

## OTP 8-Bit CMOS MCU with LIN Transceiver

### Devices included in this Data Sheet:

- PIC16C432

### High Performance RISC CPU:

- Only 35 instructions to learn
- All single cycle instructions (200 ns), except for program branches which are two-cycle
- Operating speed:
  - DC - 20 MHz clock input
  - DC - 200 ns instruction cycle

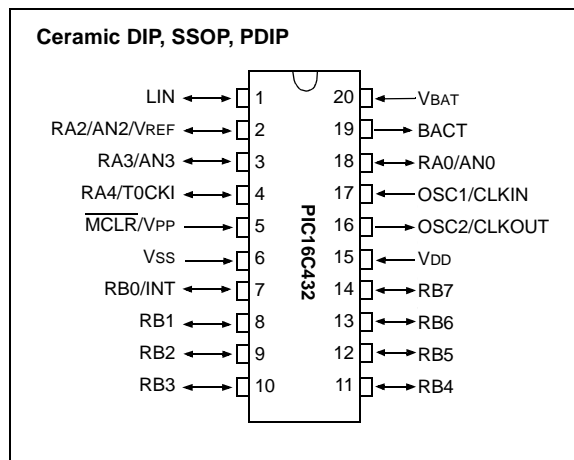
Device	Program Memory	RAM Data Memory
PIC16C432	2K x 14	128 x 8

- Interrupt capability
- 16 special function hardware registers
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

### Peripheral Features:

- 12 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs can be output signals
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Integrated LIN Transceiver
- Wake-up on bus activity
- 12V battery operation for Transceiver
- Thermal shutdown for Transceiver
- Ground loss protection

### PIN DIAGRAM



### Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) (via two pins)
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Four user programmable ID locations

### CMOS Technology:

- Low power, high speed CMOS EPROM/HV-CMOS technology
- Fully static design
- Operating voltage range
  - 4.5V to 5.5V
- Industrial and extended temperature range

# PIC16C432

## Table of Contents

1.0	General Description.....	3
2.0	PIC16C432 Device Varieties.....	5
3.0	Memory Organization.....	7
4.0	I/O Ports.....	17
5.0	LIN Transceiver.....	23
6.0	Timer0 Module.....	27
7.0	Comparator Module.....	33
8.0	Voltage Reference Module.....	41
9.0	Special Features of the CPU.....	43
10.0	Instruction Set Summary.....	59
11.0	Development Support.....	73
12.0	Electrical Specifications.....	79
13.0	DC and AC Characteristics Graphs and Tables.....	91
14.0	Packaging Information.....	93
	Appendix A: Code for LIN Communication.....	97
	Index:.....	99
	On-Line Support.....	101
	Systems Information and Upgrade Hot Line.....	101
	Reader Response.....	102
	Product Identification System.....	103

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@mail.microchip.com](mailto:docerrors@mail.microchip.com) or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com/cn](http://www.microchip.com/cn) to receive the most current information on all of our products.

## 1.0 GENERAL DESCRIPTION

The PIC16C432 is a 20-pin EPROM-based member of the versatile PIC® family of low cost, high performance, CMOS, fully-static, 8-bit microcontrollers with an integrated LIN transceiver.

The LIN physical layer is implemented in hardware with a voltage range from 0V to 18V, with a 40V transient capability. The LIN protocol is to be implemented in firmware, which enables flexibility with future revisions of the LIN protocol.

All PIC® microcontrollers employ an advanced RISC architecture. The PIC16C432 device has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C432 microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC16C432 has 12 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler. In addition, the PIC16C432 adds two analog comparators with a programmable on-chip voltage reference module. The comparator module is ideally suited for applications requiring a low cost analog interface (e.g., battery chargers, threshold detectors, white goods controllers, etc.).

PIC16C432 devices have special features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power savings. The user can wake-up the chip from SLEEP through several external and internal interrupts and RESET.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV erasable CERDIP packaged version is ideal for code development, while the cost effective One-Time-Programmable (OTP) version is suitable for production in any volume.

A simplified block diagram of the PIC16C432 is shown in Figure 4-1.

The PIC16C432 series fits perfectly in automotive and industrial applications, which require LIN as a communication platform. The EPROM technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16C432 very versatile.

### 1.1 Development Support

The PIC16C432 family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost development programmer and a full-featured programmer. A "C" compiler is also available.

# PIC16C432

---

---

NOTES:

## 2.0 PIC16C432 DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C432 Product Identification System section at the end of this data sheet.

### 2.1 UV Erasable Devices

The UV erasable version, offered in the CERDIP package is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PRO MATE<sup>®</sup> programmers support programming of the PIC16C432.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications. In addition to the program memory, the configuration bits must also be programmed.

### 2.3 Quick-Turn-Programming (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices, but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

### 2.4 Serialized Quick-Turn-Programming (SQTP<sup>SM</sup>) Devices

Microchip offers a unique programming service where a few user defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry code, password or ID number.

# PIC16C432

---

---

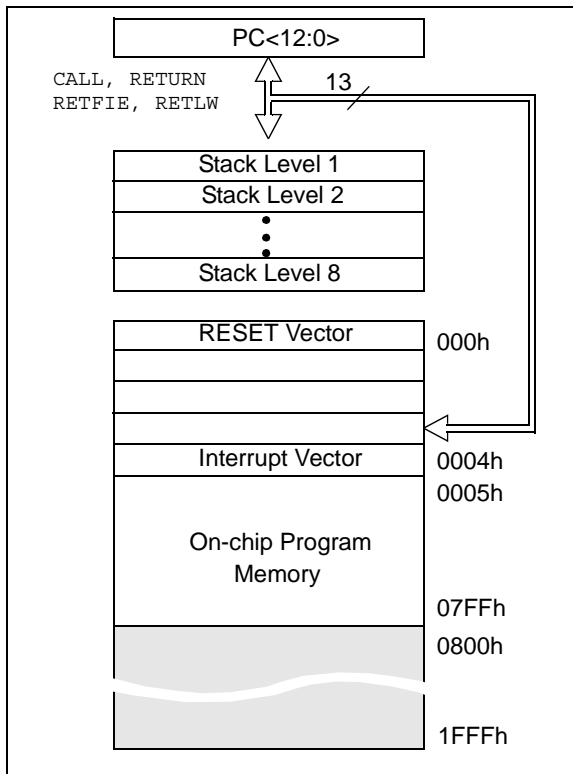
NOTES:

## 3.0 MEMORY ORGANIZATION

### 3.1 Program Memory Organization

The PIC16C432 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 2K x 14 (0000h - 07FFh) are implemented for the PIC16C432. Accessing a location above these boundaries will cause a wrap-around within the first 2K x 14 space. The RESET Vector is at 0000h and the Interrupt Vector is at 0004h (Figure 3-1).

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C432**



### 3.2 Data Memory Organization

The data memory (Figure 3-2) is partitioned into two Banks, which contain the General Purpose Registers and the Special Function Registers. Bank 0 is selected when the RP0 bit is cleared. Bank 1 is selected when the RP0 bit (STATUS <5>) is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations 20-7Fh (Bank 0) and A0-BFh (Bank 1) are General Purpose Registers implemented as static RAM. Some special purpose registers are mapped in Bank 1. In the microcontroller, address space F0h-FFh (Bank 1) is mapped to 70-7Fh (Bank 0) as common RAM.

#### 3.2.1 GENERAL PURPOSE REGISTER FILE


The register file is organized as 128 x 8 in the PIC16C432. Each is accessed either directly or indirectly through the File Select Register FSR (Section 3.4).

# PIC16C432

**FIGURE 3-2: DATA MEMORY MAP FOR THE PIC16C432**

File Address			File Address
00h	INDF <sup>(1)</sup>	INDF <sup>(1)</sup>	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h			88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh			8Fh
10h		LININTF	90h
11h			91h
12h			92h
13h			93h
14h			94h
15h			95h
16h			96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh			9Eh
1Fh	CMCON	VRCON	9Fh
20h	General Purpose Register	General Purpose Register	A0h
			BFh
		C0h	
		F0h	
		Accesses 70h-7Fh	
7Fh			FFh

Bank 0                      Bank 1

 Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

## 3.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (Table 3-1). These registers are static RAM.

The special registers can be classified into two sets (core and peripheral). The Special Function Registers 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 3-1: SPECIAL REGISTERS FOR THE PIC16C432**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Value on all other RESETS <sup>(1)</sup>
<b>Bank 0</b>											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	16
01h	TMR0	Timer0 Module's Register								xxxx xxxx	27
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	15
03h	STATUS	IRP <sup>(2)</sup>	RP1 <sup>(2)</sup>	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	10
04h	FSR	Indirect data memory address pointer								xxxx xxxx	16
05h	PORTA	—	—	—	RA4	RA3	RA2	LINRX	RA0	---x 0000	17
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	20
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter				---0 0000	15	
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	12
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	-0-- ----	13
0Dh-1Eh	—	Unimplemented								—	—
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	33
<b>Bank 1</b>											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	16
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	11
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	15
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	10
84h	FSR	Indirect data memory address pointer								xxxx xxxx	16
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(3)</sup>	TRISA0	---1 1111	17
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	20
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter				---0 0000	15	
8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	12
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	-0-- ----	13
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOD}$	---- -0x	14
8Fh-9Eh	—	Unimplemented								—	—
90h	LININTF	—	—	—	—	—	LINTX	—	LINVDD	---- -1-1	23
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	41

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

**Note 1:** Other (non power-up) RESETS include MCLR Reset, Brown-out Reset and Watchdog Timer Reset during normal operation.

**Note 2:** IRP & RP1 bits are reserved; always maintain these bits clear.

**Note 3:** TLINRX must set to '1' at all times.

# PIC16C432

## 3.2.2.1 STATUS Register

The STATUS register, shown in Register 3-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the  $\overline{TO}$  and  $\overline{PD}$  bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000uu1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any STATUS bit. For other instructions, not affecting any STATUS bits, see the "Instruction Set Summary".

**Note 1:** The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C432 and should be programmed as '0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

**2:** The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

### REGISTER 3-1: STATUS REGISTER (ADDRESS 03h OR 83h)

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

- bit 7 **IRP:**  
The IRP bit is reserved on the PIC16C432, always maintain this bit clear
- bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)  
11 = Bank 3 (180h - 1FFh)  
10 = Bank 2 (100h - 17Fh)  
01 = Bank 1 (80h - FFh)  
00 = Bank 0 (00h - 7Fh)  
Each bank is 128 bytes. The RP1 bit is reserved, always maintain this bit clear.
- bit 4  **$\overline{TO}$ :** Timeout bit  
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction  
0 = A WDT timeout occurred
- bit 3  **$\overline{PD}$ :** Power-down bit  
1 = After power-up or by the `CLRWDT` instruction  
0 = By execution of the `SLEEP` instruction
- bit 2 **z:** Zero bit  
1 = The result of an arithmetic or logic operation is zero  
0 = The result of an arithmetic or logic operation is not zero
- bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for borrow the polarity is reversed)  
1 = A carry-out from the 4th low order bit of the result occurred  
0 = No carry-out from the 4th low order bit of the result
- bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)  
1 = A carry-out from the Most Significant bit of the result occurred  
0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

**Legend:**

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

## 3.2.2.2 OPTION Register

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0 and the weak pull-ups on PORTB.

**Note:** To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1).

### REGISTER 3-2: OPTION REGISTER (ADDRESS 81h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	<b>RBPU</b>	<b>INTEDG</b>	<b>T0CS</b>	<b>T0SE</b>	<b>PSA</b>	<b>PS2</b>	<b>PS1</b>	<b>PS0</b>
bit7								bit0

- bit 7 **RBPU:** PORTB Pull-up Enable bit  
1 = PORTB pull-ups are disabled  
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG:** Interrupt Edge Select bit  
1 = Interrupt on rising edge of RB0/INT pin  
0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS:** TMR0 Clock Source Select bit  
1 = Transition on RA4/T0CKI pin  
0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE:** TMR0 Source Edge Select bit  
1 = Increment on high-to-low transition on RA4/T0CKI pin  
0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA:** Prescaler Assignment bit  
1 = Prescaler is assigned to the WDT  
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

**Legend:**

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

# PIC16C432

## 3.2.2.3 INTCON Register

The INTCON register is a readable and writable register which contains the various enable and flag bits for all interrupt sources, except the comparator module. See Section 3.2.2.4 and Section 3.2.2.5 for a description of the comparator enable and 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>).

### REGISTER 3-3: INTCON REGISTER (ADDRESS 0Bh OR 8Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7								bit0

- bit 7     **GIE:** Global Interrupt Enable bit  
1 = Enables all unmasked interrupts  
0 = Disables all interrupts
- bit 6     **PEIE:** Peripheral Interrupt Enable bit  
1 = Enables all un-masked peripheral interrupts  
0 = Disables all peripheral interrupts
- bit 5     **TOIE:** TMR0 Overflow Interrupt Enable bit  
1 = Enables the TMR0 interrupt  
0 = Disables the TMR0 interrupt
- bit 4     **INTE:** RB0/INT External Interrupt Enable bit  
1 = Enables the RB0/INT external interrupt  
0 = Disables the RB0/INT external interrupt
- bit 3     **RBIE:** RB Port Change Interrupt Enable bit  
1 = Enables the RB port change interrupt  
0 = Disables the RB port change interrupt
- bit 2     **TOIF:** TMR0 Overflow Interrupt Flag bit  
1 = TMR0 register has overflowed (must be cleared in software)  
0 = TMR0 register did not overflow
- bit 1     **INTF:** RB0/INT External Interrupt Flag bit  
1 = The RB0/INT external interrupt occurred (must be cleared in software)  
0 = The RB0/INT external interrupt did not occur
- bit 0     **RBIF:** RB Port Change Interrupt Flag bit  
1 = When at least one of the RB<7:4> pins changed state (must be cleared in software)  
0 = None of the RB<7:4> pins have changed state

Legend:

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

## 3.2.2.4 PIE1 Register

This register contains the individual enable bit for the comparator interrupt.

### REGISTER 3-4: PIE1 REGISTER (ADDRESS 8CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIE	—	—	—	—	—	—
bit7							bit0

- bit 7     **Unimplemented:** Read as '0'
- bit 6     **CMIE:** Comparator Interrupt Flag bit  
           1 = Enables the Comparator interrupt  
           0 = Disables the Comparator interrupt
- bit 5-0   **Unimplemented:** Read as '0'

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

## 3.2.2.5 PIR1 Register

This register contains the individual flag bit for the comparator interrupt.

**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 3-5: PIR1 REGISTER (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIF	—	—	—	—	—	—
bit7							bit0

- bit 7     **Unimplemented:** Read as '0'
- bit 6     **CMIF:** Comparator Interrupt Flag bit  
           1 = Comparator input has changed  
           0 = Comparator input has not changed
- bit 5-0   **Unimplemented:** Read as '0'

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

# PIC16C432

## 3.2.2.6 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:**  $\overline{\text{BOD}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if  $\overline{\text{BOD}}$  is cleared, indicating a brown-out has occurred. The  $\overline{\text{BOD}}$  status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by programming BODEN bit in the configuration word).

### REGISTER 3-6: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOD}}$
bit7						bit0	

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **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 **BOD:** 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)

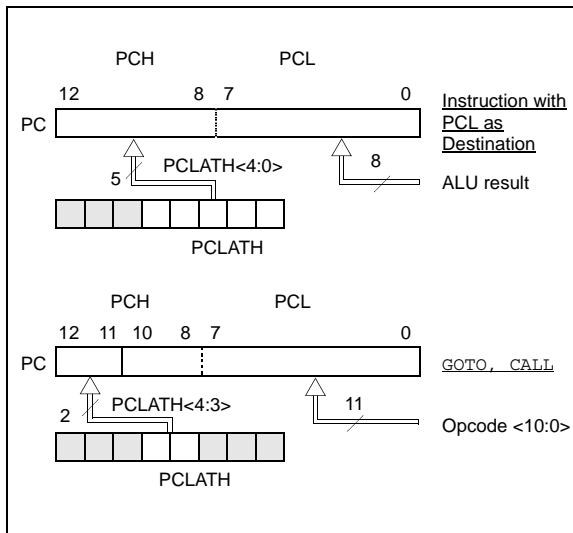
**Legend:**

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

## 3.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte ( $PC<12:8>$ ) is not directly readable or writable and comes from PCLATH. On any RESET, the PC is cleared. Figure 3-3 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL ( $PCLATH<4:0> \rightarrow PCH$ ). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction ( $PCLATH<4:3> \rightarrow PCH$ ).

**FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS**



### 3.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter ( $ADDWF PCL$ ). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the Application Note, "Implementing a Table Read" (AN556).

### 3.3.2 STACK

The PIC16C432 family has an 8 level deep x 13-bit wide hardware stack (Figure 3-1 and Figure 3-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth PUSH overwrites the value that was stored from the first PUSH. The tenth PUSH overwrites the second PUSH (and so on).

**Note 1:** There are no STATUS bits to indicate stack overflow or stack underflow conditions.

**2:** There are no instruction/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or the vectoring to an interrupt address.

# PIC16C432

## 3.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 3-4. However, IRP is not used in the PIC16C432.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 3-1.

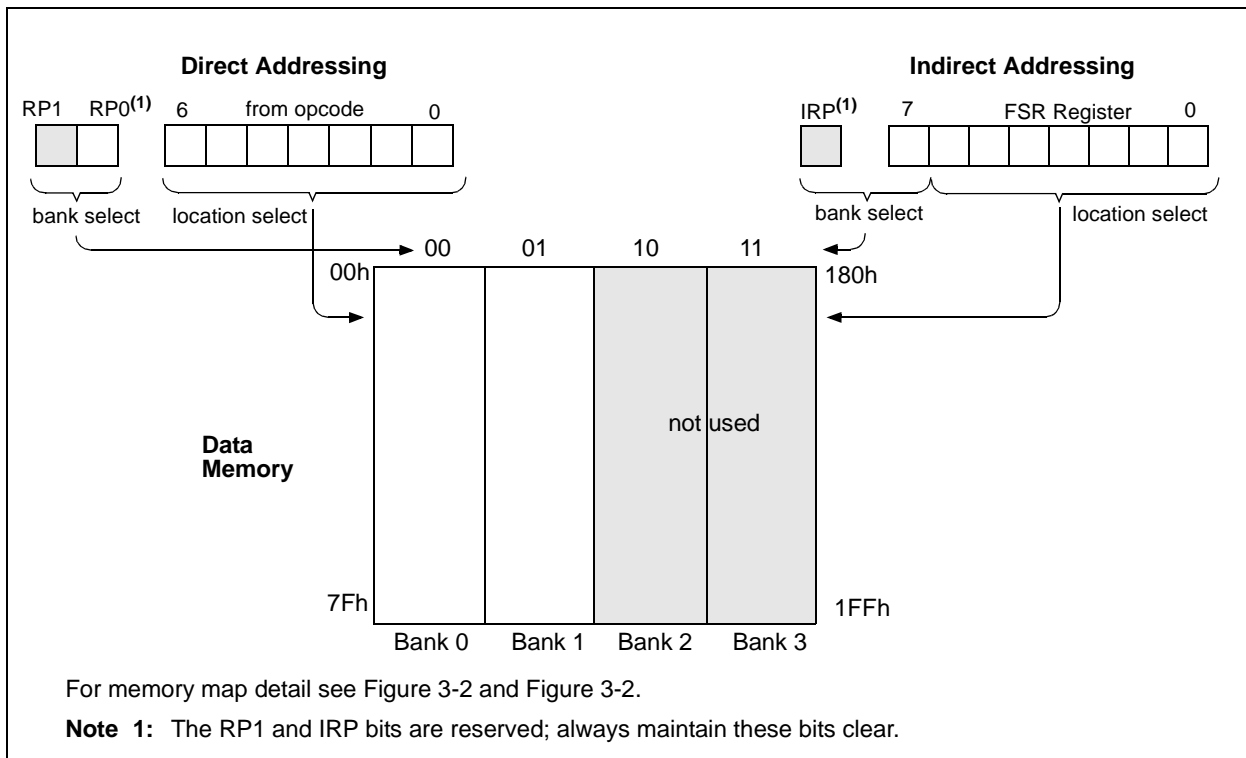
### EXAMPLE 3-1: INDIRECT ADDRESSING

```

movlw 0x20 ;initialize pointer
movwf FSR ;to RAM
NEXT   clrf INDF ;clear INDF register
       incf FSR ;inc pointer
       btfss FSR,4 ;all done?
       goto NEXT ;no clear next
                               ;yes continue
    
```

CONTINUE:

FIGURE 3-4: DIRECT/INDIRECT ADDRESSING PIC16C432





## 4.0 I/O PORTS

The PIC16C432 parts have two ports, PORTA and PORTB. Some pins for these I/O ports are multiplexed with an alternate pins function for 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.

### 4.1 PORTA and TRISA Registers

PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers), which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (Comparator Control Register) register and the VRCON (Voltage Reference Control Register) register. When selected as a comparator input, these pins will read as '0's.

**Note:** On RESET, the TRISA register is set to all inputs. The digital inputs are disabled and the comparator inputs are forced to ground, to reduce excess current consumption.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

The RA2 pin will also function as the output for the voltage reference. When in this mode, the VREF pin is a very high impedance output. The user must configure TRISA<2> bit as an input and use high impedance loads.

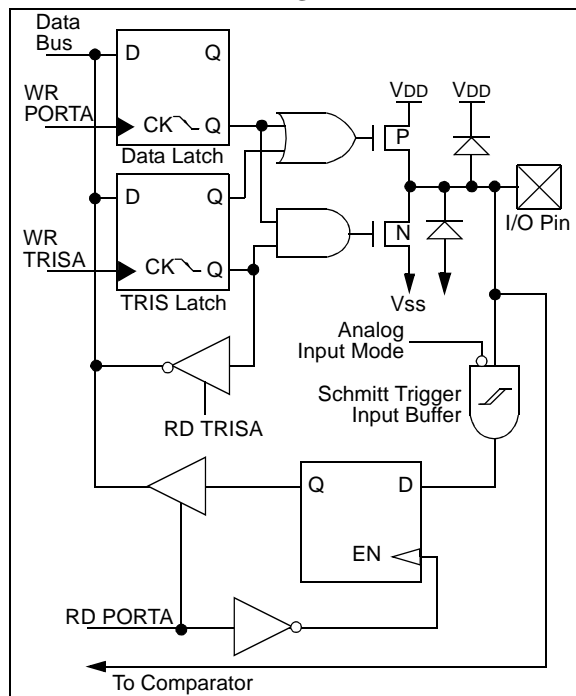
In one of the comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be cleared to enable outputs to use this function.

#### EXAMPLE 4-1: INITIALIZING PORTA

```

CLRFB   PORTA           ;Initialize PORTA by setting
                        ;output data latches
MOVLW   0X07           ;Turn comparators off and
MOVWF   CMCON          ;enable pins for I/O
                        ;functions
BSF     STATUS,        ;Select Bank1
        RP0
MOVLW   0x1F           ;Value used to initialize
                        ;data direction
MOVWF   TRISA          ;Set RA<4:0> as inputs
                        ;TRISA<7:5> are always
                        ;read as '0'.
    
```

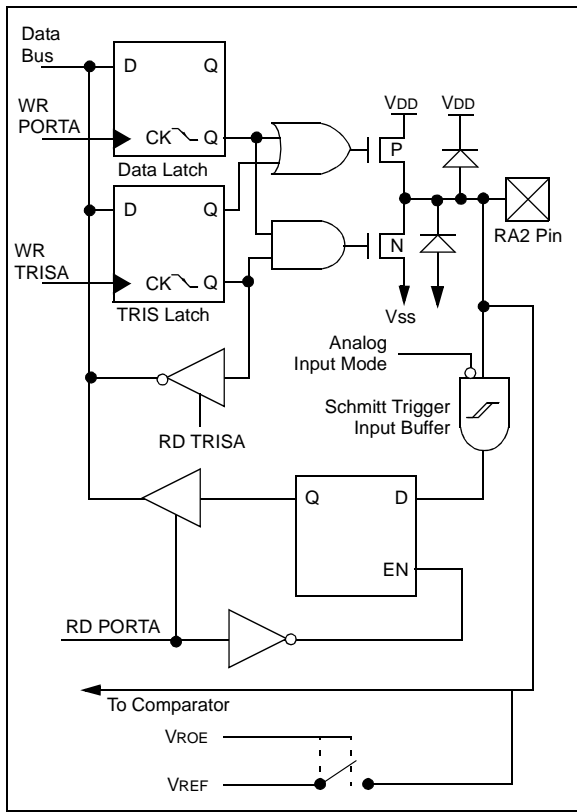
**FIGURE 4-1: BLOCK DIAGRAM OF RA0 PINS**



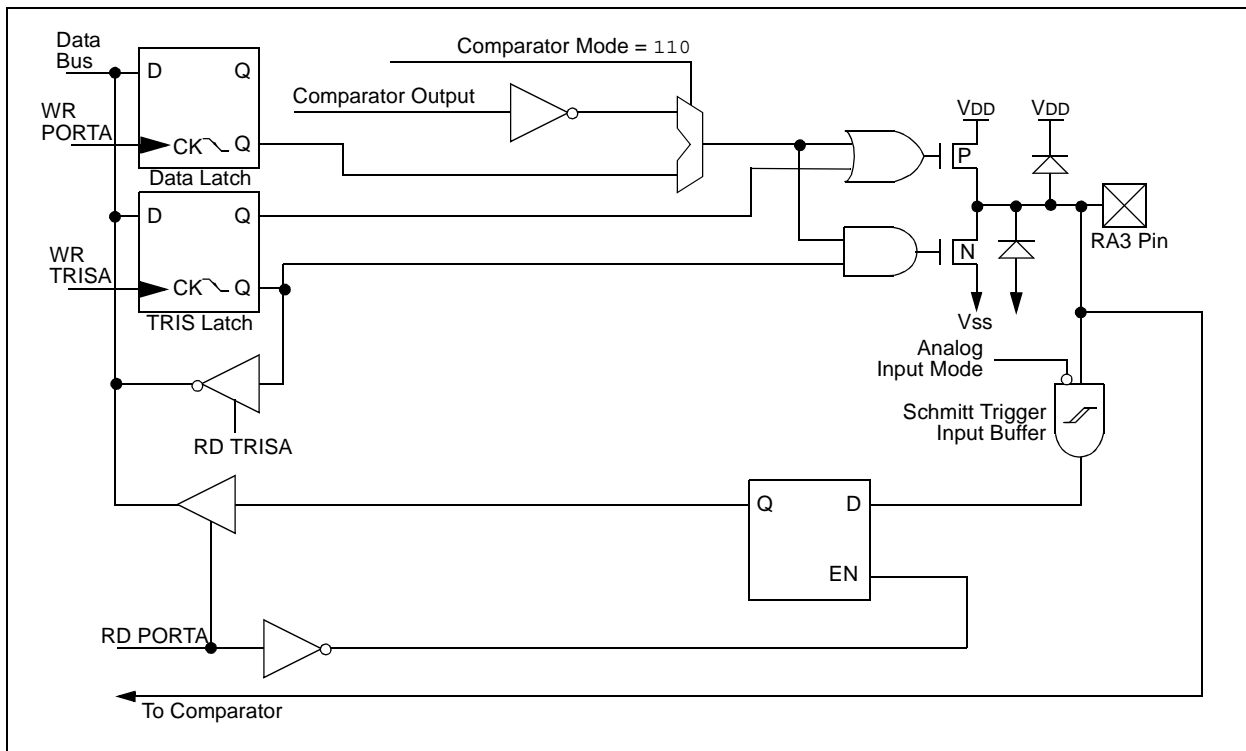
**Note 1:** BACT pin is an output and must be left open if unused.

