



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



28/40-Pin, 8-Bit CMOS FLASH Microcontrollers

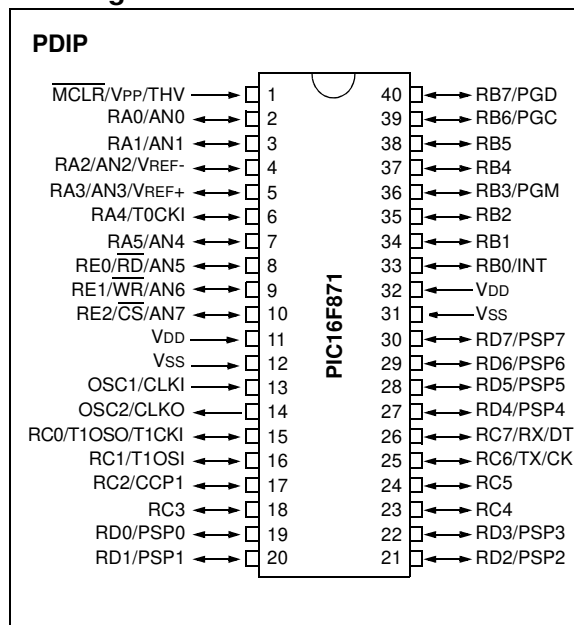
Devices Included in this Data Sheet:

- PIC16F870
- PIC16F871

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory
128 x 8 bytes of Data Memory (RAM)
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16CXXX 28 and 40-pin devices
- Interrupt capability (up to 11 sources)
- Eight level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low power consumption:
 - < 1.6 mA typical @ 5V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

