 3-1, and a description of the device pins in Table 3-2.

Two types of data memory are provided on the PIC16F627A/628A/648A devices. Nonvolatile EEPROM data memory is provided for long term storage of data, such as calibration values, look-up table data, and any other data which may require periodic updating in the field. These data types are not lost when power is removed. The other data memory provided is regular RAM data memory. Regular RAM data memory is provided for temporary storage of data during normal operation. Data is lost when power is removed.

PIC16F627A/628A/648A

FIGURE 3-1: BLOCK DIAGRAM



 Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

PIC16F627A/628A/648A

4.2.2.5 PIR1 Register

This register contains interrupt flag bits.

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

REGISTER 4-5: PIR1 – PERIPHERAL INTERRUPT REGISTER 1 (ADDRESS: 0Ch)

R/W-0	R/W-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0
EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7 **EEIF:** EEPROM Write Operation Interrupt Flag bit
 1 = The write operation completed (must be cleared in software)
 0 = The write operation has not completed or has not been started
- bit 6 **CMIF:** Comparator Interrupt Flag bit
 1 = Comparator output has changed
 0 = Comparator output has not changed
- bit 5 **RCIF:** USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 **TXIF:** USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture Mode
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
Compare Mode
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
PWM Mode
 Unused in this mode
- bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
 1 = TMR2 to PR2 match occurred (must be cleared in software)
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
 1 = TMR1 register overflowed (must be cleared in software)
 0 = TMR1 register did not overflow

Legend:

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

PIC16F627A/628A/648A

4.2.2.6 PCON Register

The PCON register contains flag bits to differentiate between a Power-on Reset, an external MCLR Reset, WDT Reset or a Brown-out Reset.

Note: $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOR}}$ is cleared, indicating a brown-out has occurred. The $\overline{\text{BOR}}$ Status bit is a “don’t care” and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration Word).

REGISTER 4-6: PCON – POWER CONTROL REGISTER (ADDRESS: 8Eh)

U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-0	R/W-x
—	—	—	—	OSCF	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

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

bit 3 **OSCF:** INTOSC Oscillator Frequency bit

1 = 4 MHz typical

0 = 48 kHz typical

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

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

PIC16F627A/628A/648A

NOTES:

PIC16F627A/628A/648A

Follow these steps when setting up an Asynchronous Reception:

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. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH. (**Section 12.1 "USART Baud Rate Generator (BRG)"**).
3. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. Enable the reception by setting bit CREN.
7. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If an OERR error occurred, clear the error by clearing enable bit CREN.

TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

12.3 USART Address Detect Function

12.3.1 USART 9-BIT RECEIVER WITH ADDRESS DETECT

When the RX9 bit is set in the RCSTA register, 9 bits are received and the ninth bit is placed in the RX9D bit of the RCSTA register. The USART module has a special provision for multiprocessor communication. Multiprocessor communication is enabled by setting the ADEN bit (RCSTA<3>) along with the RX9 bit. The port is now programmed such that when the last bit is received, the contents of the Receive Shift Register (RSR) are transferred to the receive buffer, the ninth bit of the RSR (RSR<8>) is transferred to RX9D, and the receive interrupt is set if and only if RSR<8> = 1. This feature can be used in a multiprocessor system as follows:

A master processor intends to transmit a block of data to one of many slaves. It must first send out an address byte that identifies the target slave. An address byte is identified by setting the ninth bit (RSR<8>) to a '1' (instead of a '0' for a data byte). If the ADEN and RX9 bits are set in the slave's RCSTA register, enabling multiprocessor communication, all data bytes will be ignored. However, if the ninth received bit is equal to a '1', indicating that the received byte is an address, the slave will be interrupted and the contents of the RSR register will be transferred into the receive buffer. This allows the slave to be interrupted only by addresses, so that the slave can examine the received byte to see if it is being addressed. The addressed slave will then clear its ADEN bit and prepare to receive data bytes from the master.

When ADEN is enabled (= 1), all data bytes are ignored. Following the Stop bit, the data will not be loaded into the receive buffer, and no interrupt will occur. If another byte is shifted into the RSR register, the previous data byte will be lost.

The ADEN bit will only take effect when the receiver is configured in 9-bit mode (RX9 = 1). When ADEN is disabled (= 0), all data bytes are received and the 9th bit can be used as the parity bit.

The receive block diagram is shown in Figure 12-4.

Reception is enabled by setting bit CREN (RCSTA<4>).