# PIC16C432

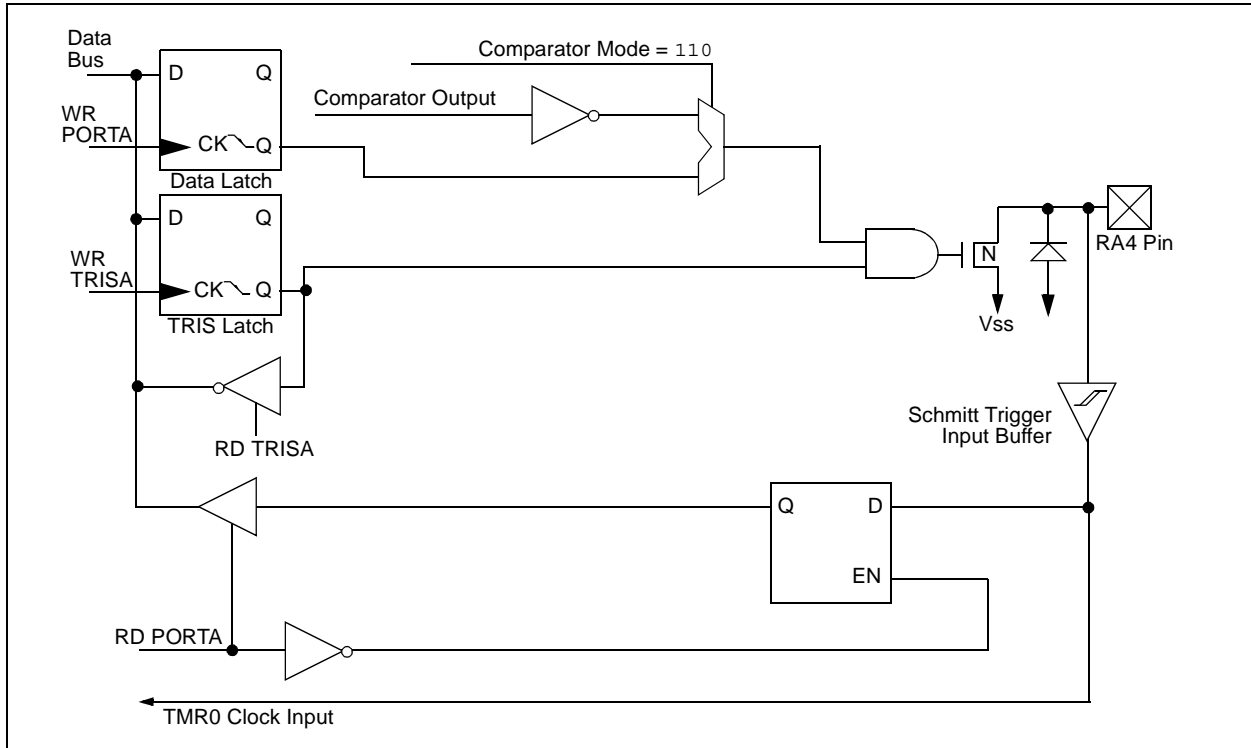
**FIGURE 4-2: BLOCK DIAGRAM OF RA2 PIN**



**FIGURE 4-3: BLOCK DIAGRAM OF RA3 PIN**



**FIGURE 4-4: BLOCK DIAGRAM OF RA4 PIN**



**TABLE 4-1: PORTA FUNCTIONS**

Name	Bit #	Buffer Type	Function
RA0/AN0	bit0	ST	Input/output or comparator input.
LINRX	bit1	ST	LIN receive pin.
RA2/AN2/VREF	bit2	ST	Input/output or comparator input or VREF output.
RA3/AN3	bit3	ST	Input/output or comparator input/output.
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0 or comparator output. Output is open drain type.

Legend: ST = Schmitt Trigger input

**TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

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
05h	PORTA	—	—	—	RA4	RA3	RA2	LINRX	RA0	---x 0000	---u 0000
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(2)</sup>	TRISA0	---1 1111	---1 1111
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

Legend: — = Unimplemented locations, read as '0', x = unknown, u = unchanged

**Note 1:** Shaded bits are not used by PORTA.

**Note 2:** TLINRX must be set to '1' at all times.

# PIC16C432

## 4.2 PORTB and TRISB Registers

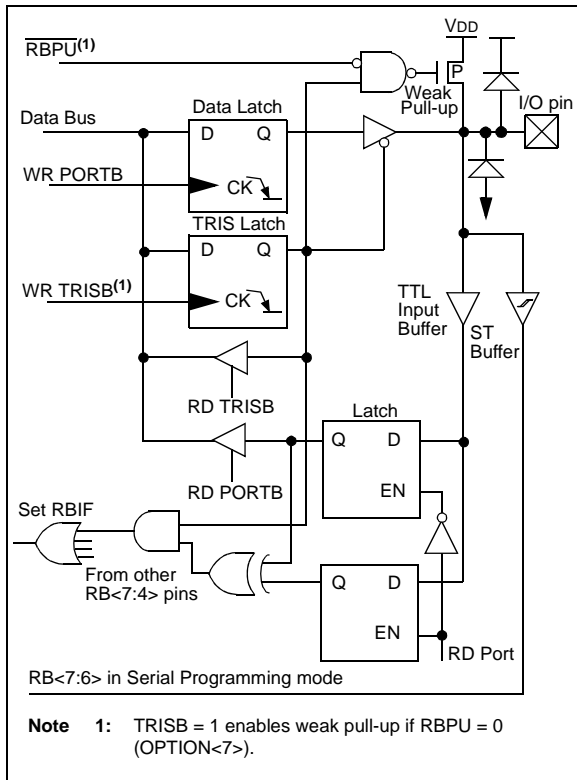
PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a High Impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up ( $\approx 200 \mu\text{A}$  typical). A single control bit can turn on all the pull-ups. This is done by clearing the RBPV (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB<7:4>, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt-on-change comparison). The input pins of RB<7:4> are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

**FIGURE 4-5: BLOCK DIAGRAM OF RB<7:4> PINS**



This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- Clear flag bit RBIF.

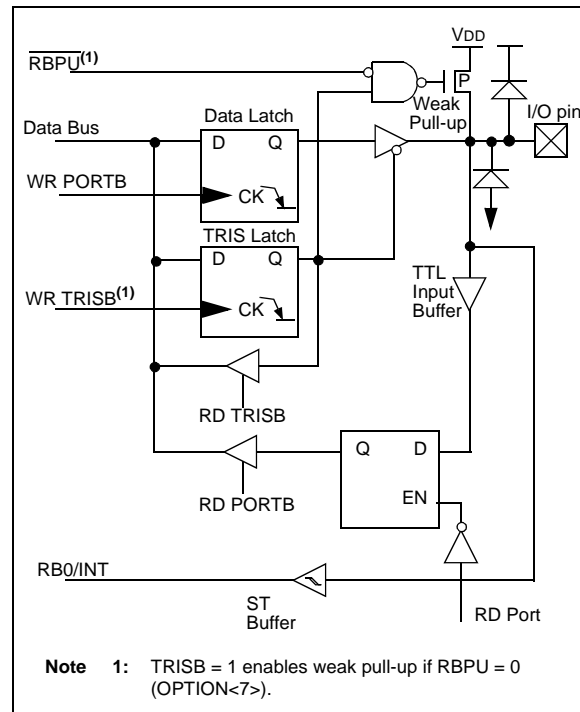
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

This interrupt-on-mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a key pad and make it possible for wake-up on key depression. (See AN552, "Implementing Wake-up on Key Strokes".)

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

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

**FIGURE 4-6: BLOCK DIAGRAM OF RB<3:0> PINS**



**TABLE 4-3: PORTB FUNCTIONS**

Name	Bit #	Buffer Type	Function
RB0/INT	bit0	TTL/ST <sup>(1)</sup>	Input/output or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock pin.
RB7	bit7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data pin.

Legend: ST = Schmitt Trigger, TTL = TTL input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**TABLE 4-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

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
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: u = unchanged, x = unknown

**Note 1:** Shaded bits are not used by PORTB.

# PIC16C432

## 4.3 I/O Programming Considerations

### 4.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of `PORTB` will cause all eight bits of `PORTB` to be read into the CPU. Then the `BSF` operation takes place on bit5 and `PORTB` is written to the output latches. If another bit of `PORTB` is used as a bi-directional I/O pin (i.e., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (i.e., `BCF`, `BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 4-2 shows the effect of two sequential read-modify-write instructions (i.e., `BCF`, `BSF`, etc.) on an I/O port.

A pin actively outputting a Low or High should not be driven from external devices at the same time, in order to change the level on this pin (“wired-or”, “wired-and”). The resulting high output currents may damage the chip.

### EXAMPLE 4-2: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```

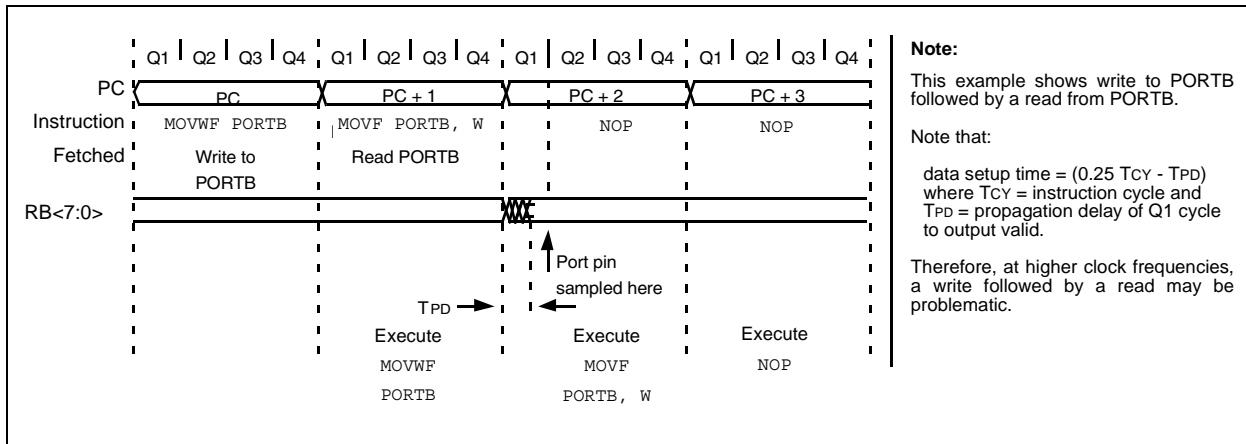
; Initial PORT settings:      PORTB<7:4> Inputs
                              PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are not connected
; to other circuitry
;
;
;                               PORT latch   PORT pins
;                               -----   -----
;
BCF PORTB, 7      ; 01pp pppp   11pp pppp
BCF PORTB, 6      ; 10pp pppp   11pp pppp
BSF STATUS,      ;
RPO
BCF TRISB, 7      ; 10pp pppp   11pp pppp
BCF TRISB, 6      ; 10pp pppp   10pp pppp
;
; NOTE: that the user may have expected the pin values to
; be 00pp pppp. The 2nd BCF caused RB7 to be latched as
; the pin value (High).

```

### 4.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 4-7). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU, rather than the new state. When in doubt, it is better to separate these instructions with a `NOP`, or another instruction not accessing this I/O port.

FIGURE 4-7: SUCCESSIVE I/O OPERATION



## 5.0 LIN TRANSCEIVER

The PIC16C432 has an integrated LIN transceiver which allows the microcontroller to communicate via LIN. The LIN protocol is handled by the microcontroller. The conversion from 5V signal to LIN signals is handled by the transceiver.

### 5.1 The LIN Protocol

The LIN protocol is not described within this document. For further information regarding the LIN protocol, please refer to [www.lin-subbus.org](http://www.lin-subbus.org).

### 5.2 LIN Interfacing

The LIN protocol is implemented and programmed by the user, using the LINTX and LINRX bits, which are used to interface to the transceiver. The LIN firmware transmits by toggling the LINTX bit in the LININTF register and is read by reading the LINRX bit in the PORTA register. All aspects of the protocol are handled by software (i.e., bit-banged), where the transceiver is used as the physical interface to the LIN network.

For LIN software implementation, please refer to Microchip's website ([www.microchip.com](http://www.microchip.com)).

**Note:** The LINTX is bit 2 of the LININTF register.

If the LINTX bit is left cleared, no other nodes on the network will be able to communicate on the LIN for this is the dominate state for the protocol. The transceiver can be powered down by clearing the LINVDD bit in the LININTF register. This can be useful to reduce current consumption but does not allow the microcontroller to wake-up on LIN activity because the transceiver will be disabled. It is recommended that the firmware verify each bit transmitted, by comparing the LINTX and LINRX bits, to ensure no bus contention or hardware failure has occurred. The LINTX bit has no associated TRIS bit and is always an output. The LINRX bit has an associated TRIS bit, TLINRX, in the TRISA register.

**Note:** TLINRX, bit 1 of TRISA register, must be set to '1' at all times.

### 5.3 LIN Hardware Interface

Figure 6-1 shows how to implement a hardware LIN interface in a master configuration and Figure 6-2 in a slave configuration using the PIC16C432. Figure 6-3 shows how to implement the hardware for a master configuration using BACT pin to generate a wake-up interrupt using RB0. The transceiver has an internal series resistor and diode, as defined in the LIN 1.2 specification, connecting VBAT and LIN.

**Note:** No resistor is required between VBAT pin and 12V supply and for slave configuration, no resistor is required between VBAT and LIN.

### 5.4 Thermal Shutdown

In thermal shutdown, the LIN output is disabled instantaneously. The output transistor is turned off, regardless of the input level at pin LINTX bit and only a limited current can flow into the receiver connected to the LIN pin.

### 5.5 Wake-up From SLEEP Upon Bus Activity

The PIC16C432 can wake-up from SLEEP upon bus activity in two ways:

1. With the use of the comparators.
2. Connecting BACT to one of PORTB<0,4:7> pins.

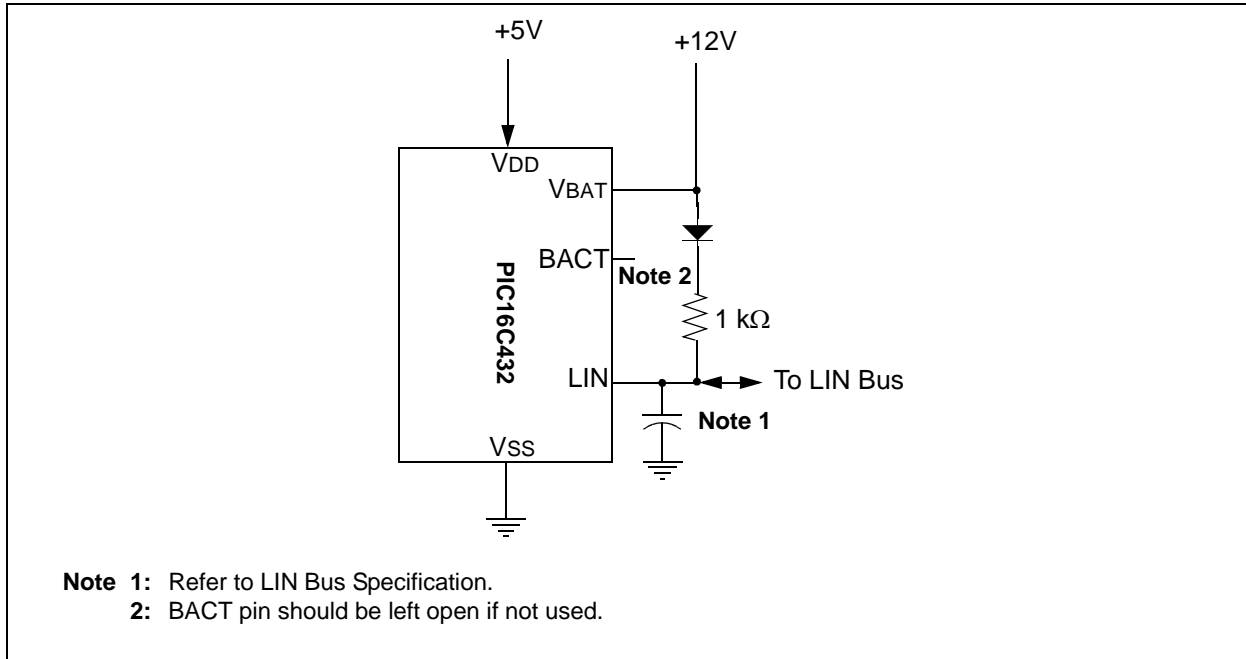
In case the comparators are used to wake-up the device upon bus activity, a reference to the LIN signal has to be supplied. This is usually  $V_{DD}/2$ . The reference can either be an external reference or the internal voltage reference. Once the device is in SLEEP mode, the comparator interrupt will wake-up the device. On RESET, LINRX is configured as an analog comparator input (Section 8.1 of Data Sheet) which can be used to generate an interrupt to wake-up the device from SLEEP on bus activity. The LINRX bit will not receive data from the bus configured as an analog input, therefore, after wake-up from comparator interrupt or RESET, LINRX must be configured as a digital input to read the bus.

The BACT output is a CMOS-levels representation of the LIN pin. This signal can be routed to one of the PORTB<0,4:7> pins. The RB0/INT external interrupt or PORTB<4:7> interrupt-on-change wakes up the device from SLEEP. Any one of the five PORTB pins can be used for wake-up where PORTB<0> offers multiple configuration options (Section 10.5.1 of Data Sheet) and PORTB<4:7> are interrupt-on-change (Section 10.5.3 of Data Sheet).

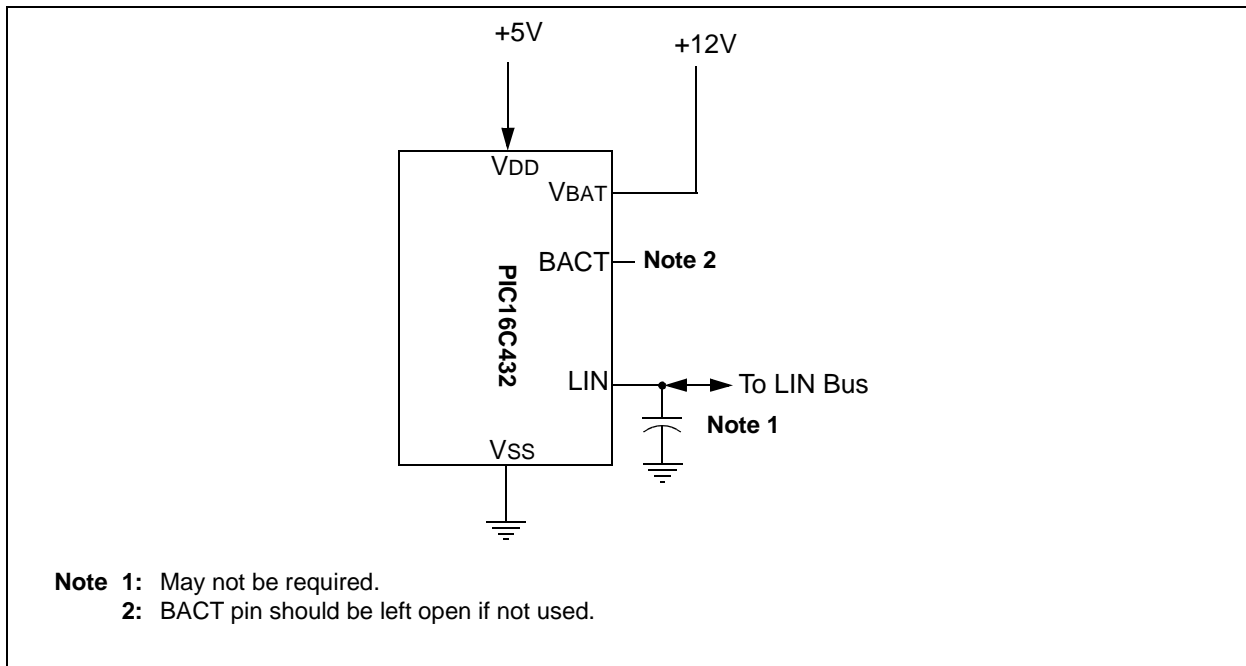
**Note:** BACT pin is an output and must be left open if unused.

# PIC16C432

**FIGURE 5-1: TYPICAL LIN BUS MASTER APPLICATION**

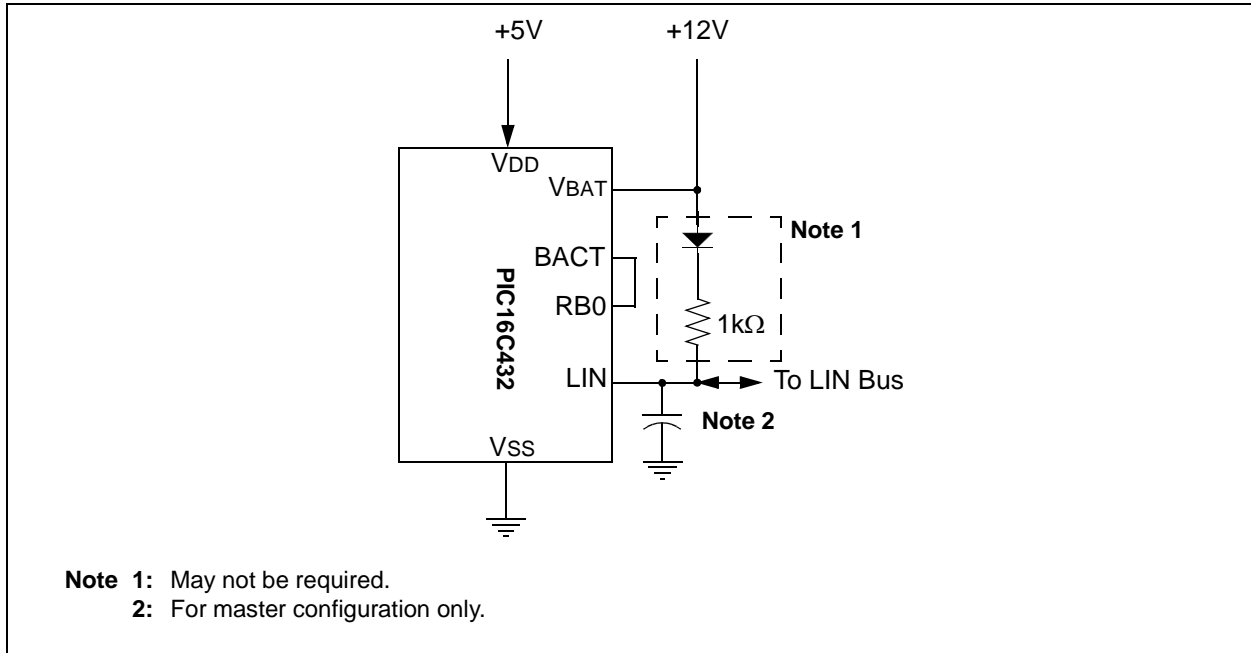


**FIGURE 5-2: TYPICAL LIN BUS SLAVE APPLICATION**





**FIGURE 5-3: LIN BUS APPLICATION USING WAKE-UP INTERRUPT**



# PIC16C432

## REGISTER 5-1: LININTF REGISTER (ADDRESS: 90h)

U-0	U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-1
—	—	—	—	—	LINTX	—	LINVDD
bit 7				bit 0			

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

bit 2 **LINTX:** LIN Bus Transmit bit

1 = LIN Bus line is high

0 = LIN Bus line is low

bit 1 **Unimplemented:** Read as '0'

bit 0 **LINVDD:** LIN Bus Transceiver VDD Supply bit

1 = VDD is supplied to the LIN Bus transceiver via microcontroller

0 = VDD is not supplied to the LIN Bus transceiver

**Note 1:** Transceiver VDD is same as microcontroller VDD.

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

**TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH LIN TRANSCEIVER**

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
05h	PORTA	—	—	—	RA4	RA3	RA2	LINRX	RA0	---x 0000	---u 0000
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(2)</sup>	TRISA0	---1 1111	---1 1111
90h	LININTF	—	—	—	—	—	LINTX	—	LINVDD	---- -1-1	---- -1-1

Legend: x = unknown, u = unchanged, — = Unimplemented locations read as '0'.

**Note 1:** Shaded bits are not used by LIN transceiver

**Note 2:** TLINRX must be set to '1' at all times.

## 6.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the TMR0 will increment every instruction cycle (without prescaler). If Timer0 is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to TMR0.

Counter mode is selected by setting the T0CS bit. In this mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge (T0SE) control

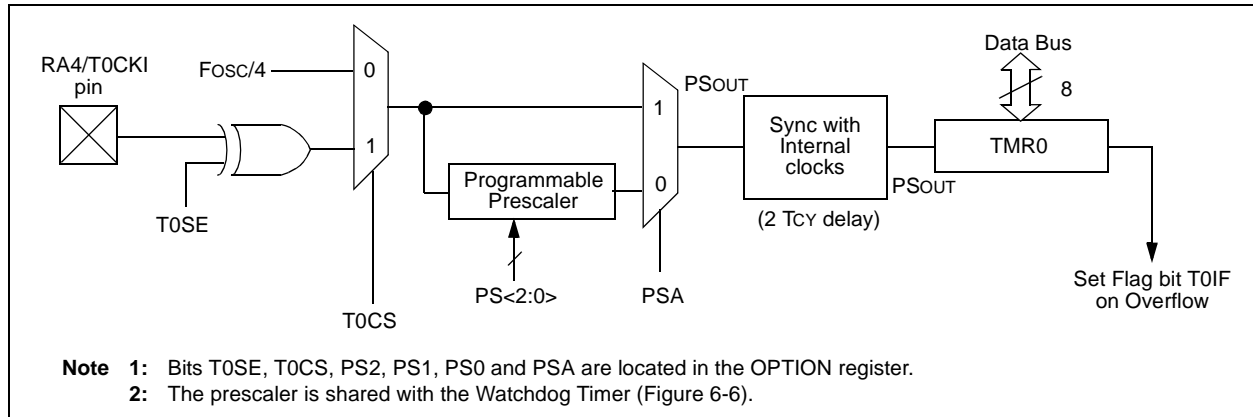
bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

The prescaler is shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4,..., 1:256 are selectable. Section 6.3 details the operation of the prescaler.

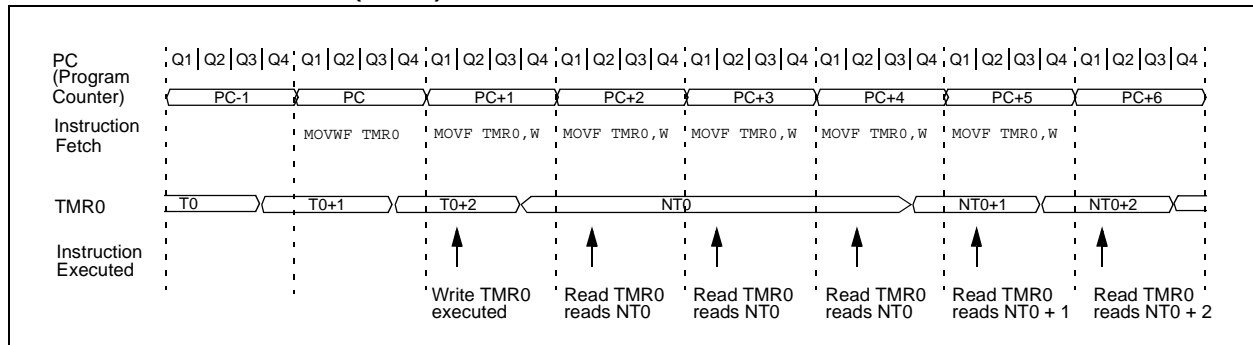
### 6.1 Timer0 Interrupt

Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the Timer0 module Interrupt Service Routine, before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from SLEEP, since the timer is shut-off during SLEEP. See Figure 6-4 for Timer0 interrupt timing.

**FIGURE 6-1: TIMER0 BLOCK DIAGRAM**

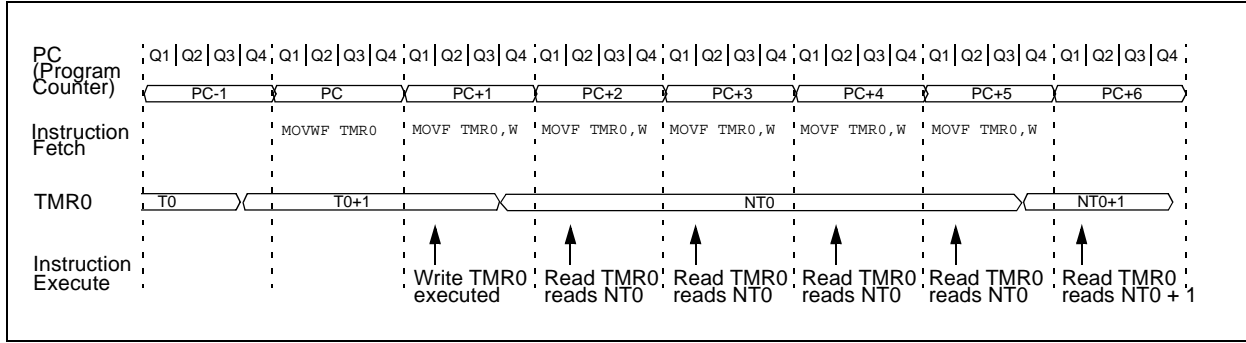


**FIGURE 6-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALER**

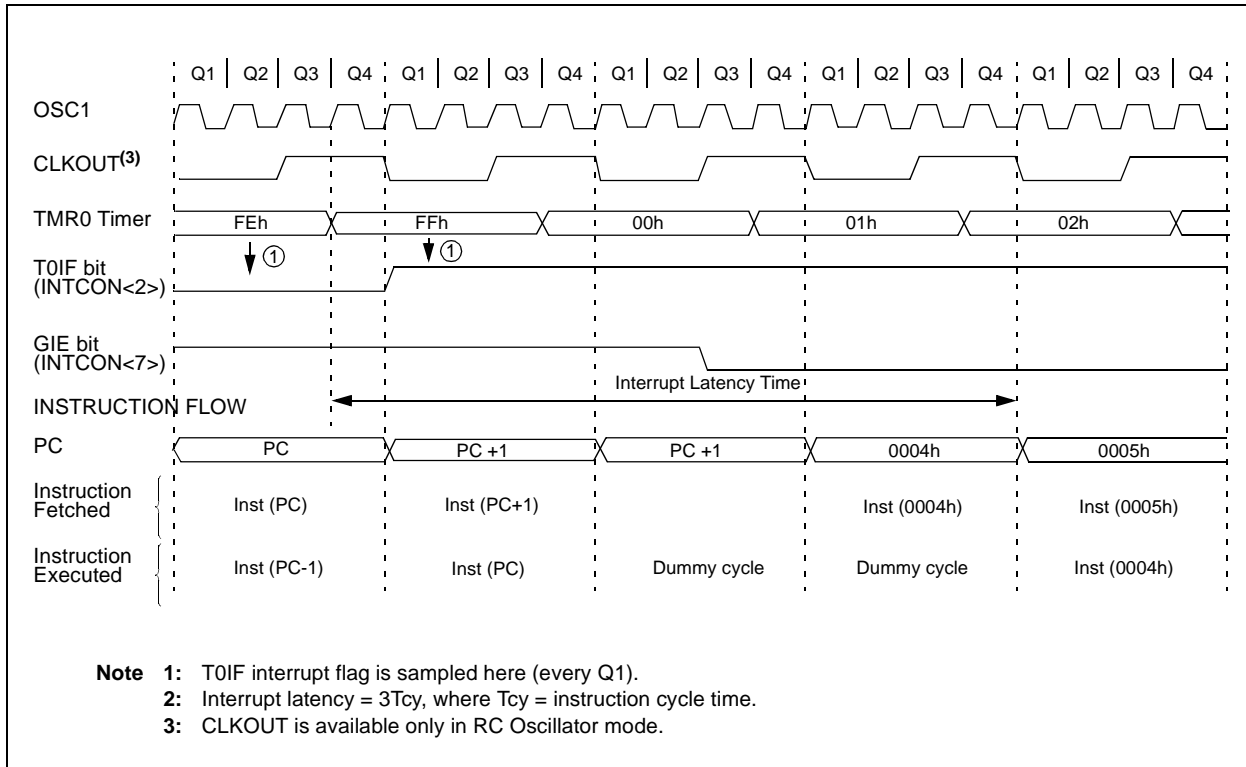


# PIC16C432

**FIGURE 6-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2**



**FIGURE 6-4: TIMER0 INTERRUPT TIMING**



## 6.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

### 6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

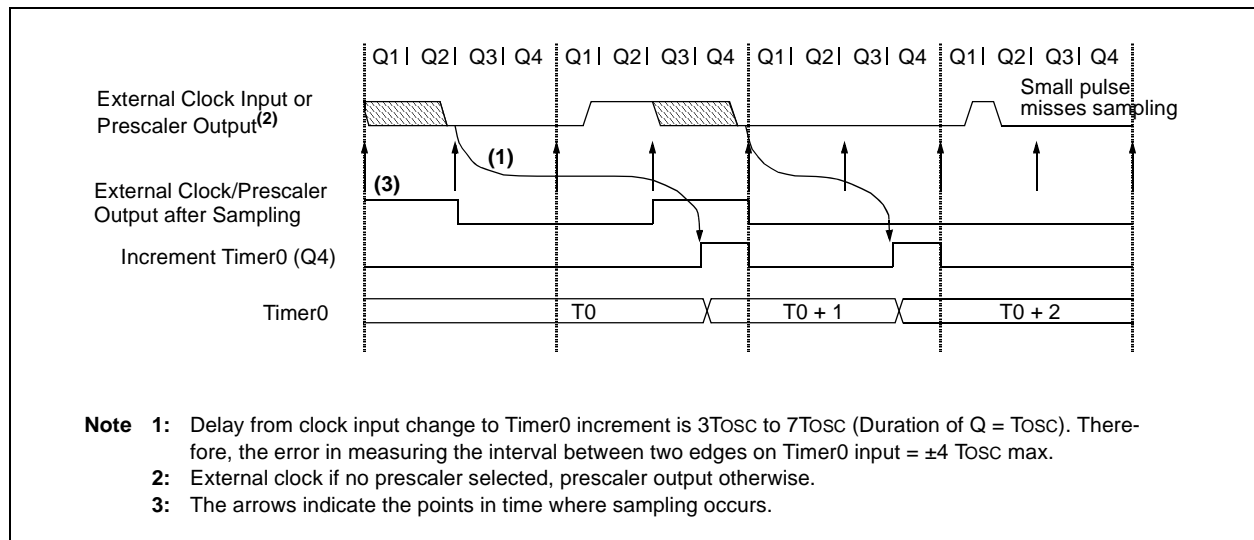
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4TOSC (and a small RC delay of 40 ns), divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

### 6.2.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

**FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK**



# PIC16C432

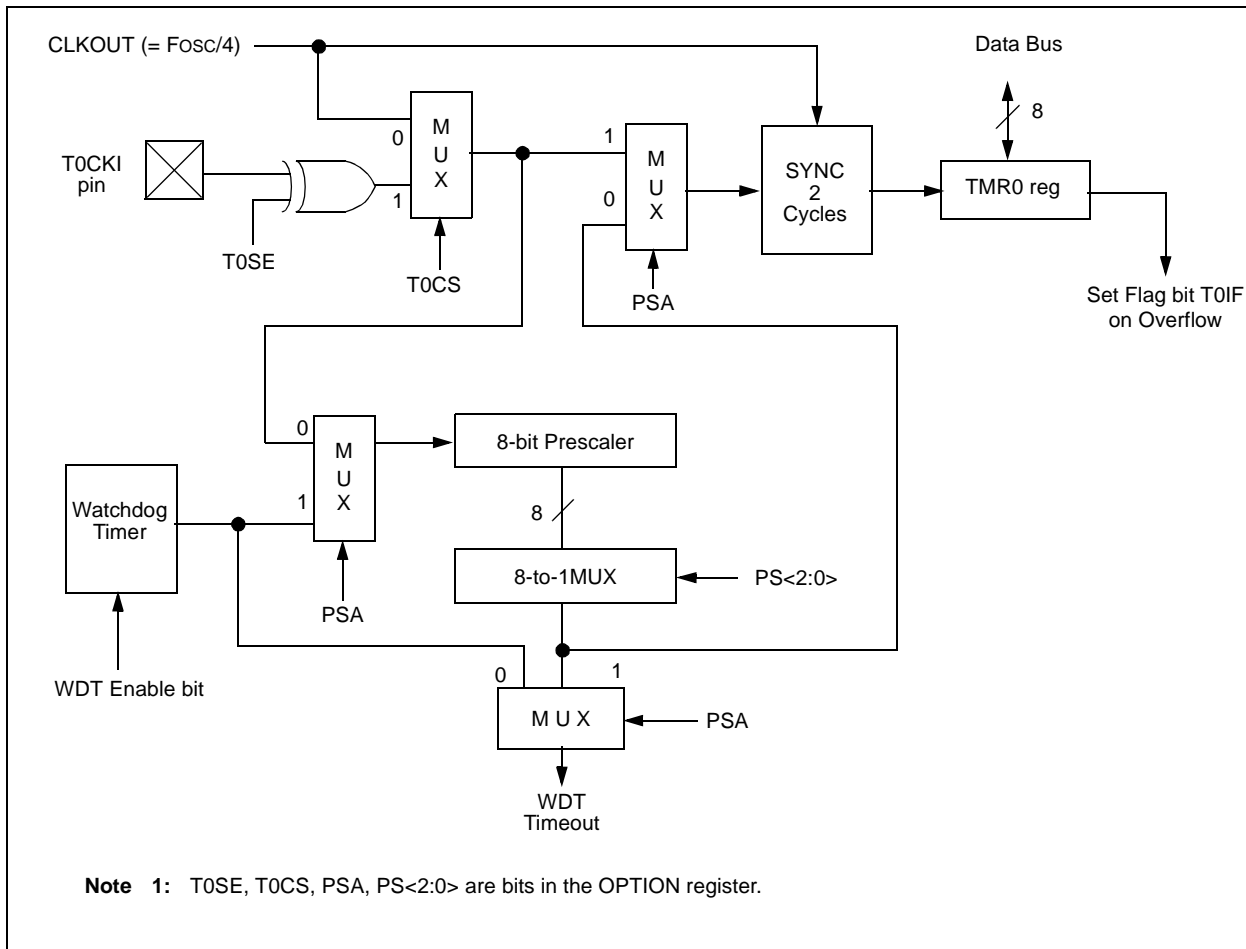
## 6.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 6-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available, which is mutually exclusive between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (i.e., CLRF 1, MOVWF 1, BSF 1,x,...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

**FIGURE 6-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



## 6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to WDT.

### EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```

1.BCF     STATUS, RP0    ;Skip if already in
                        ; Bank 0
2.CLRWDT                    ;Clear WDT
3.CLRF   TMR0            ;Clear TMR0 & Prescaler
4.BSF   STATUS, RP0    ;Bank 1
5.MOVLW '00101111'b    ;These 3 lines (5, 6, 7)
6.MOVWF OPTION            ; are required only
                        ; if desired PS<2:0>
                        ; are
7.CLRWDT                    ; 000 or 001
8.MOVLW '00101xxx'b    ;Set Postscaler to
9.MOVWF OPTION            ; desired WDT rate
10.BCF   STATUS, RP0    ;Return to Bank 0
    
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 6-2. This precaution must be taken, even if the WDT is disabled.

### EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```

CLRWDT                    ;Clear WDT and
                        ;prescaler
BSF   STATUS, RP0
MOVLW b'xxxx0xxx'      ;Select TMR0, new
                        ;prescale value and
                        ;clock source
MOVWF OPTION_REG
BCF   STATUS, RP0
    
```

**TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0**

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
01h	TMR0	Timer0 module register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	RBP <sub>U</sub>	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(2)</sup>	TRISA0	---1 1111	---1 1111

Legend: — = Unimplemented locations, read as '0', x = unknown, u = unchanged

**Note 1:** Shaded bits are not used by TMR0 module.

**Note 2:** TLINRX must be set to '1' at all times.

# PIC16C432

---

---

NOTES:



## 7.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the RA0 through RA3 pins. The on-chip voltage reference (Section 8.0) can also be an input to the comparators.

The CMCON register, shown in Register 7-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 7-1.

**REGISTER 7-1: CMCON REGISTER (ADDRESS 1Fh)**

	R-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
	bit7							bit0

- bit 7     **C2OUT:** Comparator 2 Output bit  
           1 = C2 VIN+ > C2 VIN-  
           0 = C2 VIN+ < C2 VIN-
- bit 6     **C1OUT:** Comparator 1 Output bit  
           1 = C1 VIN+ > C1 VIN-  
           0 = C1 VIN+ < C1 VIN-
- bit 5-4   **Unimplemented:** Read as '0'
- bit 3     **CIS:** Comparator Input Switch bit  
           When CM<2:0> = 001:  
           1 = C1 VIN- connects to RA3  
           0 = C1 VIN- connects to RA0  
           When CM<2:0> = 010:  
           1 = C1 VIN- connects to RA3  
               C2 VIN- connects to RA2  
           0 = C1 VIN- connects to RA0  
               C2 VIN- connects to LINRX
- bit 2-0   **CM<2:0>:** Comparator Mode bits  
           (See Figure 7-1)

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

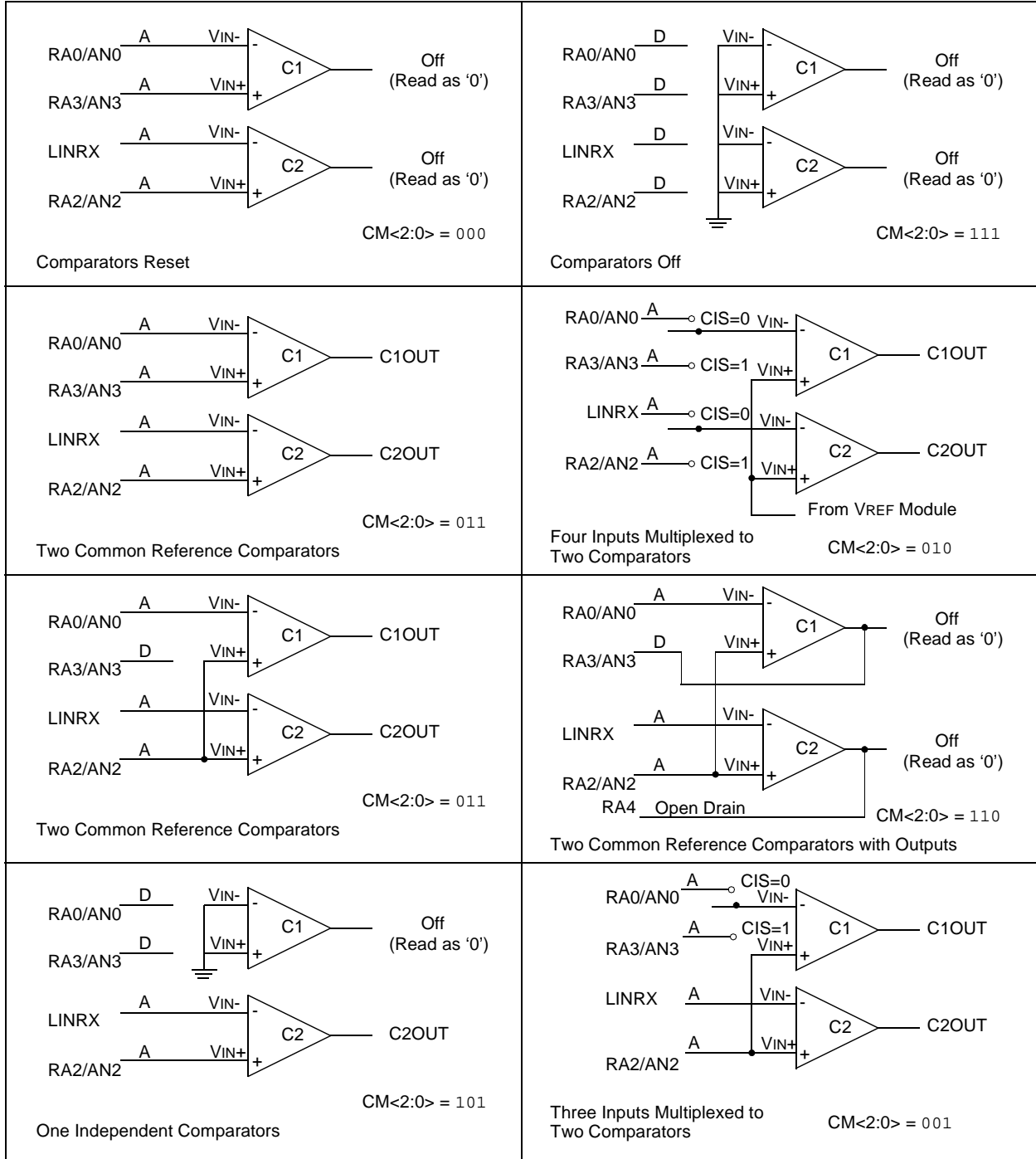
### 7.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. Figure 7-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 12-1.

**Note:** Comparator interrupts should be disabled during a Comparator mode change, otherwise a false interrupt may occur.

# PIC16C432

**FIGURE 7-1: COMPARATOR I/O OPERATING MODES**



Legend: A = Analog Input, Port Reads '0' Always    D = Digital Input    CIS = CMCON<3>, Comparator Input Switch

The code example in Example 7-1 depicts the steps required to configure the comparator module. RA3 and RA4 are configured as digital output. RA0 and RA1 are configured as the V- inputs and RA2 as the V+ input to both comparators.

## EXAMPLE 7-1: INITIALIZING COMPARATOR MODULE

```
FLAG_REG EQU      0X20
CLRF  FLAG_REG    ;Init flag register
CLRF  PORTA       ;Init PORTA
MOVF  CMCON,W     ;Move comparator contents to W
ANDLW 0xC0        ;Mask comparator bits
IORWF FLAG_REG,F  ;Store bits in flag register
MOVLW 0x03        ;Init comparator mode
MOVWF CMCON       ;CM<2:0> = 011
BSF   STATUS,RP0  ;Select Bank1
MOVLW 0x07        ;Initialize data direction
MOVWF TRISA       ;Set RA<2:0> as inputs
                          ;RA<4:3> as outputs
                          ;TRISA<7:5> always read '0'

BCF   STATUS,RP0  ;Select Bank 0
CALL  DELAY 10    ;10ms delay
MOVF  CMCON,F     ;Read CMCON to end change condition
BCF   PIR1,CMIF   ;Clear pending interrupts
BSF   STATUS,RP0  ;Select Bank 1
BSF   PIE1,CMIE   ;Enable comparator interrupts
BCF   STATUS,RP0  ;Select Bank 0
BSF   INTCON,PEIE ;Enable peripheral interrupts
BSF   INTCON,GIE  ;Global interrupt enable
```

## 7.2 Comparator Operation

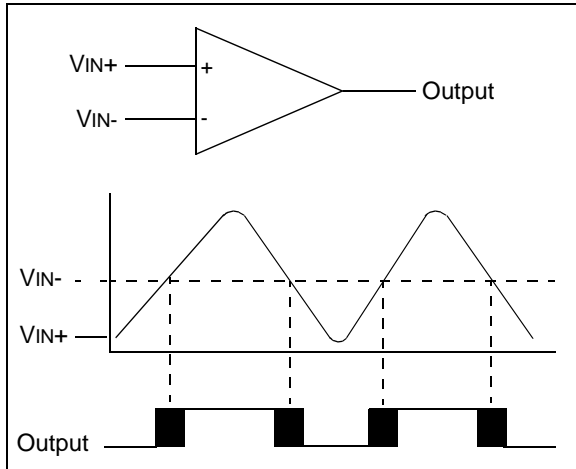
A single comparator is shown in Figure 7-2, along with the relationship between the analog input levels and the digital output. When the analog input at  $V_{IN+}$  is less than the analog input  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog input at  $V_{IN+}$  is greater than the analog input  $V_{IN-}$ , the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 7-2 represent the uncertainty due to input offsets and response time.

## 7.3 Comparator Reference

An external or internal reference signal may be used, depending on the Comparator Operating mode. The analog signal that is present at  $V_{IN-}$  is compared to the signal at  $V_{IN+}$ , and the digital output of the comparator is adjusted accordingly (Figure 7-2).

# PIC16C432

**FIGURE 7-2: SINGLE COMPARATOR**



## 7.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same, or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between  $V_{SS}$  and  $V_{DD}$  and can be applied to either pin of the comparator(s).

## 7.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. Section 8.0, Voltage Reference Module, contains a detailed description of the Voltage Reference Module that provides this signal. The internal reference signal is used when the comparators are in mode  $CM\langle 2:0 \rangle = 010$  (Figure 7-1). In this mode, the internal voltage reference is applied to the  $V_{IN+}$  pin of both comparators.

## 7.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs, otherwise the maximum delay of the comparators should be used (Table 12.1).

## 7.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When the CM<2:0> = 110, multiplexors in the output path of the RA3 and RA4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 7-3 shows the comparator output block diagram.

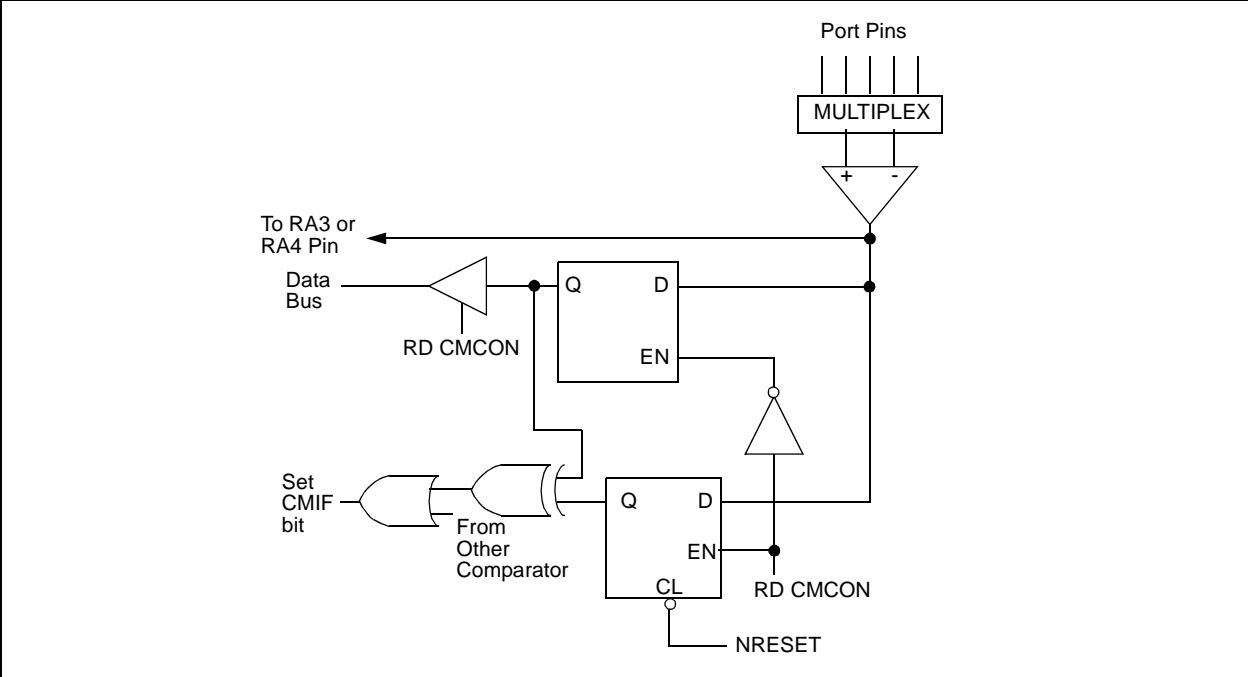
The TRISA bits will still function as an output enable/disable for the RA3 and RA4 pins while in this mode.

**Note 1:** When reading the PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.

**2:** Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

# PIC16C432

FIGURE 7-3: COMPARATOR OUTPUT BLOCK DIAGRAM



## 7.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit, PIR1<6>, is the comparator interrupt flag. The CMIF bit must be reset by clearing '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE1<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

**Note:** If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<6>) interrupt flag may not get set.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- a) Any read or write of CMCON. This will end the mismatch condition.
- b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition and allow flag bit CMIF to be cleared.

## 7.7 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from SLEEP mode when enabled. While the comparator is powered up, higher SLEEP currents than shown in the power-down current specification will occur. Each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM<2:0> = 111, before entering SLEEP. If the device wakes up from SLEEP, the contents of the CMCON register are not affected.