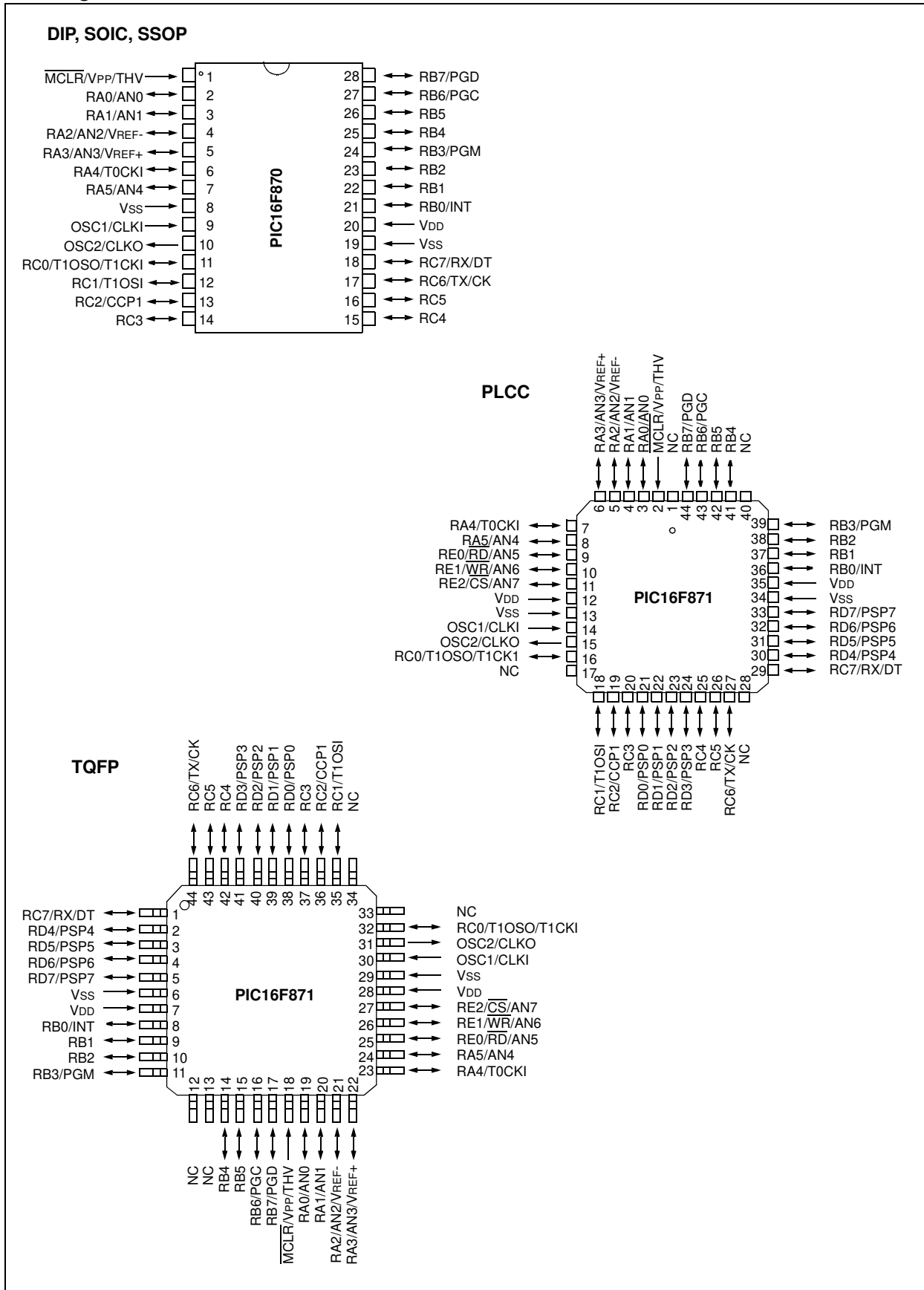


Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external \overline{RD} , \overline{WR} and \overline{CS} controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

PIC16F870/871

Pin Diagrams



PIC16F870/871

Key Features PICmicro™ Mid-Range MCU Family Reference Manual (DS33023)	PIC16F870	PIC16F871
Operating Frequency	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	2K	2K
Data Memory (bytes)	128	128
EEPROM Data Memory	64	64
Interrupts	10	11
I/O Ports	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3
Capture/Compare/PWM modules	1	1
Serial Communications	USART	USART
Parallel Communications	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions

PIC16F870/871

Table of Contents

1.0	Device Overview	5
2.0	Memory Organization	11
3.0	Data EEPROM and Flash Program Memory	27
4.0	I/O Ports	33
5.0	Timer0 Module	45
6.0	Timer1 Module	49
7.0	Timer2 Module	53
8.0	Capture/Compare/PWM Modules	55
9.0	Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART)	61
10.0	Analog-to-Digital (A/D) Converter Module	79
11.0	Special Features of the CPU	87
12.0	Instruction Set Summary	103
13.0	Development Support	111
14.0	Electrical Characteristics	117
15.0	DC and AC Characteristics Graphs and Tables	137
16.0	Packaging Information	149
	Appendix A: Revision History	157
	Appendix B: Device Differences	157
	Appendix C: Conversion Considerations	158
	Appendix D: Migration from Mid-Range to Enhanced Devices	158
	Appendix E: Migration from High-End to Enhanced Devices	159
	Index	161
	On-Line Support	167
	Systems Information and Upgrade Hot Line	167
	Reader Response	168
	PIC16F870/871 Product Identification System	169

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 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 to receive the most current information on all of our products.