12.3.1.1 Setting up 9-bit mode with Address Detect

Follow these steps when setting up Asynchronous Reception with Address Detect Enabled:

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. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH.
3. Enable asynchronous communication by setting or clearing bit SYNC and setting bit SPEN.
4. If interrupts are desired, then set enable bit RCIE.
5. Set bit RX9 to enable 9-bit reception.
6. Set ADEN to enable address detect.
7. Enable the reception by setting enable bit CREN or SREN.
8. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
9. Read the 8-bit received data by reading the RCREG register to determine if the device is being addressed.
10. If an OERR error occurred, clear the error by clearing enable bit CREN if it was already set.
11. If the device has been addressed (RSR<8> = 1 with address match enabled), clear the ADEN and RCIF bits to allow data bytes and address bytes to be read into the receive buffer and interrupt the CPU.

TABLE 12-8: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

PIC16F627A/628A/648A

12.4.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>) or enable bit CREN (RCSTA<4>). Data is sampled on the RB1/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register, will load bit RX9D with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

Follow these steps when setting up a Synchronous Master Reception:

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. Initialize the SPBRG register for the appropriate baud rate. (**Section 12.1 "USART Baud Rate Generator (BRG)"**).
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
4. Ensure bits CREN and SREN are clear.
5. If interrupts are desired, then set enable bit RCIE.
6. If 9-bit reception is desired, then set bit RX9.
7. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
8. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
9. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If an OERR error occurred, clear the error by clearing bit CREN.

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

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
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EPIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for synchronous master reception.

13.8 Data EEPROM Operation During Code-Protect

When the device is code-protected, the CPU is able to read and write data to the data EEPROM.

TABLE 13-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets
9Ah	EEDATA	EEPROM Data Register								xxxx xxxx	uuuu uuuu
9Bh	EEADR	EEPROM Address Register								xxxx xxxx	uuuu uuuu
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
9Dh	EECON2 ⁽¹⁾	EEPROM Control Register 2								-----	-----

Legend: x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM.

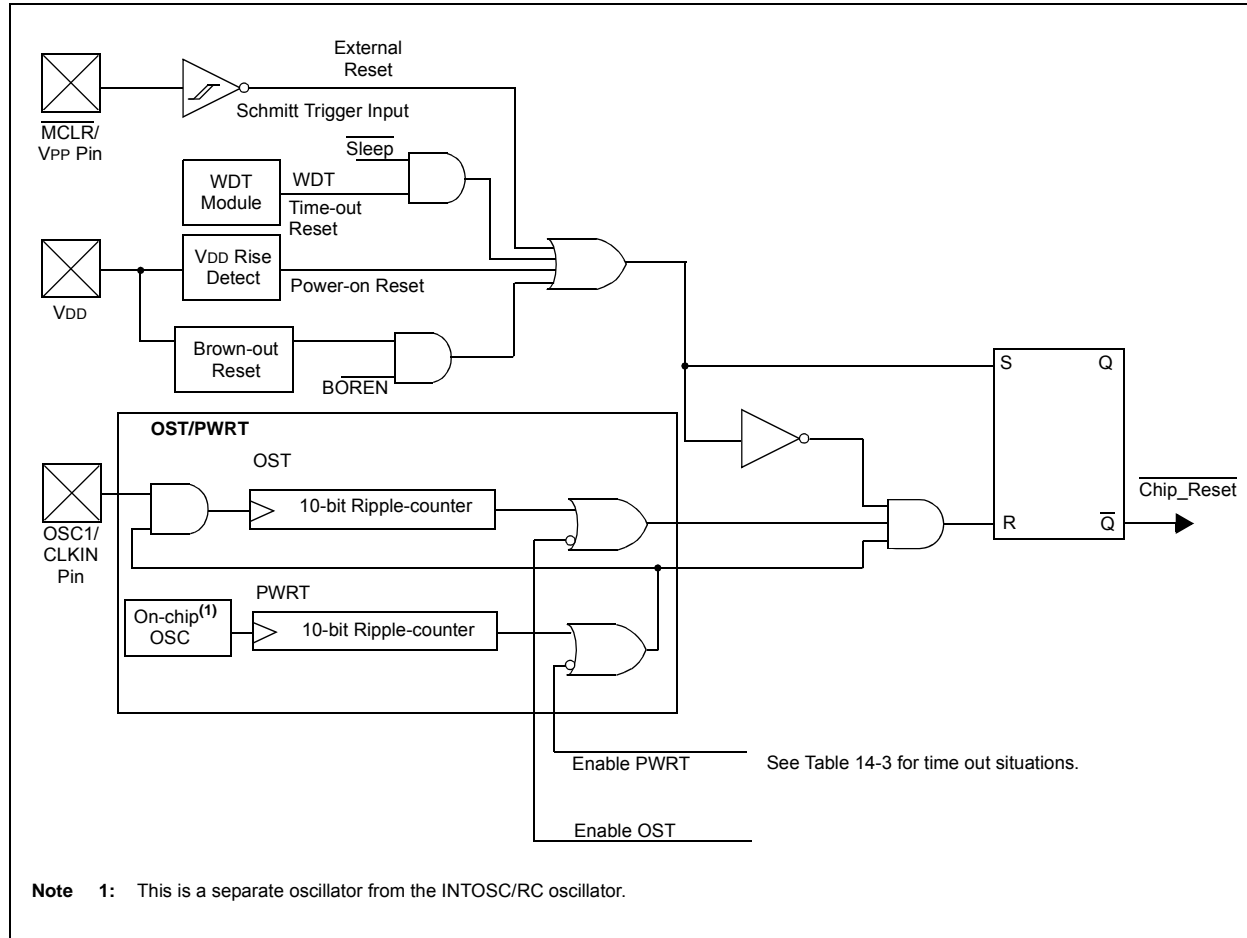
Note 1: EECON2 is not a physical register.