## 7.8 Effects of a RESET

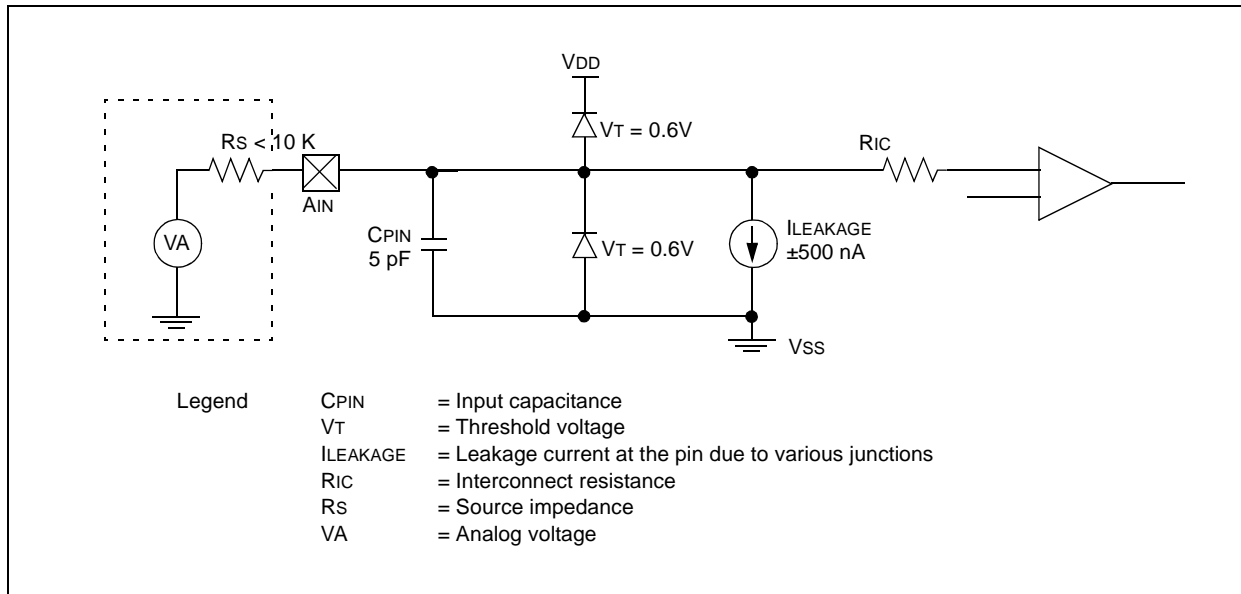
A device RESET forces the CMCON register to its RESET state. This forces the comparator module to be in the Comparator RESET mode, CM<2:0> = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at RESET time. The comparators will be powered down during the RESET interval.

## 7.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 7-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latchup may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

# PIC16C432

**FIGURE 7-4: ANALOG INPUT MODEL**



**TABLE 7-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

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
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	-0-- ----	-0-- ----
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	-0-- ----	-0-- ----
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(1)</sup>	TRISA0	---1 1111	---1 1111

Legend: — = Unimplemented, read as '0', x = unknown, u = unchanged

**Note 1:** TLINRX must be set to '1' at all times.



## 8.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference is not being used. The VRCON register controls the operation of the reference as shown in Register 8-1. The block diagram is given in Figure 8-1.

## 8.1 Configuring the Voltage Reference

The Voltage Reference can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

$$\text{if } VRR = 1: VREF = (VR<3:0>/24) \times VDD$$

$$\text{if } VRR = 0: VREF = (VDD \times 1/4) + (VR<3:0>/32) \times VDD$$

The setting time of the Voltage Reference must be considered when changing the VREF output (Table 12.1). Example 8-1 shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with VDD = 5.0V.

**REGISTER 8-1: VRCON REGISTER (ADDRESS 9Fh)**

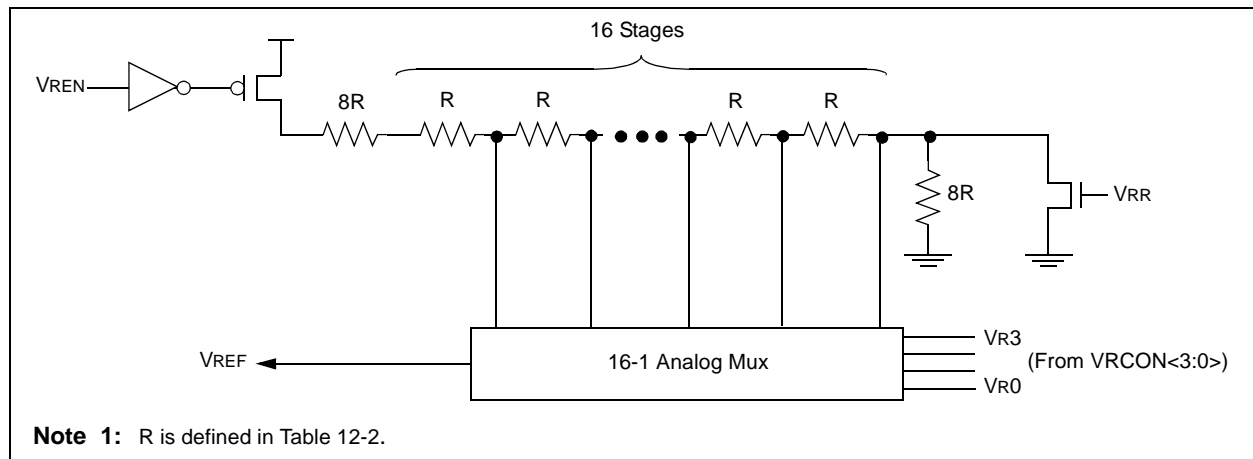
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	
bit7								bit0

- bit 7 **VREN:** VREF Enable bit  
1 = VREF circuit powered on  
0 = VREF circuit powered down, no IDD drain
- bit 6 **VROE:** VREF Output Enable bit  
1 = VREF is output on RA2 pin  
0 = VREF is disconnected from RA2 pin
- bit 5 **VRR:** VREF Range Selection bit  
1 = Low Range  
0 = High Range
- bit 4 **Unimplemented:** Read as '0'
- bit 3-0 **VR<3:0>:** VREF Value Selection  $0 \leq VR [3:0] \leq 15$   
when VRR = 1:  $VREF = (VR<3:0>/24) \times VDD$   
when VRR = 0:  $VREF = 1/4 \times VDD + (VR<3:0>/32) \times VDD$

Legend:

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

**FIGURE 8-1: VOLTAGE REFERENCE BLOCK DIAGRAM**



# PIC16C432

## EXAMPLE 8-1: VOLTAGE REFERENCE CONFIGURATION

```

MOVLW    0x02           ; 4 Inputs Muxed
MOVWF    CMCON          ; to 2 comps.
BSF      STATUS,RP0    ; go to Bank 1
MOVLW    0x07           ; RA3-RA0 are
MOVWF    TRISA          ; outputs
MOVLW    0xA6           ; enable VREF
MOVWF    VRCON          ; low range
                        ; set VR<3:0>=6
BCF      STATUS,RP0    ; go to Bank 0
CALL     DELAY10        ; 10µs delay
    
```

## 8.2 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-1) keep VREF from approaching VSS or VDD. The Voltage Reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The absolute accuracy of the Voltage Reference can be found in Table 12-2.

## 8.3 Operation During SLEEP

When the device wakes up from SLEEP through an interrupt or a Watchdog Timer timeout, the contents of the VRCON register are not affected. To minimize current consumption in SLEEP mode, the Voltage Reference should be disabled.

## 8.4 Effects of a RESET

A device RESET disables the Voltage Reference by clearing bit VREN (VRCON<7>). This RESET also disconnects the reference from the RA2 pin by clearing bit VROE (VRCON<6>) and selects the high voltage range by clearing bit VRR (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

## 8.5 Connection Considerations

The Voltage Reference Module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit is set and the VROE bit, VRCON<6>, is set. Enabling the Voltage Reference output onto the RA2 pin, with an input signal present, will increase current consumption. Connecting RA2 as a digital output with VREF enabled will also increase current consumption.

The RA2 pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the Voltage Reference output for external connections to VREF. Figure 8-2 shows an example buffering technique.

FIGURE 8-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

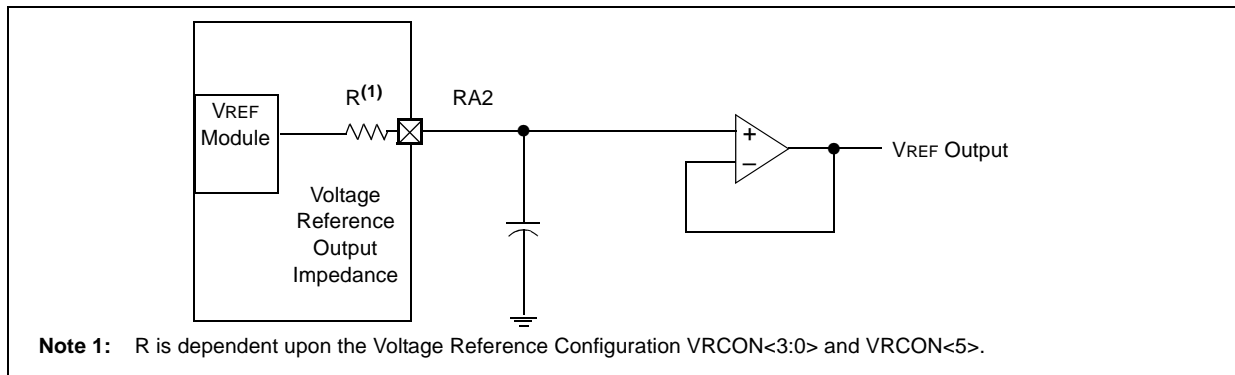


TABLE 8-2: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value On POR/BOD	Value On All Other RESETS
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TLINRX <sup>(1)</sup>	TRISA0	---1 1111	---1 1111

Legend: — = Unimplemented, read as '0'

**Note 1:** TLINRX must be set to '1' at all times.

## 9.0 SPECIAL FEATURES OF THE CPU

Special circuits to deal with the needs of real-time applications are what sets a microcontroller apart from other processors. The PIC16C432 device has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection.

These are:

1. OSC Selection
2. RESET
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-Up Timer (OST)
  - Brown-out Reset (BOD)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code Protection
7. ID Locations
8. In-circuit Serial Programming

The PIC16C432 has a Watchdog Timer which is controlled by configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, and is designed to keep the part in RESET while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which provides at least a 72 ms RESET. With these three functions on-chip, most applications need no external RESET circuitry.

The SLEEP mode is designed to offer a very low current Power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options.

### 9.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h – 3FFFh), which can be accessed only during programming.

# PIC16C432

## REGISTER 9-1: CONFIGURATION WORD

CP1	CP0 <sup>(2)</sup>	CP1	CP0 <sup>(2)</sup>	CP1	CP0 <sup>(2)</sup>	—	BODEN <sup>(1)</sup>	CP1	CP0 <sup>(2)</sup>	$\overline{\text{PWRTE}}^{\text{(1)}}$	WDTE	F0SC1	F0SC0
bit 13											bit 0		

bit 13-8 **CP1:CP0 Pairs:** Code protection bit pairs<sup>(2)</sup>

bit 5-4 **Code protection for 2K program memory bits**

11 = Program memory code protection off  
 10 = 0400h-07FFh code protected  
 01 = 0200h-07FFh code protected  
 00 = 0000h-07FFh code protected

bit 7 **Unimplemented:** Read as '1'

bit 6 **BODEN:** Brown-out Reset Enable bit <sup>(1)</sup>

1 = BOD enabled  
 0 = BOD disabled

bit 3  **$\overline{\text{PWRTE}}$ :** Power-up Timer Enable bit <sup>(1)</sup>

1 = PWRT disabled  
 0 = PWRT enabled

bit 2 **WDTE:** Watchdog Timer Enable bit

1 = WDT enabled  
 0 = WDT disabled

bit 1-0 **F0SC1:F0SC0:** Oscillator Selection bits

11 = RC oscillator  
 10 = HS oscillator  
 01 = XT oscillator  
 00 = LP oscillator

**Note 1:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit  $\overline{\text{PWRTE}}$ . Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

**2:** All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.

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

## 9.2 Oscillator Configurations

### 9.2.1 OSCILLATOR TYPES

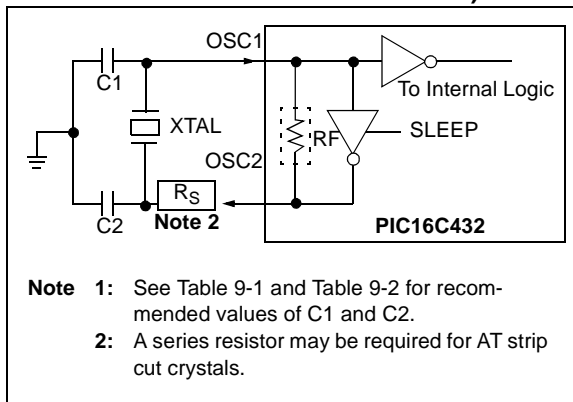
The PIC16C432 can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP - Low Power Crystal
- XT - Crystal/Resonator
- HS - High Speed Crystal/Resonator
- RC - Resistor/Capacitor

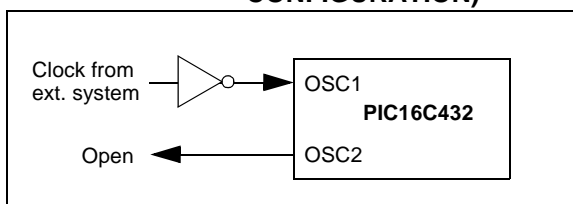
### 9.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 9-1). The PIC16C432 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 9-2).

**FIGURE 9-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 9-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**TABLE 9-1: CERAMIC RESONATORS, PIC16C432**

Ranges Tested:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF

**These values are for design guidance only. See notes at bottom of page.**

**TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR, PIC16C432**

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

**These values are for design guidance only. See notes at bottom of page.**

- Note 1:** Recommended values of C1 and C2 are identical to the ranges tested table.
- 2:** Higher capacitance increases the stability of oscillator, but also increases the start-up time.
- 3:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
- 4:** Rs may be required in HS mode, as well as XT mode, to avoid over driving crystals with low drive level specification.

# PIC16C432

## 9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used, or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 9-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 9-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

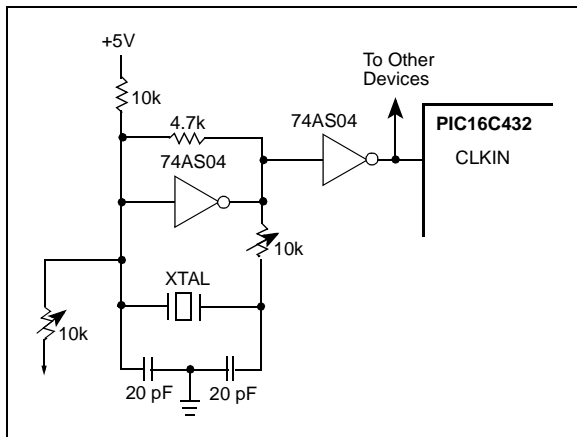
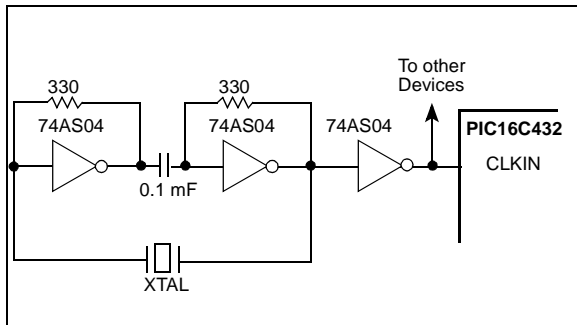


Figure 9-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 9-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**



## 9.2.4 RC OSCILLATOR

For timing insensitive applications, the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{EXT}$ ) and capacitor ( $C_{EXT}$ ) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low  $C_{EXT}$  values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-5 shows how the R/C combination is connected to the PIC16C432. For  $R_{EXT}$  values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high  $R_{EXT}$  values (i.e., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, it is recommended to keep  $R_{EXT}$  between 3 kΩ and 100 kΩ.

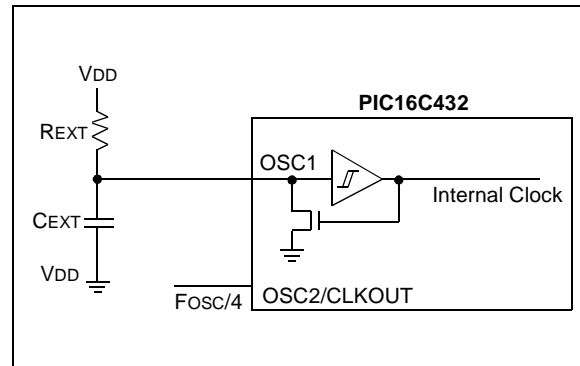
Although the oscillator will operate with no external capacitor ( $C_{EXT} = 0$  pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance, or package lead frame capacitance.

The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 2.0 for variation of oscillator frequency due to  $V_{DD}$  for given  $R_{EXT}/C_{EXT}$  values, as well as frequency variation due to operating temperature for given R, C, and  $V_{DD}$  values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin and can be used for test purposes, or to synchronize other logic (see Figure 4-2 for waveform).

**FIGURE 9-5: RC OSCILLATOR MODE**



## 9.3 RESET

The PIC16C432 differentiates between various kinds of RESET:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during SLEEP
- WDT Reset (normal operation)
- WDT wake-up (SLEEP)
- Brown-out Reset (BOD)

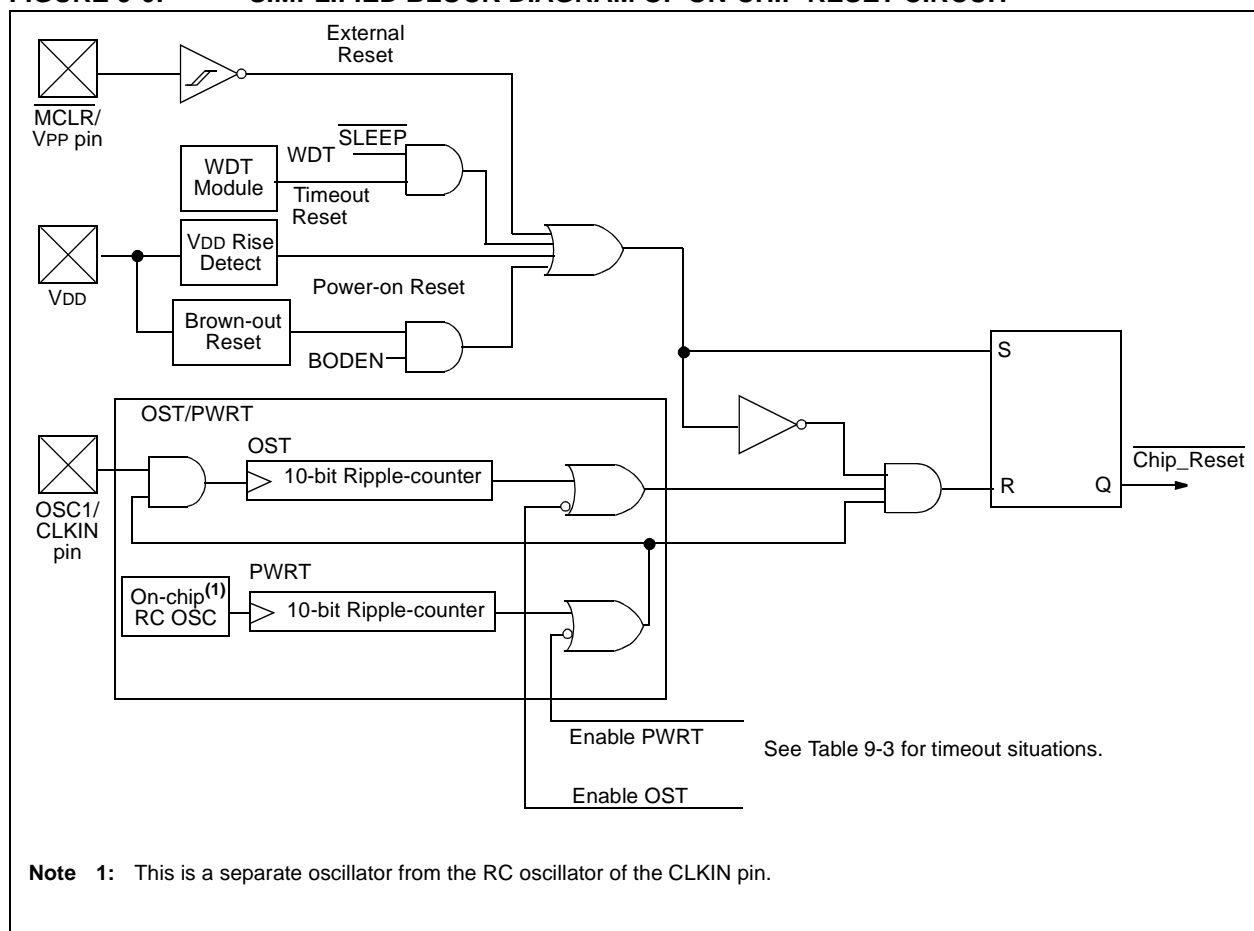
Some registers are not affected in any RESET condition. Their status is unknown on POR and unchanged in any other RESET. Most other registers are RESET to a "RESET state" on Power-on Reset, MCLR Reset,

WDT Reset and  $\overline{\text{MCLR}}$  Reset during SLEEP. They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. TO and PD bits are set or cleared differently in different RESET situations, as indicated in Table 9-4. These bits are used in software to determine the nature of the RESET. See Table 9-6 for a full description of RESET states of all registers.

A simplified block diagram of the On-chip Reset Circuit is shown in Figure 9-6.

The  $\overline{\text{MCLR}}$  Reset path has a noise filter to detect and ignore small pulses. See Table 12-6 for pulse width specification.

**FIGURE 9-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC16C432

## 9.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOD)

### 9.4.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in RESET until VDD has reached a high enough level for proper operation. To take advantage of the POR, just tie the MCLR pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See 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 operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met.

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

### 9.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) timeout on power-up only, from POR or Brown-out Reset. The Power-up Timer 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, PWRTE, can disable (if set), or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always 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 for details.

### 9.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST timeout is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

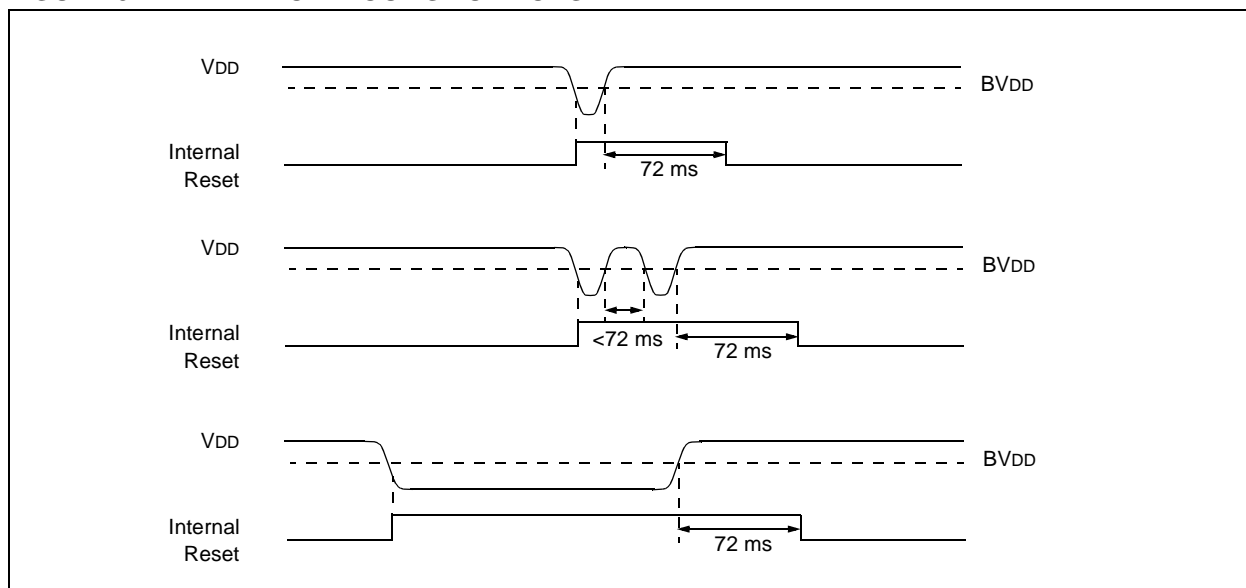
### 9.4.4 BROWN-OUT RESET (BOD)

The PIC16C432 has an on-chip Brown-out Reset circuitry. A configuration bit, BOREN, can disable (if clear/programmed), or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (refer to BVDD parameter D005) for greater than parameter (TBOR) in Table 12-6, the brown-out situation will reset the chip. A RESET won't occur if VDD falls below 4.0V for less than parameter (TBOR).

On any RESET (Power-on, Brown-out, Watchdog, etc.), the chip will remain in RESET until VDD rises above BVDD. The Power-up Timer will then be invoked and will keep the chip in RESET an additional 72 ms.

If VDD drops below BVDD 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 BVDD, the Power-up Timer will execute a 72 ms RESET. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 9-7 shows typical Brown-out situations.

FIGURE 9-7: BROWN-OUT SITUATIONS





## 9.4.5 TIMEOUT SEQUENCE

On power-up, the timeout sequence is as follows: First PWRT timeout is invoked after POR has expired, then OST is activated. The total timeout will vary based on oscillator configuration and  $\overline{\text{PWRTE}}$  bit status. For example, in RC mode with  $\overline{\text{PWRTE}}$  bit erased ( $\overline{\text{PWRT}}$  disabled), there will be no timeout at all. Figure 9-8, Figure 9-8 and Figure 9-9 depict timeout sequences.

Since the timeouts occur from the POR pulse, if  $\overline{\text{MCLR}}$  is kept low long enough, the timeouts will expire. Then bringing  $\overline{\text{MCLR}}$  high will begin execution immediately (see Figure 9-8). This is useful for testing purposes or to synchronize more than one PIC® device operating in parallel.

Table 9-5 shows the RESET conditions for some special registers, while Table 9-6 shows the RESET conditions for all the registers.

## 9.4.6 POWER CONTROL (PCON)/STATUS REGISTER

The power control/status register, PCON (address 8Eh), has two bits.

Bit0 is  $\overline{\text{BOR}}$  (Brown-out).  $\overline{\text{BOR}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if  $\overline{\text{BOR}} = 0$ , indicating that 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 setting BODEN bit = 0 in the Configuration word).

Bit1 is  $\overline{\text{POR}}$  (Power-on Reset). It is a ‘0’ on Power-on Reset and unaffected otherwise. The user must write a ‘1’ to this bit following a Power-on Reset. On a subsequent RESET, if  $\overline{\text{POR}}$  is ‘0’, it will indicate that a Power-on Reset must have occurred ( $\text{VDD}$  may have gone too low).

**TABLE 9-3: TIMEOUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up		Brown-out Reset	Wake-up from SLEEP
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$		
XT, HS, LP	72 ms + 1024 T <sub>osc</sub>	1024 T <sub>osc</sub>	72 ms + 1024 T <sub>osc</sub>	1024 T <sub>osc</sub>
RC	72 ms	—	72 ms	—

**TABLE 9-4: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	x	x	Brown-out Reset
1	1	0	u	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
1	1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP

Legend: x = unknown, u = unchanged

# PIC16C432

**TABLE 9-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --uu
MCLR Reset during SLEEP	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 uuuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	000x xuuu	---- --u0
Interrupt Wake-up from SLEEP	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and global enable bit GIE is set, the PC is loaded with the interrupt vector (0004h) after execution of PC+1.