PIC16F870/871

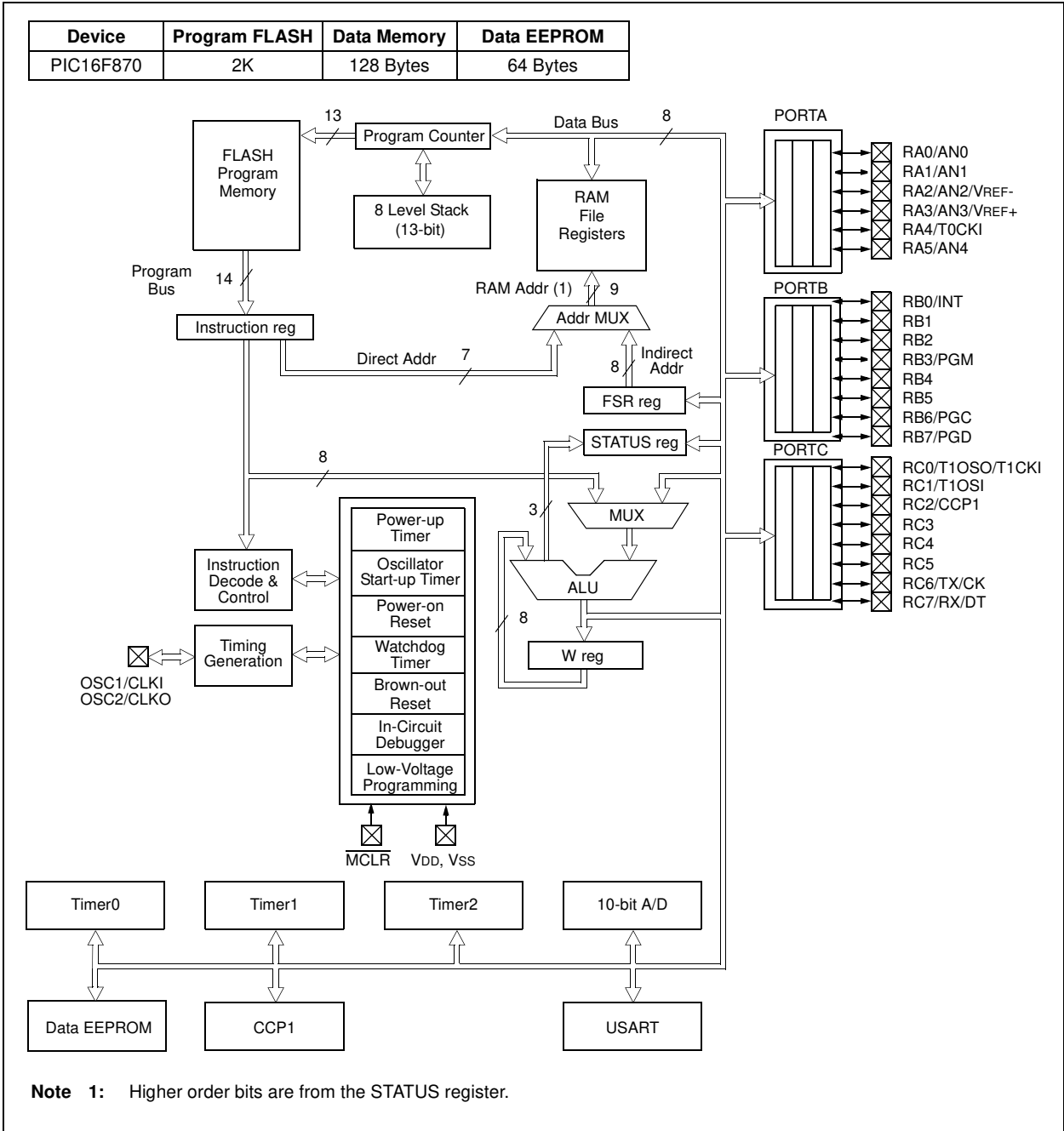
1.0 DEVICE OVERVIEW

This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are two devices (PIC16F870 and PIC16F871) covered by this data sheet. The PIC16F870 device comes in a 28-pin package and the PIC16F871 device comes in a 40-pin package. The 28-pin device does not have a Parallel Slave Port implemented.