PIC16F627A/628A/648A

NOTES:

PIC16F627A/628A/648A

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



14.5 Interrupts

The PIC16F627A/628A/648A has 10 sources of interrupt:

- External Interrupt RB0/INT
- TMR0 Overflow Interrupt
- PORTB Change Interrupts (pins RB<7:4>)
- Comparator Interrupt
- USART Interrupt TX
- USART Interrupt RX
- CCP Interrupt
- TMR1 Overflow Interrupt
- TMR2 Match Interrupt
- Data EEPROM Interrupt

The Interrupt Control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on Reset.

The “return-from-interrupt” instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enables RB0/INT interrupts.

The INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flag is contained in the special register PIR1. The corresponding interrupt enable bit is contained in special registers PIE1.

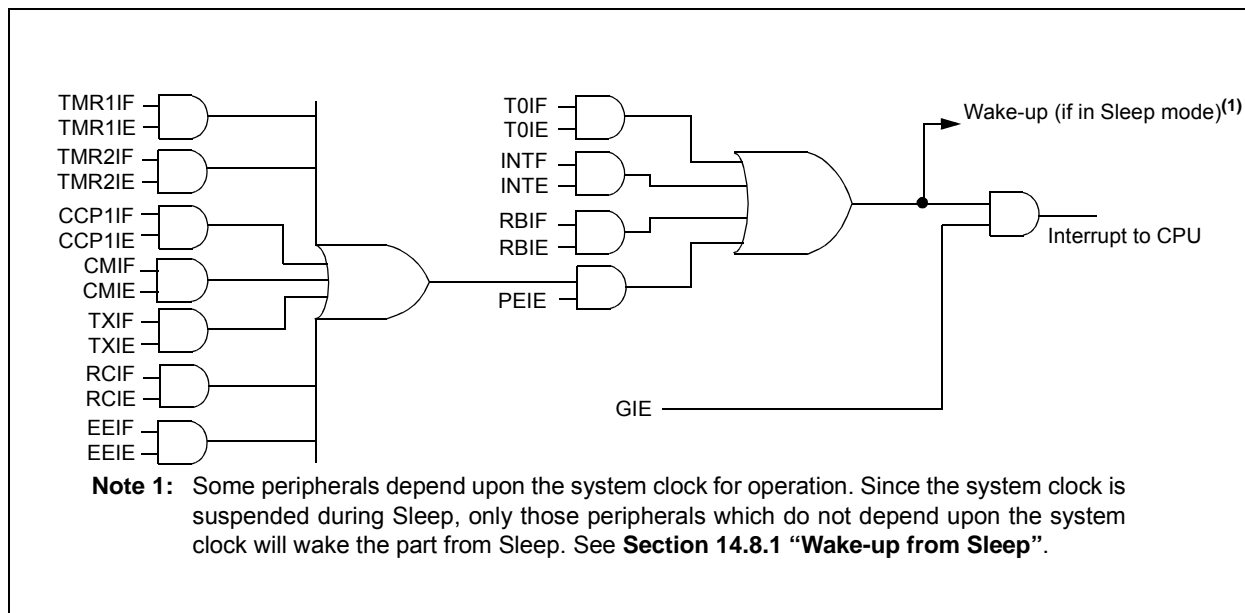
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid RB0/INT recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 14-15). The latency is the same for one or two-cycle instructions. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a NOP in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