**TABLE 9-6: INITIALIZATION CONDITION FOR REGISTERS**

Register	Address	Power-on Reset	MCLR Reset during Normal Operation MCLR Reset during SLEEP WDT Reset Brown-out Reset <sup>(1)</sup>	Wake-up from SLEEP through Interrupt Wake-up from SLEEP through WDT Timeout
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	—	-	-
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000 0000	0000 0000	PC + 1 <sup>(3)</sup>
STATUS	03h	0001 1xxx	000q quuu <sup>(4)</sup>	uuuq quuu <sup>(4)</sup>
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CMCON	1Fh	00-- 0000	00-- 0000	uu-- uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uqqq <sup>(2)</sup>
PIR1	0Ch	-0-- ----	-0-- ----	-q-- ---- <sup>(2,5)</sup>
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	-0-- ----	-0-- ----	-u-- ----
PCON	8Eh	---- --0x	---- --uq <sup>(1,6)</sup>	---- --uu
LININTF	90h	---- -111	---- -1-1	---- -1-1
VRCON	9Fh	000- 0000	000- 0000	uuu- uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

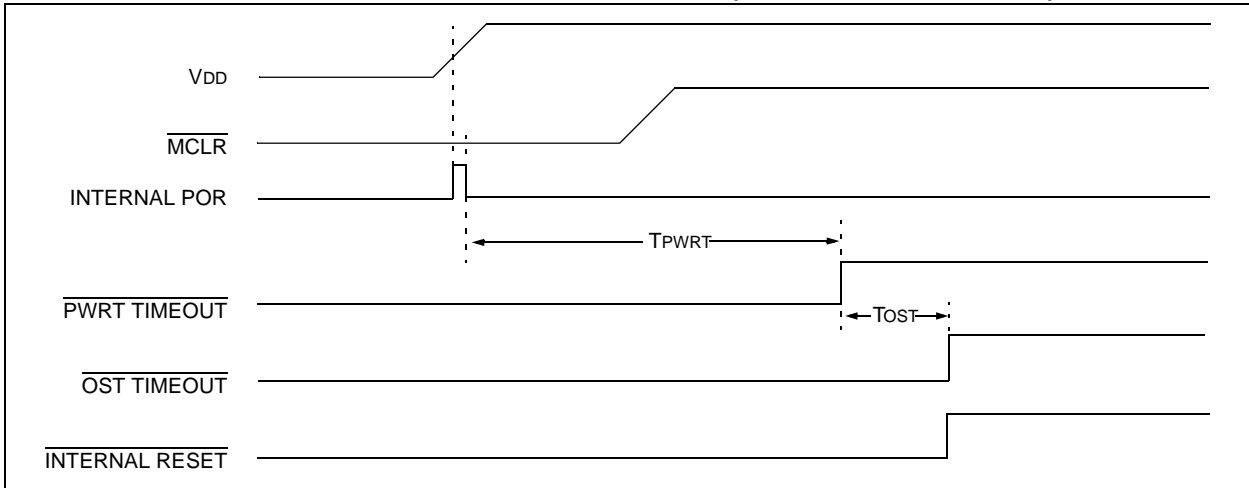
**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 9-5 for RESET value for specific conditions.

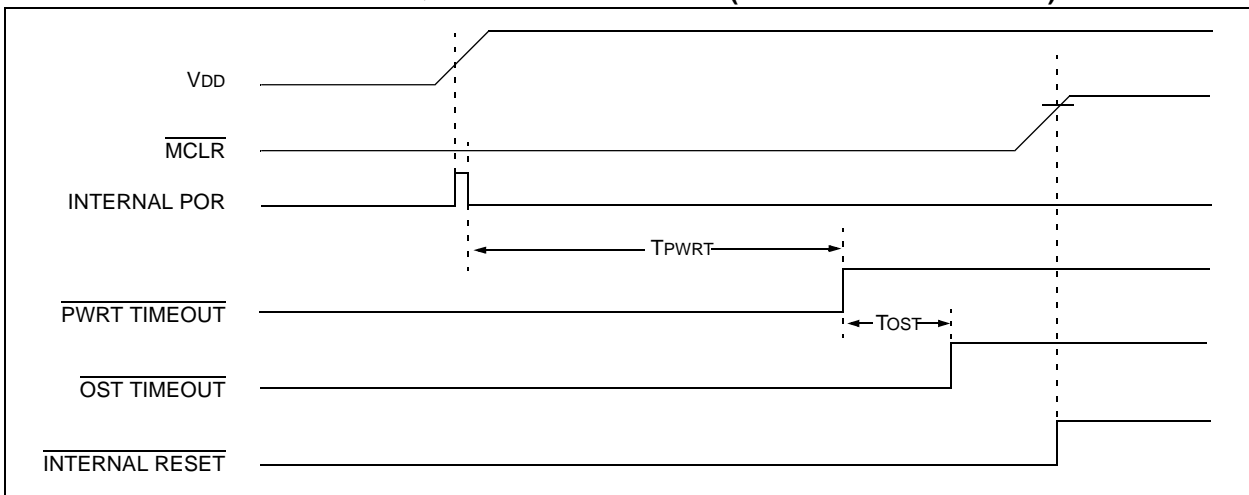
**5:** If wake-up was due to comparator input changing, then bit 6 = 1. All other interrupts generating a wake-up will cause bit 6 = u.

**6:** If RESET was due to brown-out, then PCON bit0 = 0. All other RESETS will cause bit0 = u.

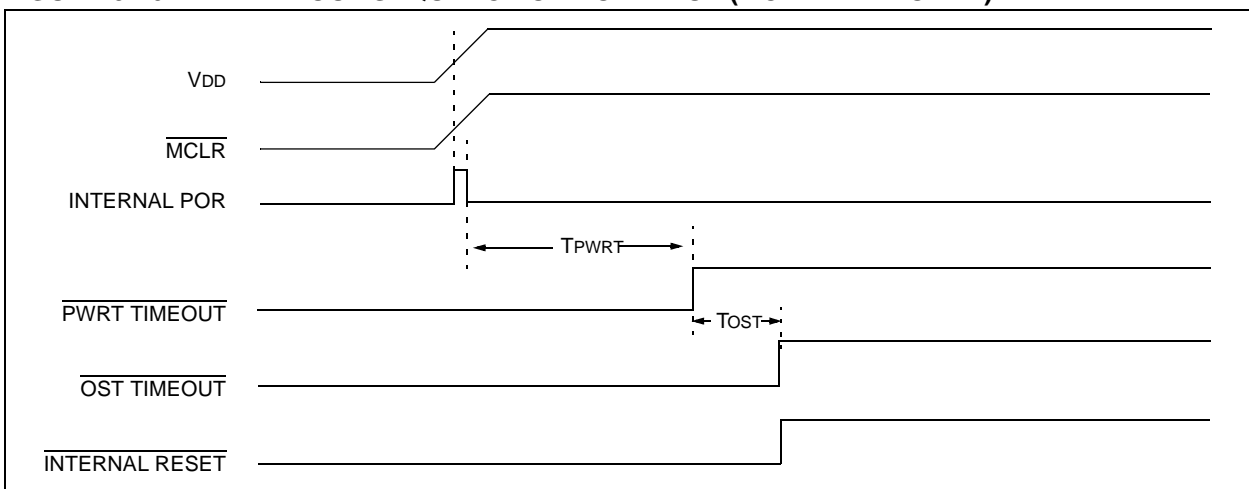
**FIGURE 9-8: TIMEOUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**



**FIGURE 9-9: TIMEOUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**

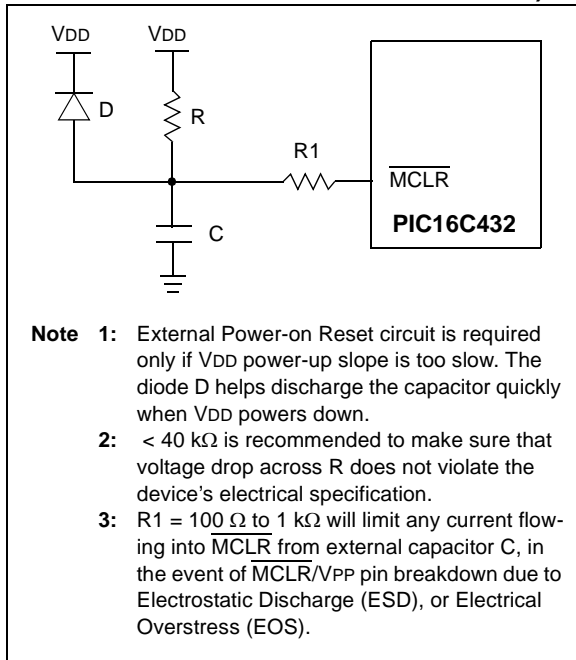


**FIGURE 9-10: TIMEOUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ )**

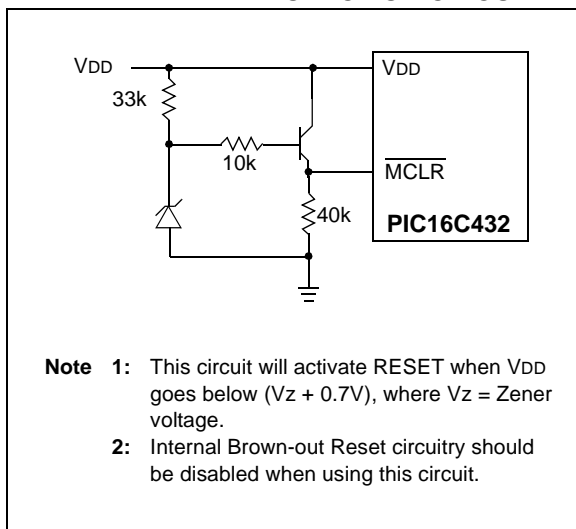


# PIC16C432

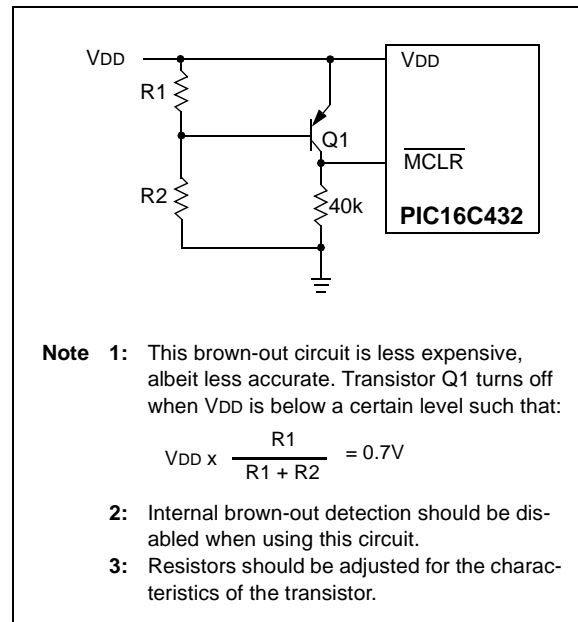
**FIGURE 9-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**



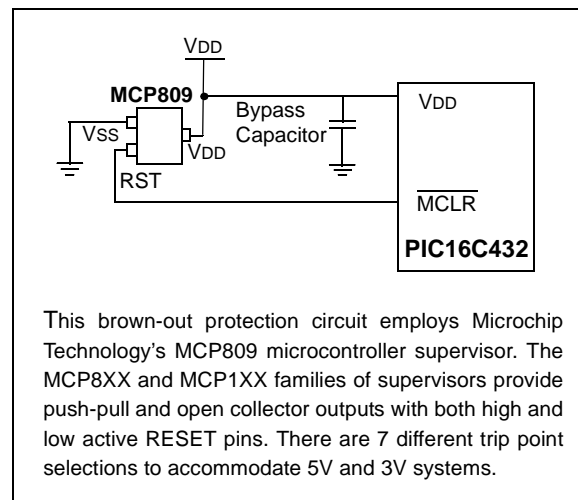
**FIGURE 9-12: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**



**FIGURE 9-13: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**



**FIGURE 9-14: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 3**



## 9.5 Interrupts

The PIC16C432 has 4 sources of interrupt:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB<7:4>)
- Comparator interrupt
- LIN Bus wake-up can be wired to RB0, or comparator

The interrupt control register (INTCON) and the Peripheral Interrupt Register (PIR1) record individual interrupt requests in flag bits. INTCON and PIR1 have 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 all unmasked 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.

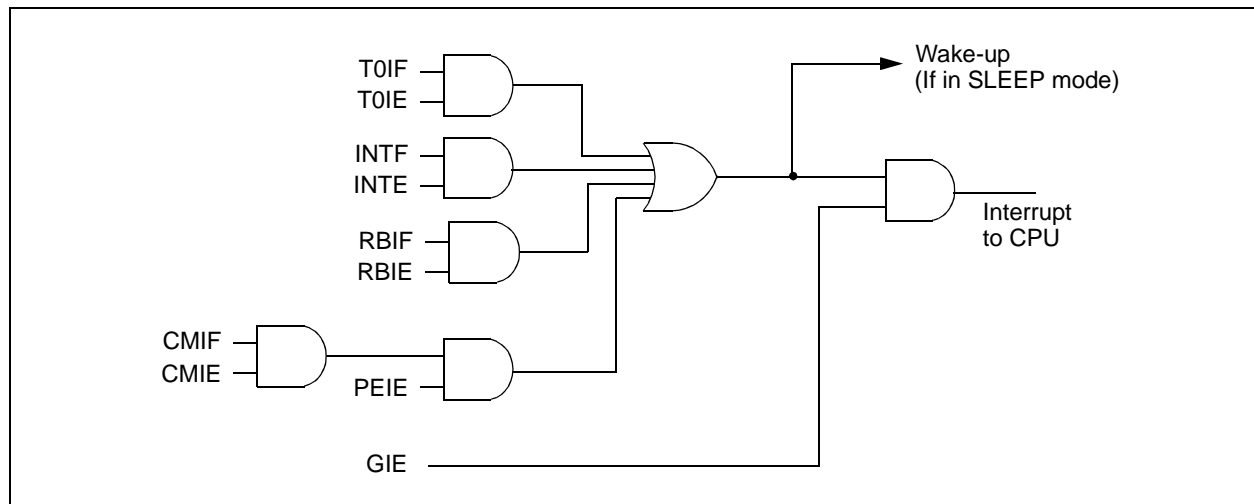
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt 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 on when the interrupt event occurs (Figure 9-16). 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 9-15: INTERRUPT LOGIC**



# PIC16C432

## 9.5.1 RB0/INT INTERRUPT

External interrupt on 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 9.8 for details on SLEEP and Figure 9-18 for timing of wake-up from SLEEP through RB0/INT interrupt.

## 9.5.2 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.

## 9.5.3 PORTB INTERRUPT

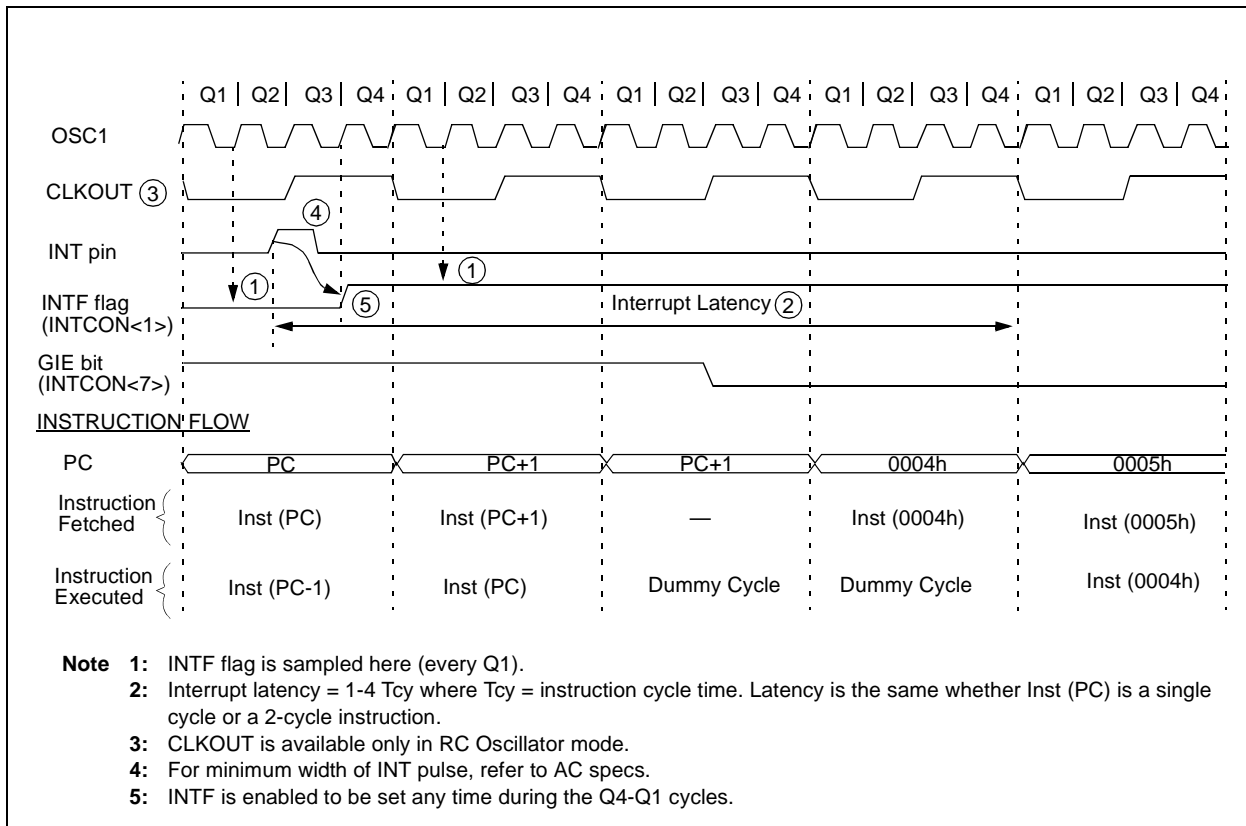
An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB (Section 4.2).

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

## 9.5.4 COMPARATOR INTERRUPT

See Section 7.6 for complete description of comparator interrupts.

**FIGURE 9-16: INT PIN INTERRUPT TIMING**



## 9.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e., W register and STATUS register). This will have to be implemented in software.

Example 9-7 stores and restores the STATUS and W registers. The user register, W\_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W\_TEMP is defined at 0x70 in Bank 0 and it must also be defined at 0xF0 in Bank 1). The user register, STATUS\_TEMP, must be defined in Bank 0. The Example 9-7:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

### EXAMPLE 9-7: SAVING THE STATUS AND W REGISTERS IN RAM

```

MOVW   W_TEMP      ;copy W to temp register,
F      ;could be in either bank

SWAP   STATUS,W     ;swap status to be saved
F      ;into W

BCF    STATUS,RP0   ;change to bank 0 regard-
                  ;less
                  ;of current bank

MOVW   STATUS_TEMP  ;save status to bank 0
F      ;register

:
:   (ISR)
:

SWAP   STATUS_TEMP  ;swap STATUS_TEMP regis-
F      ,W           ;ter
                  ;into W, sets bank to
                  ;original
                  ;state

MOVW   STATUS       ;move W into STATUS regis-
F      ;ter

SWAP   W_TEMP,F     ;swap W_TEMP
F

SWAP   W_TEMP,W     ;swap W_TEMP into W
F
    
```

# PIC16C432

## 9.7 Watchdog Timer (WDT)

The Watchdog Timer is a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run even if the clock on the OSC1 and OSC2 pins of the device have been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT timeout generates a device RESET. If the device is in SLEEP mode, a WDT timeout causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as clear (Section 9.1).

### 9.7.1 WDT PERIOD

The WDT has a nominal timeout period of 18 ms, (with no prescaler). The timeout periods vary with temperature, VDD and process variations from part to part (see

DC specs). If longer timeout periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control, by writing to the OPTION register. Thus, timeout periods up to 2.3 seconds can be realized.

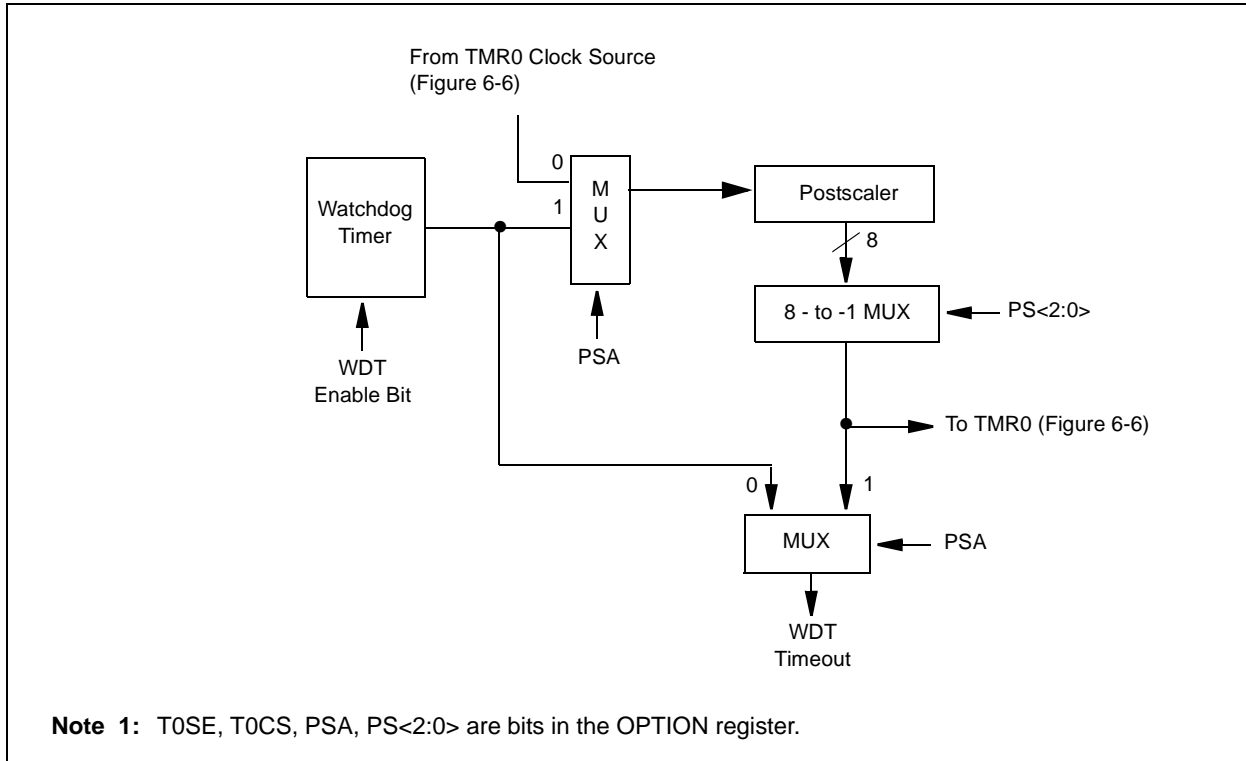
The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

The  $\overline{TO}$  bit in the STATUS register will be cleared upon a Watchdog Timer timeout.

### 9.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT timeout occurs.

**FIGURE 9-17: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 9-8: SUMMARY OF WATCHDOG TIMER REGISTERS**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	—	BOREN	CP1	CP0	$\overline{PWRTE}$	WDTE	FOSC1	FOSC0
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: — = Unimplemented location, read as "0", + = Reserved for future use

**Note 1:** Shaded cells are not used by the Watchdog Timer.



## 9.8 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the PD bit in the STATUS register is cleared, the TO bit is set and the oscillator driver is turned off. The I/O ports maintain the status they had before SLEEP was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin, and the comparators and VREF should be disabled. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The MCLR pin must be at a logic high level (VIHMC).

**Note:** It should be noted that a RESET generated by a WDT timeout does not drive MCLR pin low.

### 9.8.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

1. External RESET input on MCLR pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB Port change, or the Peripheral Interrupt (Comparator).
4. LIN activity.

The first event will cause a device RESET. The two latter events are considered a continuation of program execution. The TO and PD bits in the STATUS register can be used to determine the cause of device RESET. PD bit, which is set on power-up is cleared when SLEEP is invoked. TO bit is cleared if WDT wake-up occurred.

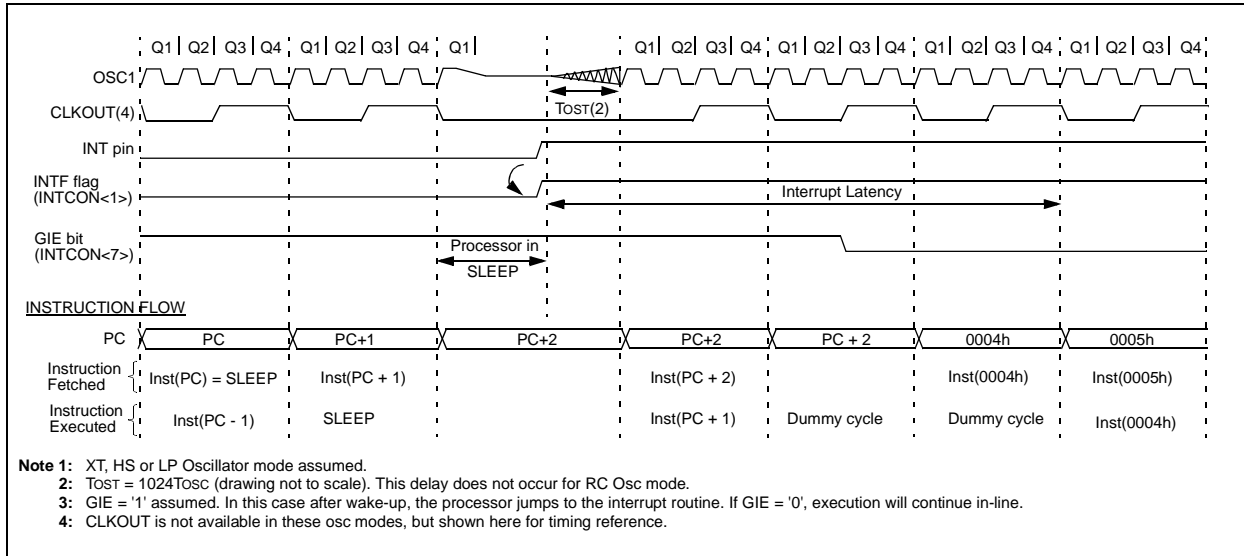
When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have an NOP after the SLEEP instruction.

**Note:** If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from SLEEP. The SLEEP instruction is completely executed.

The WDT is cleared when the device wakes up from SLEEP, regardless of the source of wake-up.

# PIC16C432

**FIGURE 9-18: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 9.9 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

**Note:** Microchip does not recommend code protecting windowed devices.

## 9.10 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during program/verify. Only the Least Significant 4 bits of the ID locations are used.

## 9.11 In-Circuit Serial Programming

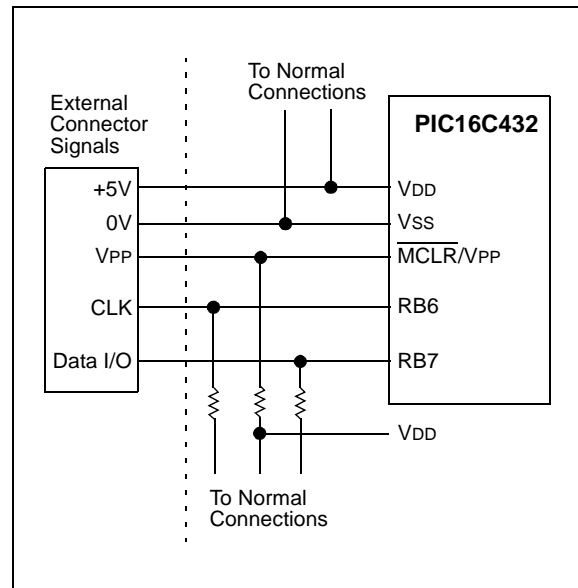
The PIC16C432 microcontroller can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6 and RB7 pins low, while raising the MCLR (VPP) pin from  $V_{IL}$  to  $V_{IH}$  (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After RESET, to place the device into Programming/Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X/9XX Programming Specifications (Literature #DS30228).

A typical in-circuit serial programming connection is shown in Figure 9-19.

**FIGURE 9-19: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 10.0 INSTRUCTION SET SUMMARY

Each PIC16C432 instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C432 instruction set summary in Table 10-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 10-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight- or eleven-bit constant, or literal value.

**TABLE 10-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top-of-Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
TO	Timeout bit
PD	Power-down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
<i>italics</i>	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 10-1 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the three general formats that the instructions can have.

**Note:** To maintain upward compatibility with future PIC® products, do not use the OPTION and TRIS instructions.

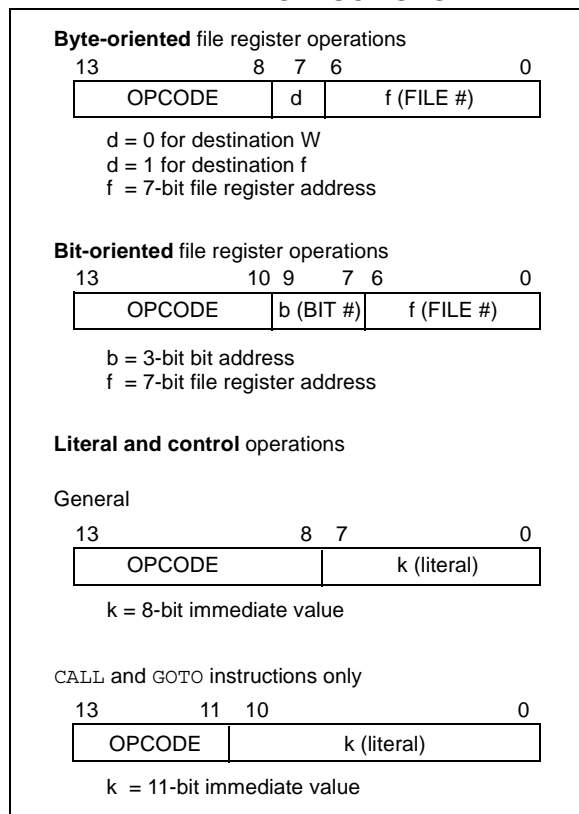
All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

# PIC16C432

**FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS**



**TABLE 10-2: PIC16C432 INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes
			MSb			LSb		
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>								
<b>ADDWF</b> f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
<b>ANDWF</b> f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
<b>CLRF</b> f	Clear f	1	00	0001	1fff	ffff	Z	2
<b>CLRWF</b> -	Clear W	1	00	0001	0000	0011	Z	
<b>COMF</b> f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
<b>DECF</b> f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
<b>DECFSZ</b> f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
<b>INCF</b> f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
<b>INCFSZ</b> f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
<b>IORWF</b> f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
<b>MOVF</b> f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
<b>MOVWF</b> f	Move W to f	1	00	0000	1fff	ffff		
<b>NOP</b> -	No Operation	1	00	0000	0xx0	0000		
<b>RLF</b> f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
<b>RRF</b> f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
<b>SUBWF</b> f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
<b>SWAPF</b> f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
<b>XORWF</b> f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>								
<b>BCF</b> f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
<b>BSF</b> f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
<b>BTFSC</b> f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
<b>BTFSS</b> f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>								
<b>ADDLW</b> k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
<b>ANDLW</b> k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
<b>CALL</b> k	Call subroutine	2	10	0kkk	kkkk	kkkk		
<b>CLRWDT</b> -	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO,PD}$	
<b>GOTO</b> k	Go to address	2	10	1kkk	kkkk	kkkk		
<b>IORLW</b> k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
<b>MOVLW</b> k	Move literal to W	1	11	00xx	kkkk	kkkk		
<b>RETFIE</b> -	Return from interrupt	2	00	0000	0000	1001		
<b>RETLW</b> k	Return with literal in W	2	11	01xx	kkkk	kkkk		
<b>RETURN</b> -	Return from Subroutine	2	00	0000	0000	1000		
<b>SLEEP</b> -	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO,PD}$	
<b>SUBLW</b> k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
<b>XORLW</b> k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note** 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1) the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module
- 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second is executed as a NOP.

# PIC16C432

## 10.1 Instruction Descriptions

### ADDLW Add Literal and W

Syntax:	[ <i>label</i> ] ADDLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	(W) + k → (W)				
Status Affected:	C, DC, Z				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 10px;">11</td><td style="padding: 2px 10px;">111x</td><td style="padding: 2px 10px;">kkkk</td><td style="padding: 2px 10px;">kkkk</td></tr></table>	11	111x	kkkk	kkkk
11	111x	kkkk	kkkk		
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.				
Words:	1				
Cycles:	1				
Example	ADDLW 0x15 Before Instruction W = 0x10 After Instruction W = 0x25				

### ANDLW AND Literal with W

Syntax:	[ <i>label</i> ] ANDLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	(W) .AND. (k) → (W)				
Status Affected:	Z				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 10px;">11</td><td style="padding: 2px 10px;">1001</td><td style="padding: 2px 10px;">kkkk</td><td style="padding: 2px 10px;">kkkk</td></tr></table>	11	1001	kkkk	kkkk
11	1001	kkkk	kkkk		
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.				
Words:	1				
Cycles:	1				
Example	ANDLW 0x5F Before Instruction W = 0xA3 After Instruction W = 0x03				

### ADDWF Add W and f

Syntax:	[ <i>label</i> ] ADDWF f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	(W) + (f) → (dest)				
Status Affected:	C, DC, Z				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 10px;">00</td><td style="padding: 2px 10px;">0111</td><td style="padding: 2px 10px;">dfff</td><td style="padding: 2px 10px;">ffff</td></tr></table>	00	0111	dfff	ffff
00	0111	dfff	ffff		
Description:	Add the contents of the W register with 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	ADDWF FSR, 0 Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0xD9 FSR = 0xC2				

### ANDWF AND W with f

Syntax:	[ <i>label</i> ] ANDWF f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	(W) .AND. (f) → (dest)				
Status Affected:	Z				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="padding: 2px 10px;">00</td><td style="padding: 2px 10px;">0101</td><td style="padding: 2px 10px;">dfff</td><td style="padding: 2px 10px;">ffff</td></tr></table>	00	0101	dfff	ffff
00	0101	dfff	ffff		
Description:	AND the W register with 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	ANDWF FSR, 1 Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0x17 FSR = 0x02				

## BCF Bit Clear f

Syntax: [ *label* ] BCF f,b  
 Operands:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 Operation:  $0 \rightarrow (f<b>)$   
 Status Affected: None  
 Encoding: 

01	00bb	bfff	ffff
----	------	------	------

  
 Description: Bit 'b' in register 'f' is cleared.  
 Words: 1  
 Cycles: 1  
 Example BCF FLAG\_REG, 7

Before Instruction  
 FLAG\_REG = 0xC7  
 After Instruction  
 FLAG\_REG = 0x47

## BTFSC Bit Test, Skip if Clear

Syntax: [ *label* ] BTFSC f,b  
 Operands:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 Operation: skip if (f<b>) = 0  
 Status Affected: None  
 Encoding: 