FIGURE 14-14: INTERRUPT LOGIC



PIC16F627A/628A/648A

14.5.1 RB0/INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered; either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See **Section 14.8 “Power-Down Mode (Sleep)”** for details on Sleep, and Figure 14-17 for timing of wake-up from Sleep through RB0/INT interrupt.

14.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the Timer0 module, see **Section 6.0 “Timer0 Module”**.

14.5.3 PORTB INTERRUPT

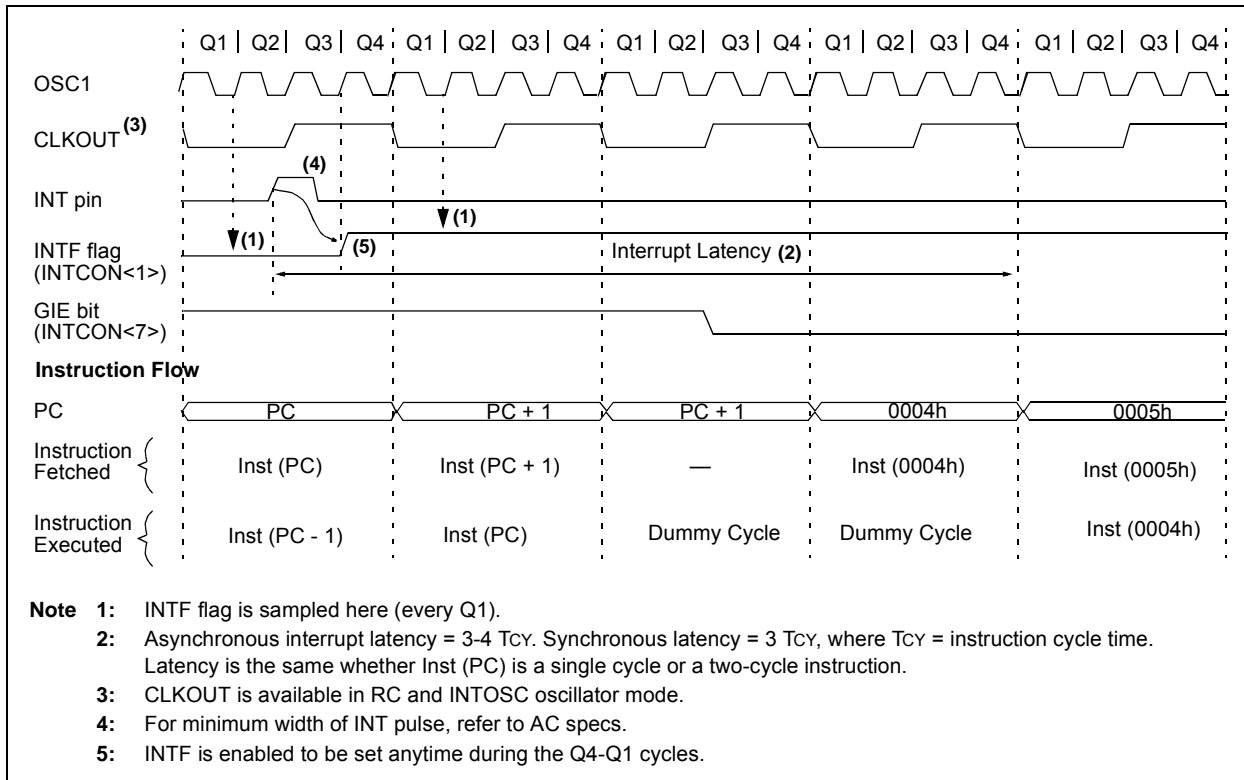
An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<3>) bit. For operation of PORTB (**Section 5.2 “PORTB and TRISB Registers”**).

Note: If a change on the I/O pin should occur when the read operation is being executed (starts during the Q2 cycle and ends before the start of the Q3 cycle), then the RBIF interrupt flag may not get set.