01	10bb	bfff	ffff
----	------	------	------

  
 Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', 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    BTFSC  FLAG,1
FALSE   GOTO  PROCESS_CODE
TRUE    .
        .
        .
```

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

## BSF Bit Set f

Syntax: [ *label* ] BSF f,b  
 Operands:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 Operation:  $1 \rightarrow (f<b>)$   
 Status Affected: None  
 Encoding: 

01	01bb	bfff	ffff
----	------	------	------

  
 Description: Bit 'b' in register 'f' is set.  
 Words: 1  
 Cycles: 1  
 Example BSF FLAG\_REG, 7

Before Instruction  
 FLAG\_REG = 0x0A  
 After Instruction  
 FLAG\_REG = 0x8A

# PIC16C432

## BTFSS Bit Test f, Skip if Set

Syntax:	[ <i>label</i> ] BTFSS f,b				
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7				
Operation:	skip if (f<b>) = 1				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">01</td> <td style="padding: 2px 10px;">11bb</td> <td style="padding: 2px 10px;">bfff</td> <td style="padding: 2px 10px;">ffff</td> </tr> </table>	01	11bb	bfff	ffff
01	11bb	bfff	ffff		
Description:	<p>If bit 'b' in register 'f' is '1' then the next instruction is skipped.</p> <p>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.</p>				
Words:	1				
Cycles:	1(2)				
Example	<pre> HERE    BTFSS    FLAG,1 FALSE   GOTO    PROCESS_CODE TRUE    .         .         . </pre> <p>Before Instruction PC = address HERE</p> <p>After Instruction if FLAG&lt;1&gt; = 0, PC = address FALSE if FLAG&lt;1&gt; = 1, PC = address TRUE</p>				

## CALL Call Subroutine

Syntax:	[ <i>label</i> ] CALL k				
Operands:	0 ≤ k ≤ 2047				
Operation:	(PC) + 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">10</td> <td style="padding: 2px 10px;">0kkk</td> <td style="padding: 2px 10px;">kkkk</td> <td style="padding: 2px 10px;">kkkk</td> </tr> </table>	10	0kkk	kkkk	kkkk
10	0kkk	kkkk	kkkk		
Description:	<p>Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits &lt;10:0&gt;. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.</p>				
Words:	1				
Cycles:	2				
Example	<pre> HERE    CALL    THERE </pre> <p>Before Instruction PC = Address HERE</p> <p>After Instruction PC = Address THERE TOS = Address HERE+1</p>				

## CLRF Clear f

Syntax:	[ <i>label</i> ] CLRF f				
Operands:	0 ≤ f ≤ 127				
Operation:	00h → (f) 1 → Z				
Status Affected:	Z				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">00</td> <td style="padding: 2px 10px;">0001</td> <td style="padding: 2px 10px;">1fff</td> <td style="padding: 2px 10px;">ffff</td> </tr> </table>	00	0001	1fff	ffff
00	0001	1fff	ffff		
Description:	<p>The contents of register 'f' are cleared and the Z bit is set.</p>				
Words:	1				
Cycles:	1				
Example	<pre> CLRF    FLAG_REG </pre> <p>Before Instruction FLAG_REG = 0x5A</p> <p>After Instruction FLAG_REG = 0x00 Z = 1</p>				



CLRW	Clear W				
Syntax:	[ <i>label</i> ] CLRW				
Operands:	None				
Operation:	00h → (W) 1 → Z				
Status Affected:	Z				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0001</td> <td>0000</td> <td>0011</td> </tr> </table>	00	0001	0000	0011
00	0001	0000	0011		
Description:	W register is cleared. Zero bit (Z) is set.				
Words:	1				
Cycles:	1				
Example	CLRW Before Instruction W = 0x5A After Instruction W = 0x00 Z = 1				

COMF	Complement f				
Syntax:	[ <i>label</i> ] COMF f,d				
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				
Operation:	( $\bar{f}$ ) → (dest)				
Status Affected:	Z				
Encoding:	<table border="1"> <tr> <td>00</td> <td>1001</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	1001	dfff	ffff
00	1001	dfff	ffff		
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.				
Words:	1				
Cycles:	1				
Example	COMF REG1, 0 Before Instruction REG1 = 0x13 After Instruction REG1 = 0x13 W = 0xEC				

CLRWDW	Clear Watchdog Timer				
Syntax:	[ <i>label</i> ] CLRWDW				
Operands:	None				
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$				
Status Affected:	$\overline{TO}$ , $\overline{PD}$				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0100</td> </tr> </table>	00	0000	0110	0100
00	0000	0110	0100		
Description:	CLRWDW instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.				
Words:	1				
Cycles:	1				
Example	CLRWDW Before Instruction WDT counter = ? After Instruction WDT counter = 0x00 WDT prescaler = 0 $\overline{TO}$ = 1 $\overline{PD}$ = 1				

DECF	Decrement f				
Syntax:	[ <i>label</i> ] DECF f,d				
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				
Operation:	(f) - 1 → (dest)				
Status Affected:	Z				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0011</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	0011	dfff	ffff
00	0011	dfff	ffff		
Description:	Decrement 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	DECF CNT, 1 Before Instruction CNT = 0x01 Z = 0 After Instruction CNT = 0x00 Z = 1				

# PIC16C432

## DECFSZ      Decrement f, Skip if 0

Syntax:            [ *label* ] DECFSZ f,d  
 Operands:         $0 \leq f \leq 127$   
                      $d \in [0,1]$   
 Operation:         $(f) - 1 \rightarrow (\text{dest});$  skip if result = 0  
 Status Affected: None

Encoding:        

00	1011	dfff	ffff
----	------	------	------

Description:     The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.  
 If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two-cycle instruction.

Words:            1  
 Cycles:           1(2)

Example            HERE      DECFSZ    CNT, 1  
     GOTO      LOOP  
     CONTINUE •  
     •  
     •

Before Instruction  
                          PC = address HERE  
 After Instruction  
                          CNT = CNT - 1  
                          if CNT = 0,  
                          PC = address CONTINUE  
                          if CNT  $\neq$  0,  
                          PC = address HERE+1

## GOTO            Unconditional Branch

Syntax:            [ *label* ] GOTO k  
 Operands:         $0 \leq k \leq 2047$   
 Operation:         $k \rightarrow \text{PC}\langle 10:0 \rangle$   
                      $\text{PCLATH}\langle 4:3 \rangle \rightarrow \text{PC}\langle 12:11 \rangle$   
 Status Affected: None

Encoding:        

10	1kkk	kkkk	kkkk
----	------	------	------

Description:     GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

Words:            1  
 Cycles:           2

Example            GOTO THERE  
 After Instruction  
                          PC = Address THERE

## INCF            Increment f

Syntax:            [ *label* ] INCF f,d  
 Operands:         $0 \leq f \leq 127$   
                      $d \in [0,1]$   
 Operation:         $(f) + 1 \rightarrow (\text{dest})$   
 Status Affected: Z

Encoding:        

00	1010	dfff	ffff
----	------	------	------

Description:     The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

Words:            1  
 Cycles:           1

Example            INCF      CNT, 1  
 Before Instruction  
                          CNT = 0xFF  
                          Z = 0  
 After Instruction  
                          CNT = 0x00  
                          Z = 1

INCFSZ	Increment f, Skip if 0				
Syntax:	[ <i>label</i> ] INCFSZ f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	$(f) + 1 \rightarrow (\text{dest})$ , skip if result = 0				
Status Affected:	None				
Encoding:	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 5px;">00</td> <td style="border: 1px solid black; padding: 2px 5px;">1111</td> <td style="border: 1px solid black; padding: 2px 5px;">dfff</td> <td style="border: 1px solid black; padding: 2px 5px;">ffff</td> </tr> </table>	00	1111	dfff	ffff
00	1111	dfff	ffff		
Description:	<p>The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.</p> <p>If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two-cycle instruction.</p>				
Words:	1				
Cycles:	1(2)				
Example	<pre> HERE      INCFSZ     CNT, 1                  GOTO     LOOP CONTINUE  . . .                 </pre> <p>Before Instruction  PC = address HERE</p> <p>After Instruction  CNT = CNT + 1  if CNT = 0,  PC = address CONTINUE  if CNT≠0,  PC = address HERE +1</p>				

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:	Z				
Encoding:	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 5px;">11</td> <td style="border: 1px solid black; padding: 2px 5px;">1000</td> <td style="border: 1px solid black; padding: 2px 5px;">kkkk</td> <td style="border: 1px solid black; padding: 2px 5px;">kkkk</td> </tr> </table>	11	1000	kkkk	kkkk
11	1000	kkkk	kkkk		
Description:	<p>The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.</p>				
Words:	1				
Cycles:	1				
Example	<pre> IORLW     0x35 W  Before Instruction           W = 0x9A After Instruction           W = 0xBF           Z = 1                 </pre>				

IORWF	Inclusive OR W with f				
Syntax:	[ <i>label</i> ] IORWF f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	$(W) .OR. (f) \rightarrow (\text{dest})$				
Status Affected:	Z				
Encoding:	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 5px;">00</td> <td style="border: 1px solid black; padding: 2px 5px;">0100</td> <td style="border: 1px solid black; padding: 2px 5px;">dfff</td> <td style="border: 1px solid black; padding: 2px 5px;">ffff</td> </tr> </table>	00	0100	dfff	ffff
00	0100	dfff	ffff		
Description:	<p>Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.</p>				
Words:	1				
Cycles:	1				
Example	<pre> IORWF     RESULT, 0  Before Instruction           RESULT = 0x13           W = 0x91 After Instruction           RESULT = 0x13           W = 0x93           Z = 1                 </pre>				

# PIC16C432

## MOVLW Move Literal to W

Syntax: [ *label* ] MOVLW *k*

Operands:  $0 \leq k \leq 255$

Operation:  $k \rightarrow (W)$

Status Affected: None

Encoding: 

11	00xx	kkkk	kkkk
----	------	------	------

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example  
MOVLW 0x5A  
After Instruction  
W = 0x5A

## MOVF Move f

Syntax: [ *label* ] MOVF *f*,*d*

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

Operation:  $(f) \rightarrow (\text{dest})$

Status Affected: Z

Encoding: 

00	1000	dfff	ffff
----	------	------	------

Description: The contents of register *f* are moved to a destination dependant upon the status of *d*. If *d* = 0, destination is W register. If *d* = 1, the destination is file register *f* itself. *d* = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example  
MOVF FSR, 0  
After Instruction  
W = value in FSR register  
Z = 1

## MOVWF Move W to f

Syntax: [ *label* ] MOVWF *f*

Operands:  $0 \leq f \leq 127$

Operation:  $(W) \rightarrow (f)$

Status Affected: None

Encoding: 

00	0000	1fff	ffff
----	------	------	------

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example  
MOVWF OPTION  
Before Instruction  
OPTION = 0xFF  
W = 0x4F  
After Instruction  
OPTION = 0x4F  
W = 0x4F

## NOP No Operation

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding: 

00	0000	0xx0	0000
----	------	------	------

Description: No operation.

Words: 1

Cycles: 1

Example  
NOP

OPTION	Load Option Register				
Syntax:	[ <i>label</i> ] OPTION				
Operands:	None				
Operation:	(W) → OPTION				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0010</td> </tr> </table>	00	0000	0110	0010
00	0000	0110	0010		
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.				
Words:	1				
Cycles:	1				
Example	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p><b>To maintain upward compatibility with future PIC® products, do not use this instruction.</b></p> </div>				

RETLW	Return with Literal in W				
Syntax:	[ <i>label</i> ] RETLW k				
Operands:	0 ≤ k ≤ 255				
Operation:	k → (W); TOS → PC				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>11</td> <td>01xx</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	11	01xx	kkkk	kkkk
11	01xx	kkkk	kkkk		
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.				
Words:	1				
Cycles:	2				
Example	<pre>CALL TABLE ;W contains table                 ;offset value                 ;W now has table                 value                 .                 .                 .                 ADDWF PC ;W = offset                 RETLW k1 ;Begin table                 RETLW k2 ;                 .                 .                 .                 RETLW kn ; End of table</pre> <p>Before Instruction W = 0x07</p> <p>After Instruction W = value of k8</p>				

RETFIE	Return from Interrupt				
Syntax:	[ <i>label</i> ] RETFIE				
Operands:	None				
Operation:	TOS → PC, 1 → GIE				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0000</td> <td>1001</td> </tr> </table>	00	0000	0000	1001
00	0000	0000	1001		
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.				
Words:	1				
Cycles:	2				
Example	<pre>RETFIE</pre> <p>After Interrupt PC = TOS GIE = 1</p>				

RETURN	Return from Subroutine				
Syntax:	[ <i>label</i> ] RETURN				
Operands:	None				
Operation:	TOS → PC				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0000</td> <td>1000</td> </tr> </table>	00	0000	0000	1000
00	0000	0000	1000		
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.				
Words:	1				
Cycles:	2				
Example	<pre>RETURN</pre> <p>After Interrupt PC = TOS</p>				

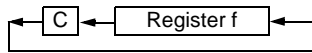
# PIC16C432

## RLF Rotate Left f through Carry

Syntax: [ *label* ] RLF f,d  
 Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 Operation: See description below  
 Status Affected: C  
 Encoding: 

00	1101	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



Words: 1  
 Cycles: 1  
 Example: RLF REG1,0

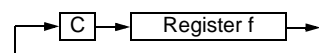
Before Instruction  
 REG1 = 1110 0110  
 C = 0  
 After Instruction  
 REG1 = 1110 0110  
 W = 1100 1100  
 C = 1

## RRF Rotate Right f through Carry

Syntax: [ *label* ] RRF f,d  
 Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 Operation: See description below  
 Status Affected: C  
 Encoding: 

00	1100	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



Words: 1  
 Cycles: 1  
 Example: RRF REG1,0

Before Instruction  
 REG1 = 1110 0110  
 C = 0  
 After Instruction  
 REG1 = 1110 0110  
 W = 0111 0011  
 C = 0

## SLEEP

Syntax: [ *label* ] SLEEP  
 Operands: None  
 Operation: 00h → WDT,  
 0 → WDT prescaler,  
 1 →  $\overline{TO}$ ,  
 0 →  $\overline{PD}$   
 Status Affected:  $\overline{TO}$ ,  $\overline{PD}$   
 Encoding: 

00	0000	0110	0011
----	------	------	------

Description: The power-down status bit,  $\overline{PD}$  is cleared. Timeout status bit,  $\overline{TO}$  is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 9.8 for more details.

Words: 1  
 Cycles: 1  
 Example: SLEEP

## SUBLW Subtract W from Literal

Syntax: [ *label* ] SUBLW *k*

Operands:  $0 \leq k \leq 255$

Operation:  $k - (W) \rightarrow (W)$

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk
----	------	------	------

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1: `SUBLW 0x02`

Before Instruction

W = 1  
C = ?

After Instruction

W = 1  
C = 1; result is positive

Example 2: Before Instruction

W = 2  
C = ?

After Instruction

W = 0  
C = 1; result is zero

Example 3: Before Instruction

W = 3  
C = ?

After Instruction

W = 0xFF  
C = 0; result is negative

## SUBWF Subtract W from f

Syntax: [ *label* ] SUBWF *f,d*

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

Operation:  $(f) - (W) \rightarrow (\text{dest})$

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

Example 2: Before Instruction

REG1 = 2  
W = 2  
C = ?

After Instruction

REG1 = 0  
W = 2  
C = 1; result is zero

Example 3: Before Instruction

REG1 = 1  
W = 2  
C = ?

After Instruction

REG1 = 0xFF  
W = 2  
C = 0; result is negative

# PIC16C432

SWAPF	Swap Nibbles in f				
Syntax:	[ <i>label</i> ] SWAPF f,d				
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				
Operation:	(f<3:0>) → (dest<7:4>), (f<7:4>) → (dest<3:0>)				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>1110</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	1110	dfff	ffff
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	<pre>SWAPF REG, 0</pre> <p>Before Instruction REG1 = 0xA5</p> <p>After Instruction REG1 = 0xA5 W = 0x5A</p>				

XORLW	Exclusive OR Literal with W				
Syntax:	[ <i>label</i> ] XORLW k				
Operands:	0 ≤ k ≤ 255				
Operation:	(W) .XOR. k → (W)				
Status Affected:	Z				
Encoding:	<table border="1"> <tr> <td>11</td> <td>1010</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	11	1010	kkkk	kkkk
11	1010	kkkk	kkkk		
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.				
Words:	1				
Cycles:	1				
Example:	<pre>XORLW 0xAF</pre> <p>W</p> <p>Before Instruction W = 0xB5</p> <p>After Instruction W = 0x1A</p>				

TRIS	Load TRIS Register				
Syntax:	[ <i>label</i> ] TRIS f				
Operands:	5 ≤ f ≤ 7				
Operation:	(W) → TRIS register f;				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0fff</td> </tr> </table>	00	0000	0110	0fff
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	<p><b>To maintain upward compatibility with future PIC® products, do not use this instruction.</b></p>				

XORWF	Exclusive OR W with f				
Syntax:	[ <i>label</i> ] XORWF f,d				
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				
Operation:	(W) .XOR. (f) → (dest)				
Status Affected:	Z				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0110</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	0110	dfff	ffff
00	0110	dfff	ffff		
Description:	Exclusive OR the contents of the W register with 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	<pre>XORW REG 1</pre> <p>F</p> <p>Before Instruction REG = 0xAF W = 0xB5</p> <p>After Instruction REG = 0x1A W = 0xB5</p>				



## 11.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ® Demonstration Board

### 11.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- On-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

### 11.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PIC MCUs.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. 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, an absolute LST file that contains source lines and generated machine code, and a COD file 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.

### 11.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI 'C' compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

## 11.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another 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 MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

## 11.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

## 11.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

## 11.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

## 11.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PIC MCUs and can be used to develop for this and other PIC microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

## 11.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in Stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In Stand-alone mode, the PRO MATE II device programmer can read, verify, or program PIC devices. It can also set code protection in this mode.

## 11.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PIC devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

## 11.11 PICDEM 1 Low Cost PIC MCU Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

## 11.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I<sup>2</sup>C™ bus and separate headers for connection to an LCD module and a keypad.

## 11.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

## 11.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## 11.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

**TABLE 11-1: DEVELOPMENT TOOLS FROM MICROCHIP**

Tool	PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXX	PIC16C7X	PIC16C8X	PIC16F8XX	PIC16G9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	PIC18FXX	24CXX/ 25CXX/ 93CXX	HCSXX	MCRFXXX	MCP2510
<b>Software Tools</b>																	
MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB® C17 C Compiler																	
MPLAB® C18 C Compiler																	
MPASM™ Assembler/ MPLINK™ Object Linker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB® ICE In-Circuit Emulator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ICEPIC™ In-Circuit Emulator	✓		✓	✓	✓	✓	✓	✓	✓								
<b>Debugger</b>																	
MPLAB® ICD In-Circuit Debugger				✓*		✓*		✓					✓				
<b>Programmers</b>																	
PICSTART® Plus Entry Level Development Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PRO MATE® II Universal Device Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Demo Boards and Eval Kits</b>																	
PICDEM™ 1 Demonstration Board			✓			↑				✓							
PICDEM™ 2 Demonstration Board				✓		↑						✓					
PICDEM™ 3 Demonstration Board								✓									
PICDEM™ 14A Demonstration Board		✓															
PICDEM™ 17 Demonstration Board										✓							
KEELOQ® Evaluation Kit															✓		
KEELOQ® Transponder Kit															✓		
microID™ Programmer's Kit																✓	
125 kHz microID™ Developer's Kit																✓	
125 kHz Anticollision microID™ Developer's Kit																✓	
13.56 MHz Anticollision microID™ Developer's Kit																✓	
MCP2510 CAN Developer's Kit																✓	✓

\* Contact the Microchip Technology Inc. web site at [www.microchip.com](http://www.microchip.com) for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.

\*\* Contact Microchip Technology Inc. for availability date.

# PIC16C432

---

---

NOTES:

## 12.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings †

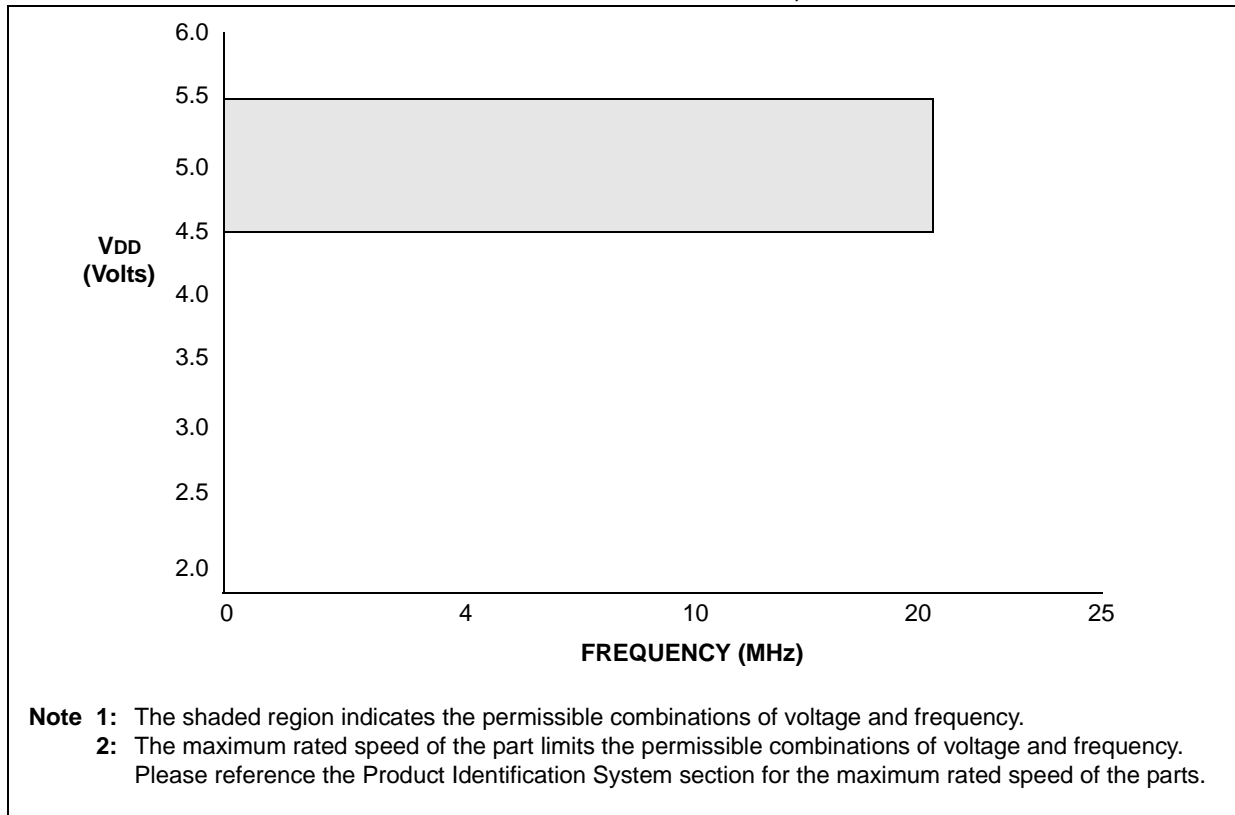
Ambient Temperature under bias .....	-40° to +125°C
Storage Temperature .....	-65° to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{\text{MCLR}}$ ) .....	-0.6V to VDD +0.6V
Voltage on VDD with respect to VSS .....	0 to +7.0V
Voltage on RA4 with respect to VSS .....	8.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS ( <b>Note 2</b> ) .....	0 to +14V
Voltage on RA4 with respect to VSS .....	8.5V
Voltage on LIN with respect to VSS .....	40V
Total power Dissipation ( <b>Note 1</b> ) .....	1.0 W
Maximum Current out of VSS pin .....	300 mA
Maximum Current into VDD pin .....	250 mA
Input clamp current by LIN pin, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>BAT</sub> ) .....	200 mA
Output clamp current by LIN pin, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>BAT</sub> ) .....	200 mA
Input Clamp Current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output Clamp Current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	±20 mA
Maximum Output Current sunk by any I/O pin (source by VDD) .....	25 mA
Maximum Current sourced by any I/O pin (source by VDD) .....	25 mA
Maximum Current sunk by PORTA and PORTB (source by VDD) .....	200 mA
Maximum Current sourced by PORTA and PORTB (source by VDD) .....	200 mA
Maximum Current sunk by LIN pin (source by V <sub>BAT</sub> ) .....	200 mA
Maximum Current sunk by BACT pin (source by V <sub>BAT</sub> ) .....	1.8 mA

**Note 1:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latchup. Thus, a series resistor of 50-100  $\Omega$  should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin, rather than pulling this pin directly to VSS.

† **NOTICE:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions, above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

# PIC16C432

FIGURE 12-1: PIC16C432 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$





## 12.1 DC CHARACTERISTICS: PIC16C432 (Industrial, Extended)

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				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	<b>Supply Voltage</b>	4.5	—	5.5	V	See Figure 12-1 through Figure 12-3
D001A	VBAT	<b>Battery Supply Voltage</b>	8.0	13.8	18	V	