14.5.4 COMPARATOR INTERRUPT

See **Section 10.6 “Comparator Interrupts”** for complete description of comparator interrupts.

FIGURE 14-15: INT PIN INTERRUPT TIMING



16.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

16.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

16.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

16.10 PICkit 3 In-Circuit Debugger/Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

PIC16F627A/628A/648A

17.2 DC Characteristics: PIC16F627A/628A/648A (Industrial) PIC16LF627A/628A/648A (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$ for industrial					
Param No.	LF and F Device Characteristics	Min†	Typ	Max	Units	Conditions	
						VDD	Note
Supply Voltage (VDD)							
D001	LF	2.0	—	5.5	V	—	
	LF/F	3.0	—	5.5	V	—	
Power-down Base Current (IPD)							
D020	LF	—	0.01	0.80	μA	2.0	WDT, BOR, Comparators, VREF and T1OSC: disabled
	LF/F	—	0.01	0.85	μA	3.0	
		—	0.02	2.7	μA	5.0	
Peripheral Module Current (ΔI _{MOD}) ⁽¹⁾							
D021	LF	—	1	2.0	μA	2.0	WDT Current
	LF/F	—	2	3.4	μA	3.0	
		—	9	17.0	μA	5.0	
D022	LF/F	—	29	52	μA	4.5	BOR Current
		—	30	55	μA	5.0	
D023	LF	—	15	22	μA	2.0	Comparator Current (Both comparators enabled)
	LF/F	—	22	37	μA	3.0	
		—	44	68	μA	5.0	
D024	LF	—	34	55	μA	2.0	VREF Current
	LF/F	—	50	75	μA	3.0	
		—	80	110	μA	5.0	
D025	LF	—	1.2	2.0	μA	2.0	T1Osc Current
	LF/F	—	1.3	2.2	μA	3.0	
		—	1.8	2.9	μA	5.0	
Supply Current (IDD)							
D010	LF	—	10	15	μA	2.0	Fosc = 32 kHz LP Oscillator Mode
	LF/F	—	15	25	μA	3.0	
		—	28	48	μA	5.0	
D011	LF	—	125	190	μA	2.0	Fosc = 1 MHz XT Oscillator Mode
	LF/F	—	175	340	μA	3.0	
		—	320	520	μA	5.0	
D012	LF	—	250	350	μA	2.0	Fosc = 4 MHz XT Oscillator Mode
	LF/F	—	450	600	μA	3.0	
		—	710	995	μA	5.0	
D012A	LF	—	395	465	μA	2.0	Fosc = 4 MHz INTOSC
	LF/F	—	565	785	μA	3.0	
		—	0.895	1.3	mA	5.0	
D013	LF/F	—	2.5	2.9	mA	4.5	Fosc = 20 MHz HS Oscillator Mode
		—	2.75	3.3	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.

PIC16F627A/628A/648A

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

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_a \leq +125^{\circ}\text{C}$ 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 (ΔI _{MOD}) ⁽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.

PIC16F627A/628A/648A

FIGURE 18-2: TYPICAL BASELINE IPD vs. VDD (85°C)