D002	VDR	<b>RAM Data Retention Voltage<sup>(1)</sup></b>	—	1.5*	—	V	Device in SLEEP mode
D003	VPOR	<b>VDD Start Voltage</b> to ensure Power-on Reset	—	VSS	—	V	See section on Power-on Reset for details
D004	SVDD	<b>VDD Rise Rate</b> to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D005	VBOR	<b>Brown-out Detect Voltage</b>	3.7	4.0	4.35	V	BOREN configuration bit is cleared
D010	IDD	<b>Supply Current<sup>(2), (4)</sup></b>	—	1.2	2.0	mA	FOSC = 4 MHz, VDD = 5.5V, WDT disabled, XT Osc mode, <sup>(4)*</sup>
			—	4.0	6.0	mA	FOSC = 20 MHz, VDD = 4.5V, WDT disabled, HS Osc mode
			—	4.0	7.0	mA	FOSC = 20 MHz, VDD = 5.5V, WDT disabled*, HS Osc mode
D020	IPD	<b>Power-down Current<sup>(3)</sup></b>	—	—	5.0	μA	VDD = 4.5V*
			—	—	9.0	μA	VDD = 5.5V
			—	—	15	μA	VDD = 5.5V Extended
D313	ΔIDD-LIN	<b>LIN Transceiver Current<sup>(5)</sup></b>			1	mA	LIN XCVR enabled
D022	ΔIWDT	<b>WDT Current<sup>(5)</sup></b>	—	6.0	10	μA	VDD = 4.0V
					12	μA	(125°C)
D022A	ΔIBOR	<b>Brown-out Reset Current<sup>(5)</sup></b>	—	75	125	μA	BOD enabled, VDD = 5.0V
D023	ΔICOMP	<b>Comparator Current for each Comparator<sup>(5)</sup></b>	—	30	60	μA	VDD = 4.0V
D023A	ΔIVREF	<b>VREF Current<sup>(5)</sup></b>	—	80	135	μA	VDD = 4.0V

\* 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:** This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

**Note 2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption. The test conditions for all IDD measurements in active Operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD,

MCLR = VDD; WDT enabled/disabled as specified.

**Note 3:** 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 hi-impedance state and tied to VDD or VSS.

**Note 4:** For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula  $I_r = VDD/2REXT$  (mA) with REXT in kΩ.

**Note 5:** 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.

**Note 6:** Commercial temperature range only.

# PIC16C432

## 12.2 DC CHARACTERISTICS: PIC16C432 (Industrial, Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial and $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ for extended					
		Operating voltage VDD range as described in DC spec Table 12.3					
Parm No.	Sym	Characteristic	Min	Typ†	Max	Unit	Conditions
	VIL	<b>Input Low Voltage</b>					
D030		I/O ports with TTL buffer	VSS	—	0.8V 0.15 VDD	V	VDD = 4.5V to 5.5V, Otherwise
D031		with Schmitt Trigger input	VSS	—	0.2 VDD	V	<b>(Note 1)</b>
D032		MCLR, RA4/T0CKI, OSC1 (in RC mode)	VSS	—	0.2 VDD	V	
D033		OSC1 (in XT and HS)	VSS	—	0.3 VDD	V	
D034	VIL_LIN	OSC1 (in LP)	VSS	—	0.6 VDD - 1.0	V	
		Low level input voltage	-8	—	0.4 VBAT	V	Dominant State
	VIH	<b>Input High Voltage</b>					
D040		I/O ports with TTL buffer	2.0V .25 VDD + 0.8V	—	VDD VDD	V	VDD = 4.5V to 5.5V
D041		with Schmitt Trigger input	0.8 VDD	—	VDD	V	<b>(Note 1)</b>
D042		MCLR RA4/T0CKI	0.8 VDD	—	VDD	V	
D043		OSC1 (XT, HS and LP)	0.7 VDD	—	VDD	V	
D043A		OSC1 (in RC mode)	0.9 VDD	—		V	
D044	VIH_LIN	High level input voltage	0.6 VBAT	—	18	V	Recessive State
D070	IPURB	<b>PORTB weak pull-up current</b>	50	200	400	μA	VDD = 5.0V, VPIN = VSS
	IIL	<b>Input Leakage Current<sup>(2), (3)</sup></b>					
D060		I/O ports (Except PORTA)	—	—	±1.0	μA	VSS ≤ VPIN ≤ VDD, pin at hi-impedance
D061		PORTA	—	—	±0.5	μA	VSS ≤ VPIN ≤ VDD, pin at hi-impedance
D063		RA4/T0CKI	—	—	±1.0	μA	VSS ≤ VPIN ≤ VDD
		OSC1, MCLR	—	—	±5.0	μA	VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
D064	IOH_LIN	High level output leakage current	—	—	±20	μA	VBUS ≤ VBAT; VBUS < 40V
	VOL	<b>Output Low Voltage</b>					
D080		I/O ports	—	—	0.6	V	IOL=8.5 mA, VDD=4.5V, -40° to +85°C
			—	—	0.6	V	IOL=7.0 mA, VDD=4.5V, +125°C
D083		OSC2/CLKOUT (RC only)	—	—	0.6	V	IOL=1.6 mA, VDD=4.5V, -40° to +85°C
			—	—	0.6	V	IOL=1.2 mA, VDD=4.5V, +125°C
D084		BACT	—	—	TBD	V	TBD
D085	VOL_LIN	Low level output voltage	—	—	0.2 VBAT	V	IOL = 200 mA VBUS = 12V
	VOH	<b>Output High Voltage<sup>(3)</sup></b>					
D090		I/O ports (Except RA4)	VDD-0.7	—	—	V	IOH=-3.0 mA, VDD=4.5V, -40° to +85°C
			VDD-0.7	—	—	V	IOH=-2.5 mA, VDD=4.5V, +125°C
D092		OSC2/CLKOUT (RC only)	VDD-0.7	—	—	V	IOH=-1.3 mA, VDD=4.5V, -40° to +85°C
			VDD-0.7	—	—	V	IOH=-1.0 mA, VDD=4.5V, +125°C
D093	VOH	BACT	4.0V	—	—	V	VBAT = 18V, VDD = 5.0V, IOH = 1.8 mA
D094	VOH_LIN	High level output voltage	0.8 VBAT	—	—	V	
D150*	VOD	<b>Open-Drain High Voltage</b>			8.5	V	RA4 pin

\* 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:** In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C432 be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as coming out of the pin.
- 4:** LIN tested 4 MHz, 14.4V VBAT, 5.0V VDD.

## 12.2 DC CHARACTERISTICS: PIC16C432 (Industrial, Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial and $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ for extended					
		Operating voltage VDD range as described in DC spec Table 12.3					
Param No.	Sym	Characteristic	Min	Typ†	Max	Unit	Conditions
<b>Capacitive Loading Specs on Output Pins</b>							
D100	COsc2	OSC2 pin			15*	pF	In XT, HS and LP modes when external clock used to drive OSC1
100A	CLIN	LIN <sup>(4)</sup>			10*	nF	
100B	CBACT	BACT			50*	pF	
D101	CIO	All I/O pins/OSC2 (in RC mode)			50*	pF	

\* 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: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C432 be driven with external clock in RC mode.
- 2: The leakage current on the MCLR pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3: Negative current is defined as coming out of the pin.
- 4: LIN tested 4 MHz, 14.4V VBAT, 5.0V VDD.

## 12.3 LIN Transceiver Bus Interface Specifications

Operating Conditions: VDD range as described in Table 12-1, $-40^{\circ}\text{C} < \text{TA} < +125^{\circ}\text{C}$							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D315	IOL_LIN_DOMINAT	Low level output current	40	—	200	mA	V <sub>BUS</sub> = 12V
D317	IOH_LIN_REVERS	Low level output current, open ground	-1	—	1	mA	
D320*	VHYS_LIN	Input hysteresis	0.05 V <sub>BAT</sub>	—	0.1V <sub>BAT</sub>	V	V <sub>IH_LIN</sub> - V <sub>IL_LIN</sub>
D321*	ISC_LIN	Short circuit current limit	0.05	—	200	mA	

\* These parameters are characterized but not tested.

# PIC16C432

## 12.4 Comparator Specifications

Operating Conditions: VDD range as described in Table 12-1, -40°C <TA< +125°C							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D300	VIOFF	Input Offset Voltage		± 5.0	± 10	mV	
D301	VICM	Input Common Mode Voltage	0		VDD - 1.5	V	
D302	CMRR	CMRR	+55*			db	
300	TRESP	Response Time <sup>(1)</sup>		150*	400*	ns	
301	TMC2OV	Comparator Mode Change to Output Valid			10*	µs	

\* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at (VDD - 1.5)/2, while the other input transitions from VSS to VDD.

## 12.5 Voltage Reference Specifications.

Operating Conditions: VDD range as described in Table 12-1, -40°C <TA< +125°C							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D310	VRES	Resolution	VDD/24		VDD/32	LSB	
D311	VRAA	Absolute Accuracy			±1/4 ±1/2	LSB LSB	Low Range (VRR=1) High Range (VRR=0)
D312	VRUR	Unit Resistor Value (R)		2K*		Ω	Figure 8.1
310	TSET	Settling Time <sup>(1)</sup>			10*	ms	

\* These parameters are characterized but not tested.

**Note 1:** Settling time measured while VRR = 1 and VR<3:0> transitions from 0000 to 1111.

## 12.6 LIN Transceiver Operating Specifications.

Operating Conditions: VDD range as described in Table 12-1, -40°C <TA< +125°C							
Param No.	Characteristics	Sym	Min	Typ	Max	Units	Comments
D313	VDD Quiescent Operating Current	IDD_LIN	—	—	1	mA	
D314	VBAT Low Power Current	IBAT	—	—	50	µA	

\* These parameters are characterized but not tested.

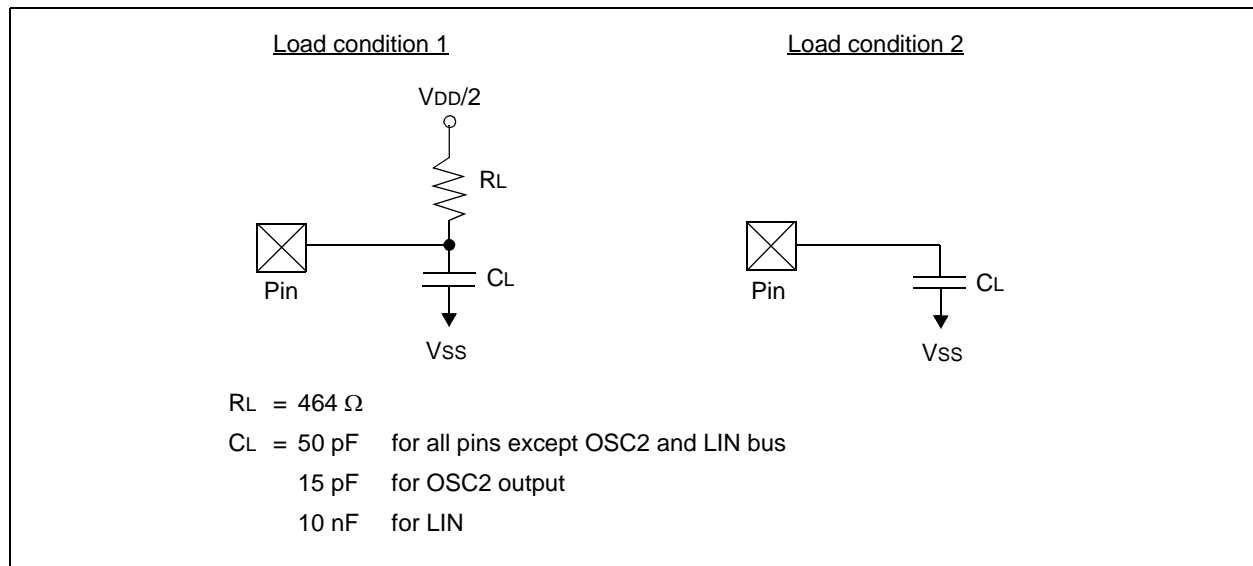
## 12.7 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

<b>T</b>			
F	Frequency	T	Time
Lowercase subscripts (pp) and their meanings:			
<b>pp</b>			
ck	CLKOUT	osc	OSC1
io	I/O port	t0	T0CKI
mc	MCLR		
Uppercase letters and their meanings:			
<b>S</b>			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-Impedance

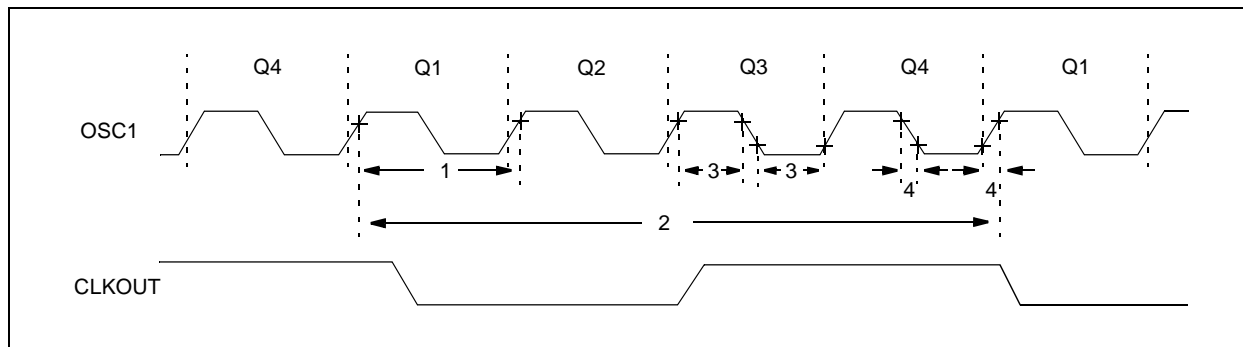
**FIGURE 12-2: LOAD CONDITIONS**



# PIC16C432

## 12.8 Timing Diagrams and Specifications

**FIGURE 12-3: EXTERNAL CLOCK TIMING**



**TABLE 12-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

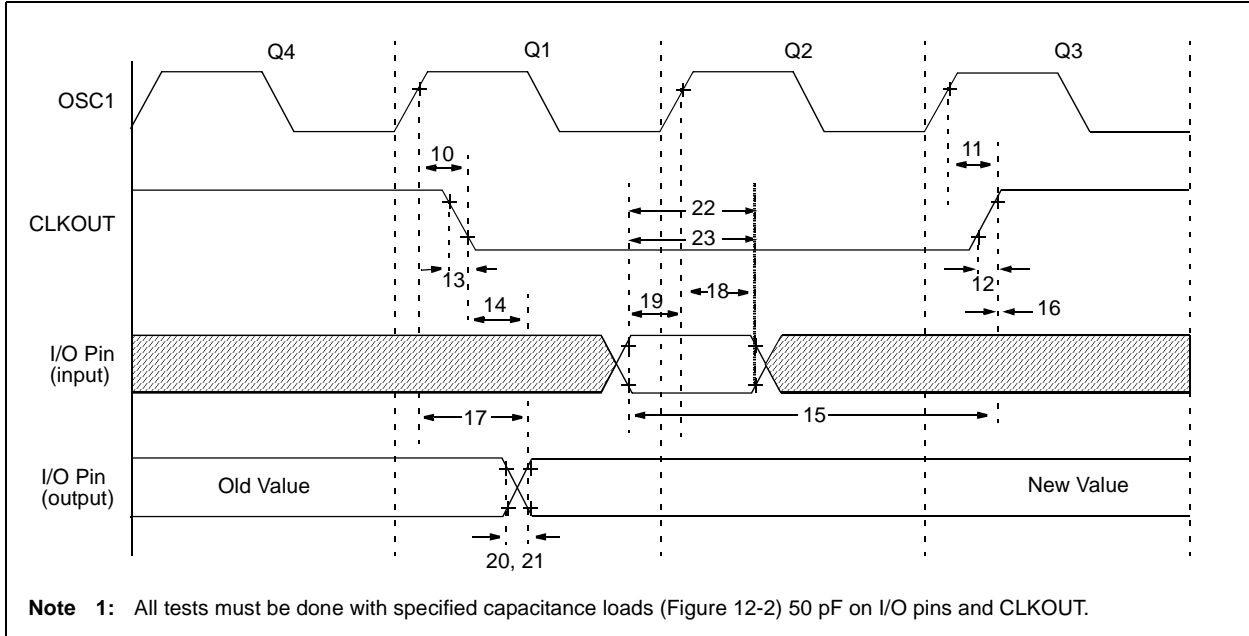
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
1A	Fosc	External CLKIN Frequency <sup>(1)</sup>	DC	—	4	MHz	XT and RC Osc mode, VDD=5.0V
			DC	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
		Oscillator Frequency <sup>(1)</sup>	DC	—	4	MHz	RC Osc mode, VDD=5.0 OV
			0.1	—	4	MHz	XT Osc mode
			1	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
1	Tosc	External CLKIN Period <sup>(1)</sup>	250	—	—	ns	XT and RC Osc mode
			50	—	—	ns	HS Osc mode
			5	—	—	ms	LP Osc mode
		Oscillator Period <sup>(1)</sup>	250	—	—	ns	RC Osc mode
			250	—	10,000	ns	XT Osc mode
			50	—	1,000	ns	HS Osc mode
			5	—	—	ms	LP Osc mode
2	Tcy	Instruction Cycle Time <sup>(1)</sup>	200	—	DC	ns	Tcy=Fosc/4
3*	TosL, TosH	External Clock in (OSC1) High or Low Time	100*	—	—	ns	XT oscillator, TosC L/H duty cycle
			2*	—	—	ms	LP oscillator, TosC L/H duty cycle
			20*	—	—	ns	HS oscillator, TosC L/H duty cycle
4*	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	25*	—	—	ns	XT oscillator
			50*	—	—	ns	LP oscillator
			15*	—	—	ns	HS oscillator

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

**FIGURE 12-4: CLKOUT AND I/O TIMING**



**TABLE 12-2: CLKOUT AND I/O TIMING REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units
10*	TosH2ckL	OSC1↑ to CLKOUT↓ <sup>(1)</sup>	—	75	200	ns
11*	TosH2ckH	OSC1↑ to CLKOUT↑ <sup>(1)</sup>	—	75	200	ns
12*	TckR	CLKOUT rise time <sup>(1)</sup>	—	35	100	ns
13*	TckF	CLKOUT fall time <sup>(1)</sup>	—	35	100	ns
14*	TckL2ioV	CLKOUT↓ to Port out valid <sup>(1)</sup>	—	—	20	ns
15*	TioV2ckH	Port in valid before CLKOUT↑ <sup>(1)</sup>	TOSC +200 ns	—	—	ns
16*	TckH2iol	Port in hold after CLKOUT↑ <sup>(1)</sup>	0	—	—	ns
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns
18*	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	100	—	—	ns
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns
20*	TioR	Port output rise time	—	10	40	ns
21*	TioF	Port output fall time	—	10	40	ns
22*	Tinp	RB0/INT pin high or low time	25	—	—	ns
23	Trbp	RB<7:4> change interrupt high or low time	T <sub>CY</sub>	—	—	ns

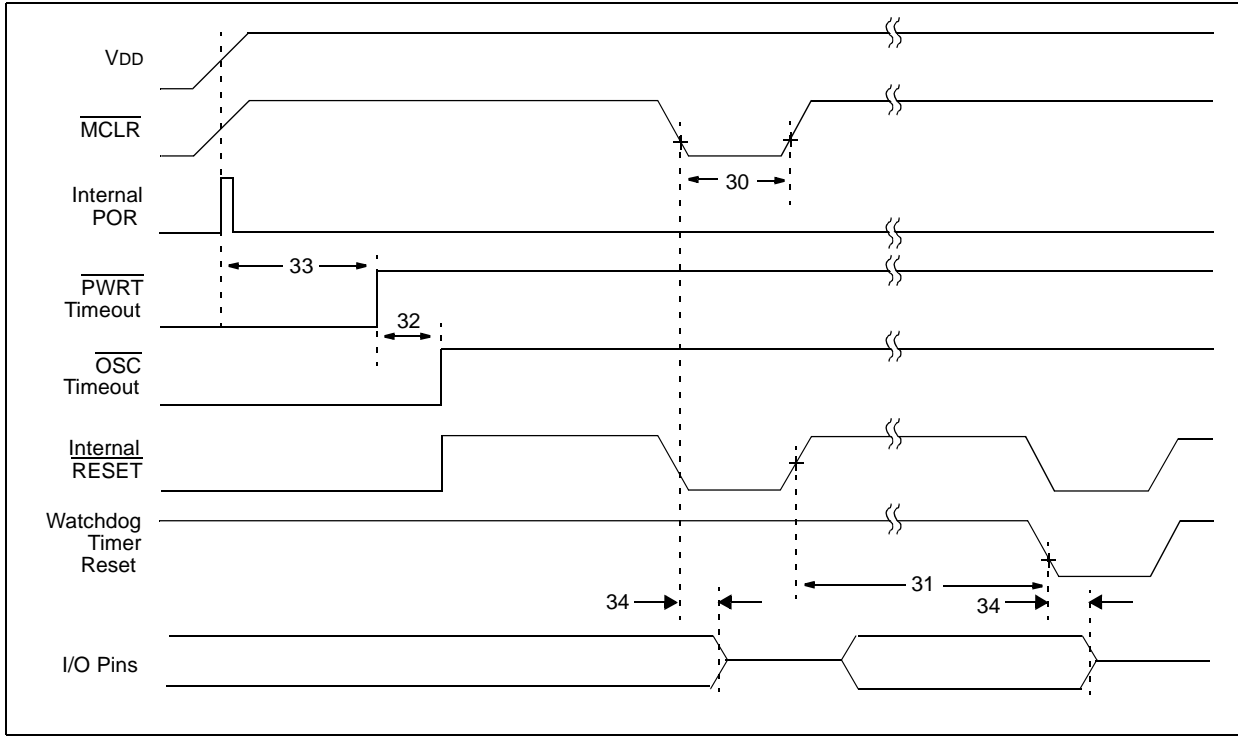
\* These parameters are characterized but not tested.

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

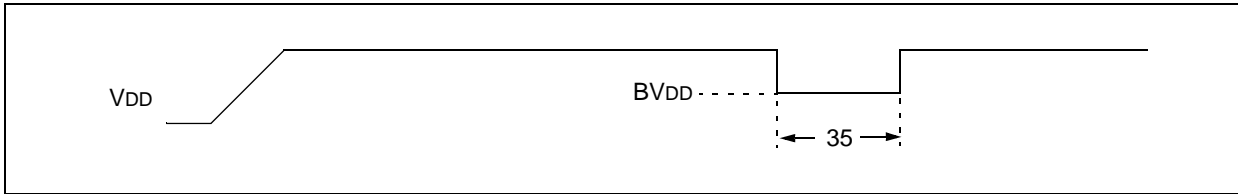
**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

# PIC16C432

**FIGURE 12-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 12-6: BROWN-OUT RESET TIMING**



**TABLE 12-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS**

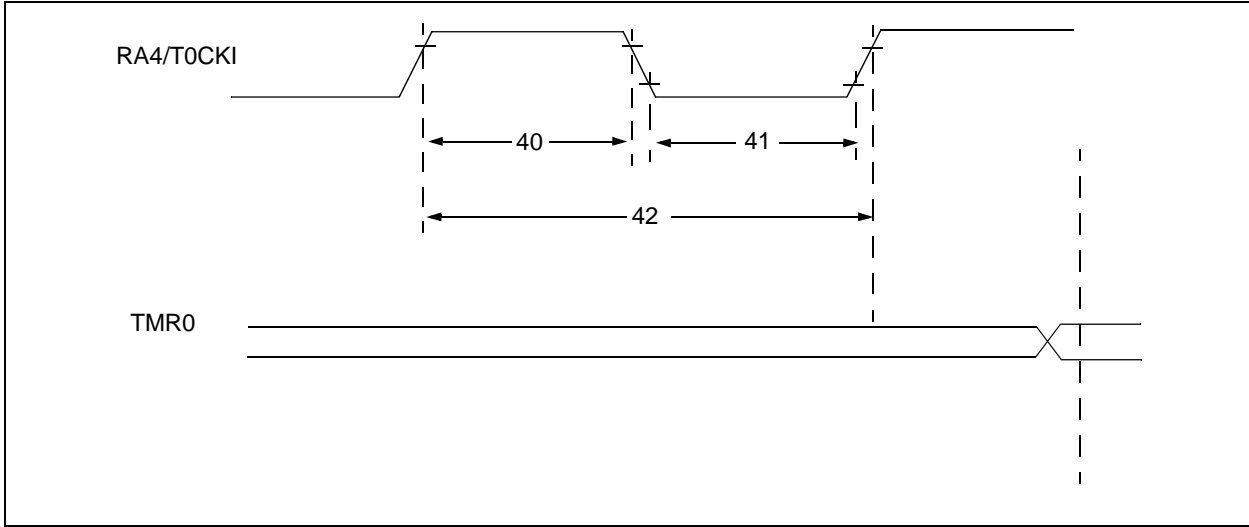
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	Tmcl	MCLR Pulse Width (low)	2000	—	—	ns	-40° to +85°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5.0V, -40° to +85°C
32	Tost	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33	Tpwrt	Power-up Timer Period	28*	72	132*	ms	VDD = 5.0V, -40° to +85°C
34	TIOZ	I/O hi-impedance from MCLR low	—	—	2.0	ms	
35	TBOR	Brown-out Reset Pulse Width	100*	—	—	ms	3.7V ≤ VDD ≤ 4.3V

\* 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 12-7: TIMER0 CLOCK TIMING**



**TABLE 12-4: TIMER0 CLOCK REQUIREMENTS**

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20^*$	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20^*$	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

\* 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.

**TABLE 12-5: LIN AC CHARACTERISTICS**

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
$ dV/dt $	Slope rising and falling edges	1	2	3	V/μs	(Note 1)
$T_{trans\_pd}$	Propagation delay of transmitter			4	μs	$T_{trans\_pd} = \max(T_{trans\_pdr} \text{ or } T_{trans\_pdf})$
$T_{rec\_pd}$	Propagation delay of receiver			6	μs	$T_{rec\_pd} = \max(T_{rec\_pdr} \text{ or } T_{rec\_pdf})$
$T_{rec\_sym}$	Symmetry of receiver propagation delay rising edge w.r.t. falling edge	-2		2	μs	$T_{rec\_sym} = T_{rec\_pdf} - T_{rec\_pdr}$
$T_{trans\_sym}$	Symmetry of transmitter propagation delay rising edge w.r.t. falling edge	-2		2	μs	$T_{trans\_sym} = T_{trans\_pdf} - T_{trans\_pdr}$

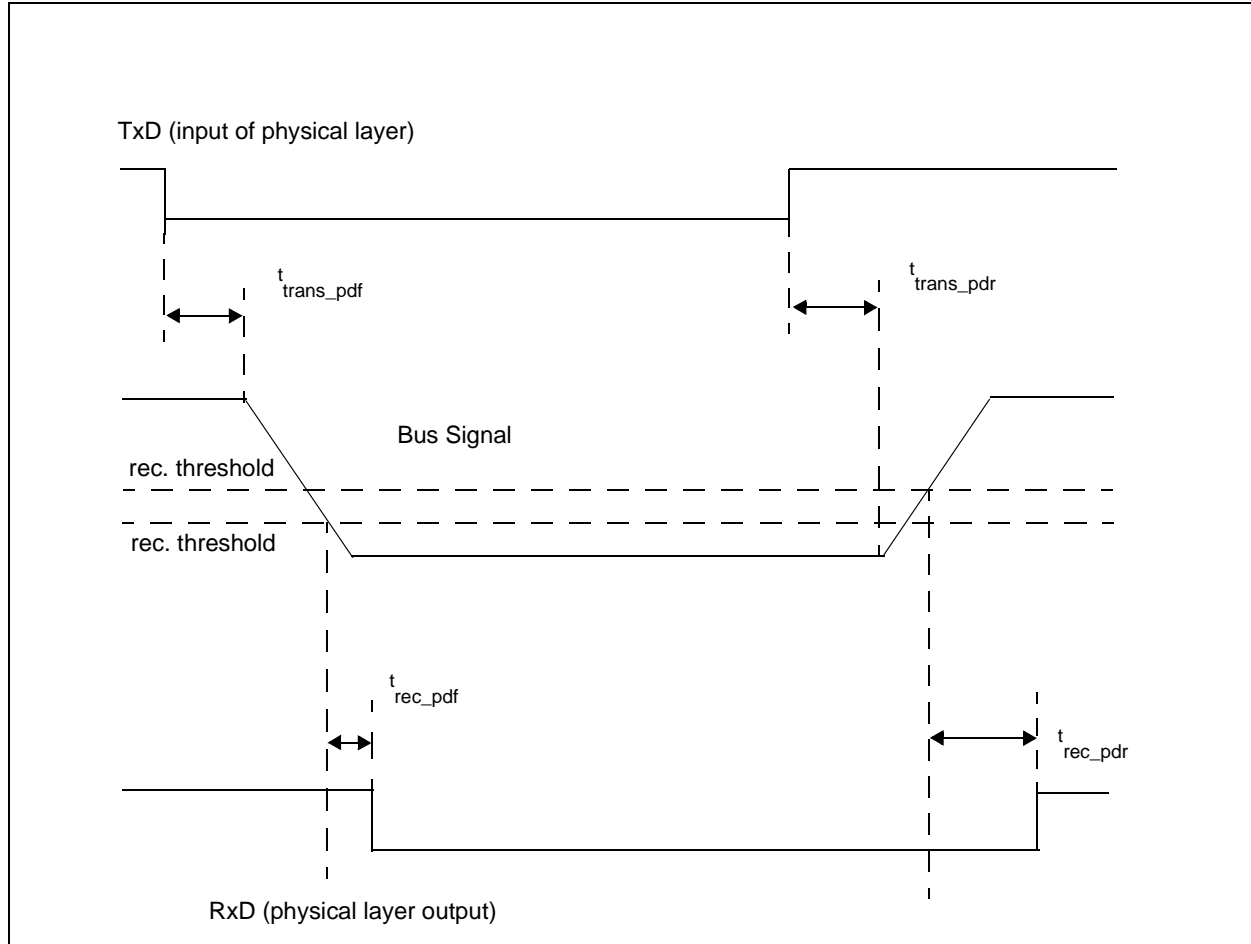
**Note 1:** Rising edge is system dependent. Value is characterized but not tested.

# PIC16C432

**TABLE 12-6: LIN THERMAL CHARACTERISTICS**

Symbol	Parameter	Typ.	Max.	Unit	Note
$\Theta_{\text{recovery}}$	Recovery Temperature	+135		°C	Information Parameter
$\Theta_{\text{shutdown}}$	Shutdown Temperature	+155		°C	Information Parameter
$T_{\text{THERM}}$	Thermal Recovery Time		1.5	ms	Information Parameter

**FIGURE 12-8: TIMING DIAGRAM**



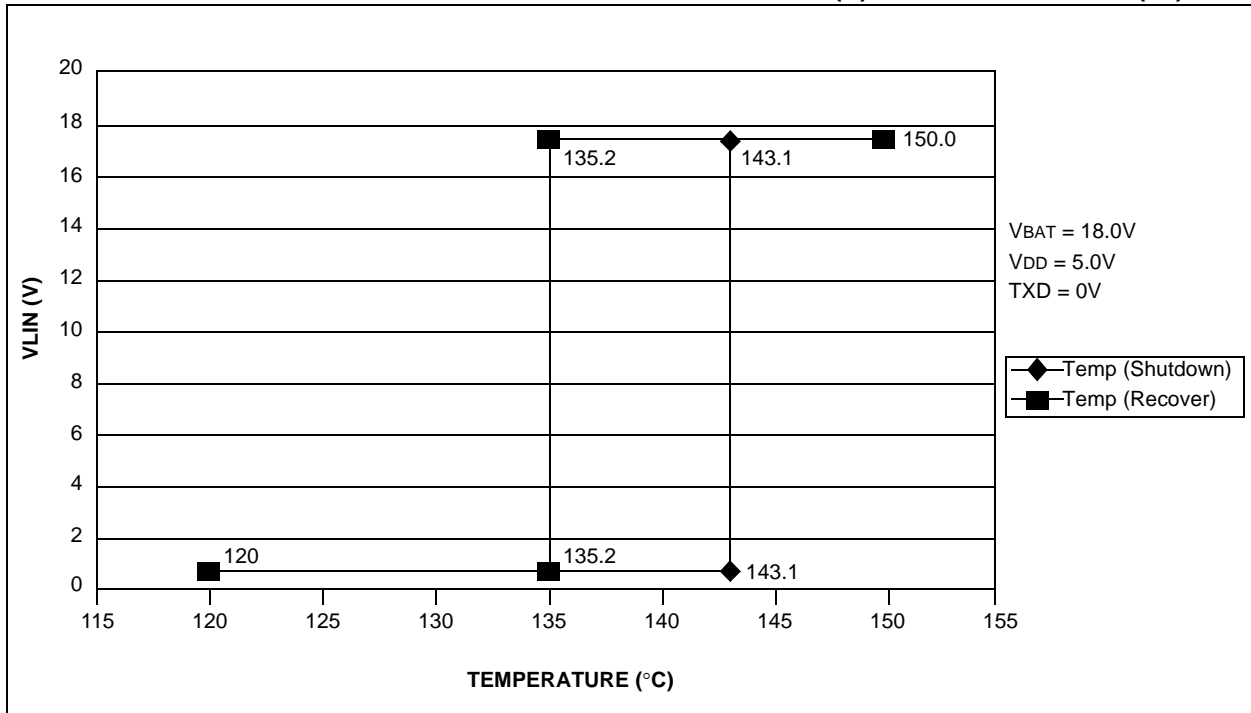
## 13.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs and tables, the data presented is **outside specified operating range** (i.e., outside specified VDD range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at 25°C. 'max' or 'min' represents (mean + 3σ) or (mean - 3σ) respectively, where σ is standard deviation, over the whole temperature range.

**FIGURE 13-1: LIN TRANSCEIVER SHUTDOWN HYSTERESIS (V) VS. TEMPERATURE (°C)**



# PIC16C432

---


---

NOTES:


## 14.0 PACKAGING INFORMATION

### 14.1 Package Marking Information


#### 20-Lead CERDIP Windowed

XXXXXXXXXXXXXXXXXX  
 XXXXXXXXXXXXXXXXXXXX  
 ○  YYWWNNN


#### Example

PIC16C432/P301  
 ○  0007CBP


#### 20-Lead SSOP

XXXXXXXXXXXX  
 XXXXXXXXXXXXXXX  
 ○  YYWWNNN


#### Example

PIC16C432  
 -I/218  
 ○  0007CBP

#### 20-Lead PDIP

XXXXXXXXXXXXXXXXXX  
 XXXXXXXXXXXXXXXXXXXX  
 ○  YYWWNNN

#### Example

PIC16C432/P301  
 ○  0007CBP

**Legend:** XX...X Customer-specific information  
 Y Year code (last digit of calendar year)  
 YY Year code (last 2 digits of calendar year)  
 WW Week code (week of January 1 is week '01')  
 NNN Alphanumeric traceability code  
 (e3) Pb-free JEDEC designator for Matte Tin (Sn)  
 \* This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# PIC16C432

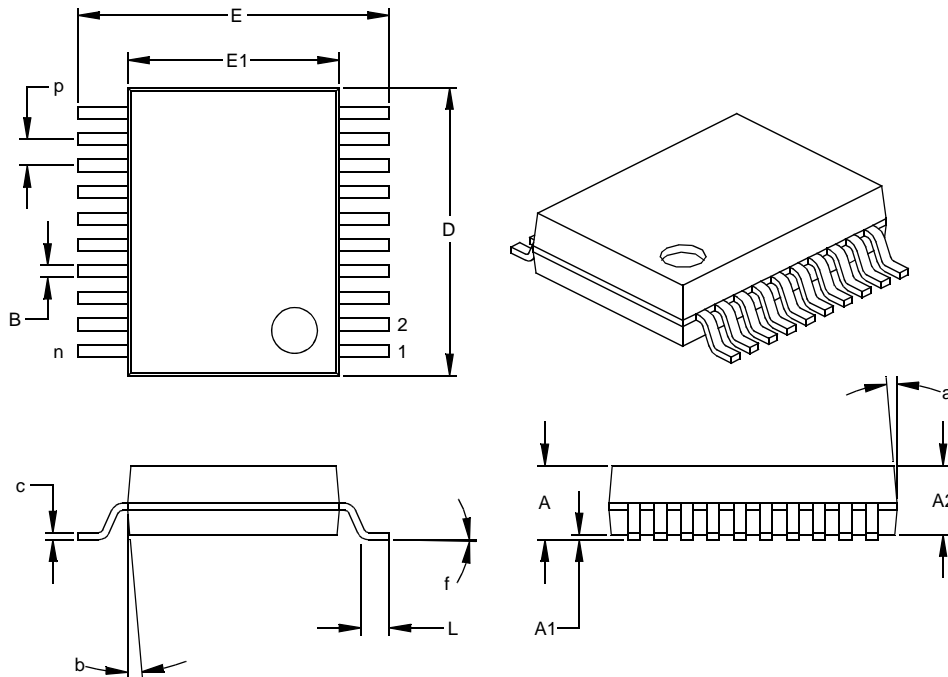
---

**20-Lead Ceramic Dual In-Line with Window (JW) - 300 mil (CERDIP)**

Package drawing not available at this time.

## 20-Lead Plastic Shrink Small Outline (SS) - 209 mil, 5.30 mm (SSOP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	P		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	f	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	a	0	5	10	0	5	10
Mold Draft Angle Bottom	b	0	5	10	0	5	10

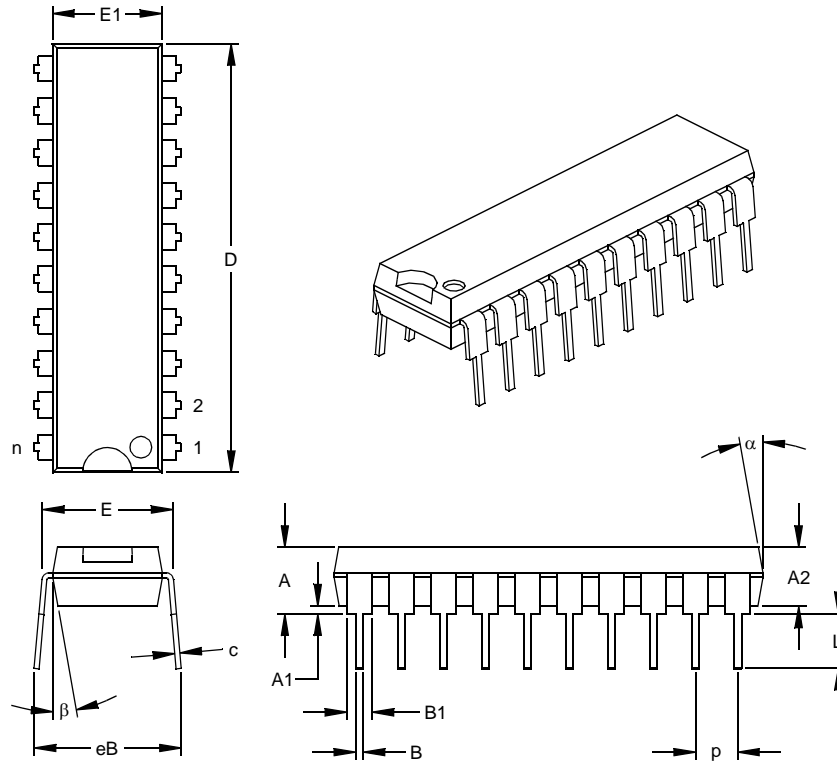
\* Controlling Parameter  
 § Significant Characteristic

Notes:  
 Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.  
 JEDEC Equivalent: MO-150  
 Drawing No. C04-072

# PIC16C432

## 20-Lead Plastic Dual In-Line (P) - 300 mil (PDIP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		INCHES*			MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.295	.310	.325	7.49	7.87	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	1.025	1.033	1.040	26.04	26.24	26.42
Tip to Seating Plane	L	.120	.130	.140	3.05	3.30	3.56
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.055	.060	.065	1.40	1.52	1.65
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

\* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-019



## **APPENDIX A: CODE FOR LIN COMMUNICATION**

Please check our web site at [www.microchip.com](http://www.microchip.com) for code availability.

## **APPENDIX B: REVISION HISTORY**

### **Revision C (January 2013)**

Added a note to each package outline drawing.

# PIC16C432

---

---

NOTES:

## INDEX

### A

ADDLW Instruction .....	62
ADDWF Instruction .....	62
ANDLW Instruction .....	62
ANDWF Instruction .....	62
Assembler	
MPASM Assembler .....	73

### B

BCF Instruction .....	63
Block Diagram	
TIMER0 .....	27
TMR0/WDT PRESCALER .....	30
Brown-out Detect (BOD) .....	48
BSF Instruction .....	63
BTFSC Instruction .....	63
BTFSS Instruction .....	64

### C

CALL Instruction .....	64
CLRF Instruction .....	64
CLRWF Instruction .....	65
CLRWDI Instruction .....	65
CMCON Register .....	33
Code Protection .....	58
COMF Instruction .....	65
Comparator Configuration .....	33
Comparator Interrupts .....	39
Comparator Module .....	33
Comparator Operation .....	35
Comparator Reference .....	35
Configuration Bits .....	43
Configuring the Voltage Reference .....	41
Crystal Operation .....	45

### D

Data Memory Organization .....	7
DECIF Instruction .....	65
DECFSZ Instruction .....	66
Development Support .....	73

### E

Errata .....	2
External Crystal Oscillator Circuit .....	46

### G

General Purpose Register File .....	7
GOTO Instruction .....	66

### I

I/O Ports .....	17
I/O Programming Considerations .....	22
ICEPIC In-Circuit Emulator .....	74
ID Locations .....	58
INCF Instruction .....	66
INCFSSZ Instruction .....	67
In-Circuit Serial Programming .....	58
Indirect Addressing, INDF and FSR Registers .....	16
Instruction Set	
ADDLW .....	62
ADDWF .....	62
ANDLW .....	62
ANDWF .....	62

BCF .....	63
BSF .....	63
BTFSC .....	63
BTFSS .....	64
CALL .....	64
CLRF .....	64
CLRWF .....	65
CLRWDI .....	65
COMF .....	65
DECIF .....	65
DECFSZ .....	66
GOTO .....	66
INCF .....	66
INCFSSZ .....	67
IORLW .....	67
IORWF .....	67
MOVF .....	68
MOVLW .....	68
MOVWF .....	68
NOP .....	68
OPTION .....	69
RETFIE .....	69
RETLW .....	69
RETURN .....	69
RLF .....	70
RRF .....	70
SLEEP .....	70
SUBLW .....	71
SUBWF .....	71
SWAPF .....	72
TRIS .....	72
XORLW .....	72
XORWF .....	72

Instruction Set Summary .....	59
INT Interrupt .....	54
INTCON Register .....	12
Interrupts .....	53
IORLW Instruction .....	67
IORWF Instruction .....	67

### K

KEELOQ Evaluation and Programming Tools .....	76
---	----

### L

LIN Hardware Interface .....	23
LIN Interfacing .....	23
LIN Protocol .....	23
LIN Transceiver .....	23

### M

MOVF Instruction .....	68
MOVLW Instruction .....	68
MOVWF Instruction .....	68
MPLAB C17 and MPLAB C18 C Compilers .....	73
MPLAB ICD In-Circuit Debugger .....	75
MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE .....	74
MPLAB Integrated Development Environment Software .....	73
MPLINK Object Linker/MPLIB Object Librarian .....	74

### N

NOP Instruction .....	68
-----------------------	----

### O

One-Time-Programmable (OTP) Devices .....	5
---	---

# PIC16C432

---

OPTION Instruction .....	69
OPTION Register .....	11
Oscillator Configurations .....	45
Oscillator Start-up Timer (OST) .....	48

## P

Package Marking Information .....	93
Packaging Information .....	93
PCL and PCLATH .....	15
PCON Register .....	14
PICDEM 1 Low Cost PIC MCU Demonstration Board .....	75
PICDEM 17 Demonstration Board .....	76
PICDEM 2 Low Cost PIC16CXX Demonstration Board .....	75
PICDEM 3 Low Cost PIC16CXXX Demonstration Board .....	76
PICSTART Plus Entry Level Development Programmer .....	75
PIE1 Register .....	13
PIR1 Register .....	13
Port RB Interrupt .....	54
PORTA .....	17
PORTB .....	20
Power Control/Status Register (PCON) .....	49
Power-down Mode (SLEEP) .....	57
Power-on Reset (POR) .....	48
Timeout (TO Bit) .....	10
Power-up Timer (PWRT) .....	48
Prescaler .....	30
Program Memory Organization .....	7

## Q

Quick-Turn-Programming (QTP) Devices .....	5
--	---

## R

RC Oscillator .....	46
RESET .....	47
RETFIE Instruction .....	69
RETLW Instruction .....	69
RETURN Instruction .....	69
RLF Instruction .....	70
RRF Instruction .....	70

## S

Serialized Quick-Turn-Programming (SQTP) Devices .....	5
SLEEP Instruction .....	70
Software Simulator (MPLAB SIM) .....	74
Special Features of the CPU .....	43
Special Function Registers .....	8
Stack .....	15
STATUS Register	
DC Bit .....	10
IRP Bit .....	10
TO Bit .....	10
Z Bit .....	10
Status Register .....	10
SUBLW Instruction .....	71
SUBWF Instruction .....	71
SWAPF Instruction .....	72

## T

Thermal Shutdown .....	23
Timer0	
TIMER0 .....	27
TIMER0 (TMR0) Interrupt .....	27
TIMER0 (TMR0) Module .....	27
TMR0 with External Clock .....	29
Timer1	

Switching Prescaler Assignment .....	31
Timing Diagrams and Specifications .....	86
TMR0 Interrupt .....	54
TRIS Instruction .....	72
TRISA .....	17
TRISB .....	20

## V

Voltage Reference Module .....	41
VRCON Register .....	41

## W

Watchdog Timer (WDT) .....	56
WWW, On-Line Support .....	2

## X

XORLW Instruction .....	72
XORWF Instruction .....	72

---

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com). Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://microchip.com/support>**

---

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent \_\_\_\_\_

RE: Reader Response

From: Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City / State / ZIP / Country \_\_\_\_\_

Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply?  Y  N

Device:

Literature Number: DS41140C

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this document easy to follow? If not, why?

---

---

4. What additions to the document do you think would enhance the structure and subject?

---

---

5. What deletions from the document could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---

## PRODUCT IDENTIFICATION SYSTEM

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

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
Device	PIC16C432: V <sub>DD</sub> range 4.0 V to 5.5 V PIC16C432T: V <sub>DD</sub> range 4.0 V to 5.5 V (Tape and Reel)		
Temperature Range	I = -40°C to +85°C E = -40°C to +125°C		
Package	SS = SSOP JW* = Windowed Cerdip		
Pattern	3-Digit Pattern Code for QTP (blank otherwise).		

**Examples:**

a) PIC16C432-E/P301 = Extra Temp, PDIP package, 4 MHz, normal V<sub>DD</sub> limits, QTP pattern #301

b) PIC16C432-I/SS Industrial Temp., SSOP package, 4 MHz, industrial V<sub>DD</sub> limits

\* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

## Sales and Support

### Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Worldwide Site ([www.microchip.com](http://www.microchip.com))





---

**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, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC, SST, SST Logo, SuperFlash 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, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale 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.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2000-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 9781620769720

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

*Microchip received ISO/TS-16949:2009 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<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> 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.*



# MICROCHIP

## 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](http://www.microchip.com)

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

**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**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

**Indianapolis**  
Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453

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

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario,  
Canada  
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: 852-2401-1200  
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 - Hangzhou**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**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

**China - Xiamen**  
Tel: 86-592-2388138  
Fax: 86-592-2388130

**China - Zhuhai**  
Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

**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: 91-20-2566-1512  
Fax: 91-20-2566-1513

**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  
Fax: 886-7-330-9305

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

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

11/29/12