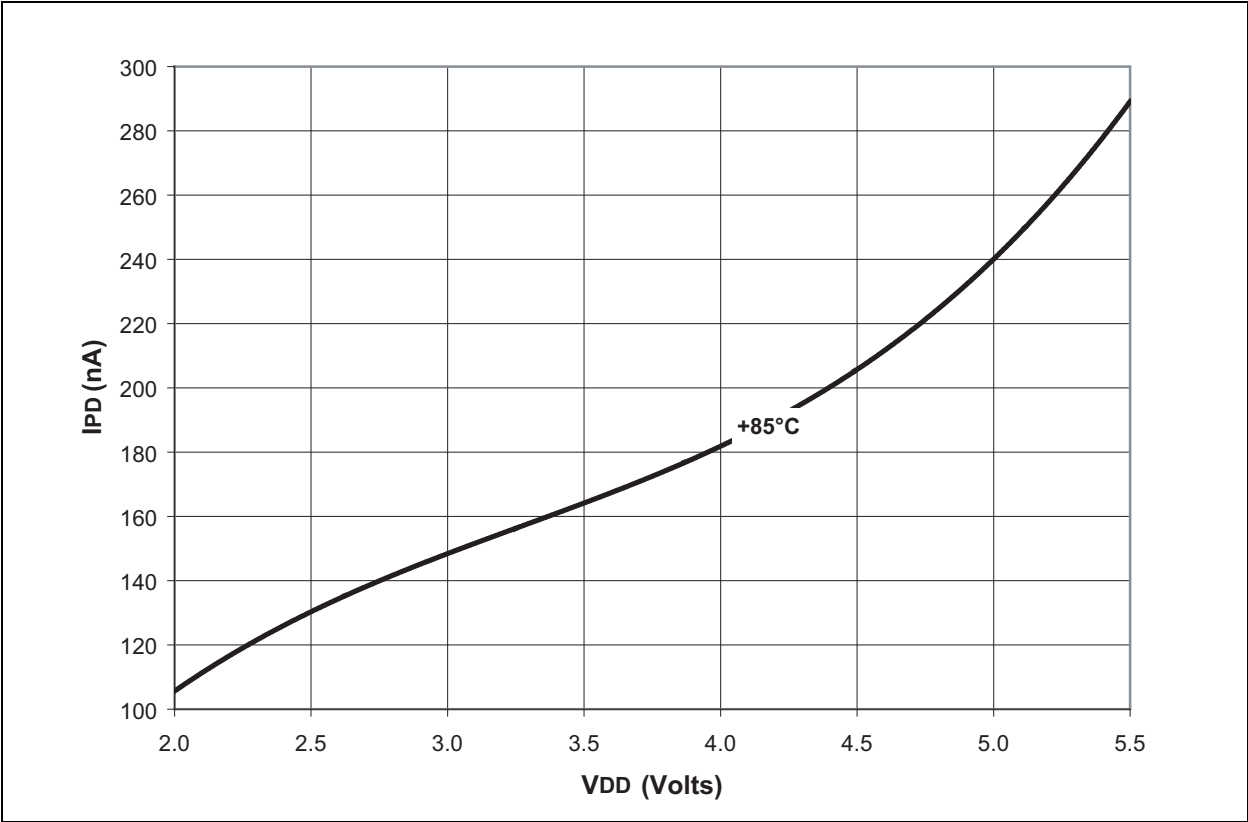


FIGURE 18-3: TYPICAL BASELINE CURRENT IPD vs. VDD (125°C)

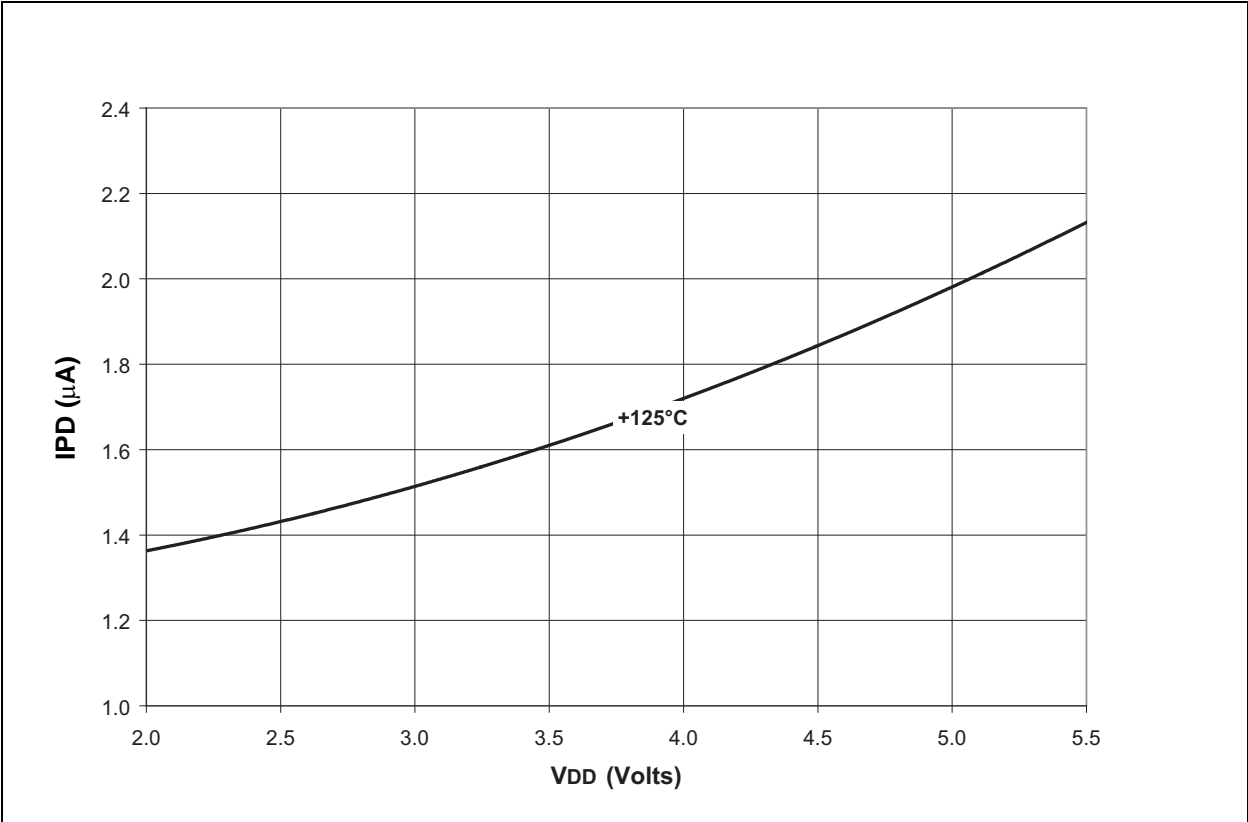
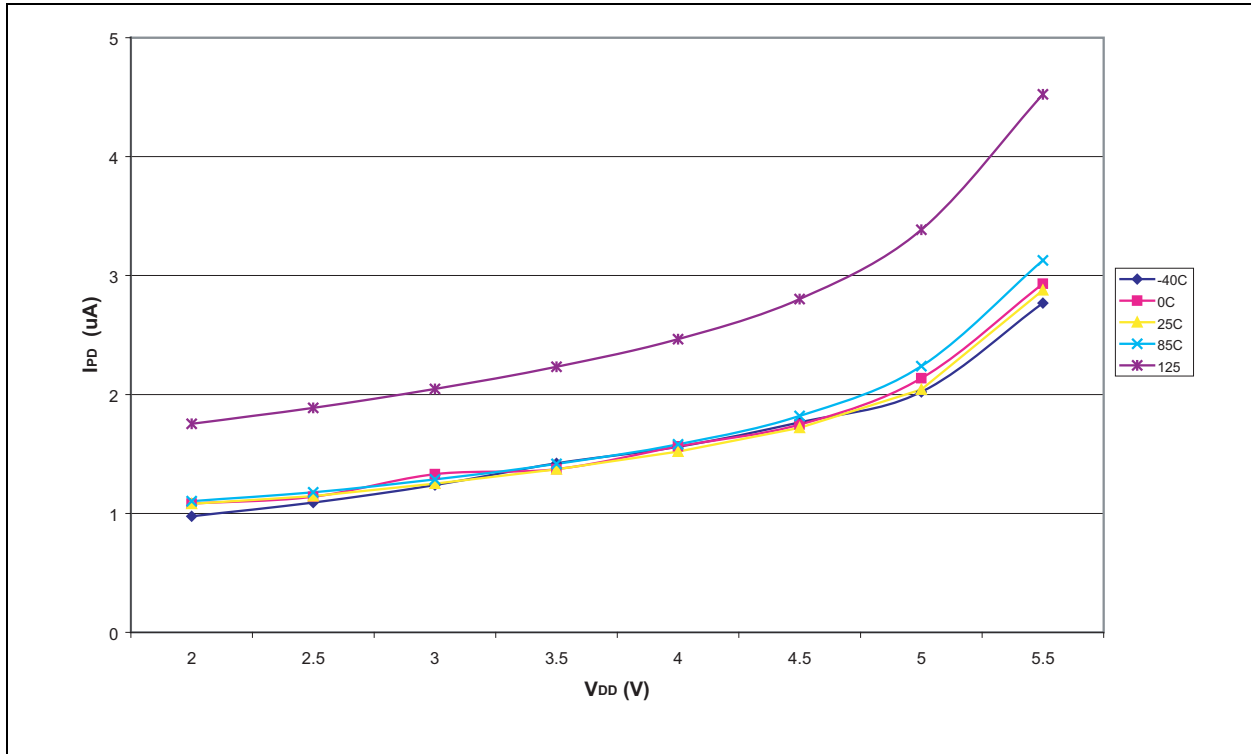


FIGURE 18-8: AVERAGE I_{PD_TIMER1}



**FIGURE 18-9: TYPICAL INTERNAL OSCILLATOR FREQUENCY vs. TEMPERATURE
 $V_{DD} = 5$ VOLTS**