The following two figures are device block diagrams sorted by pin number: 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F870 BLOCK DIAGRAM



PIC16F870/871

FIGURE 1-2: PIC16F871 BLOCK DIAGRAM

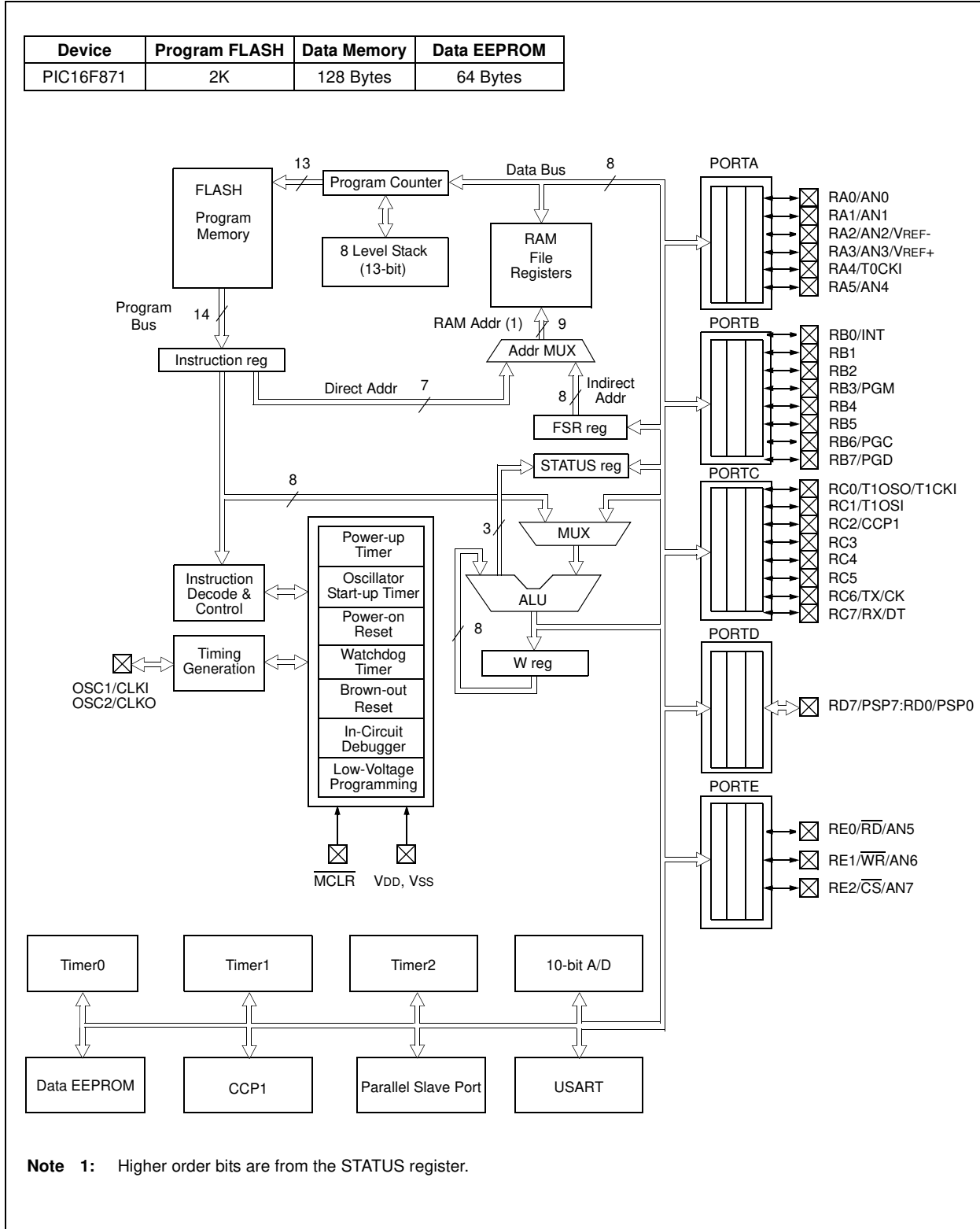


TABLE 1-1: PIC16F870 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKO	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, the OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input or High Voltage Test mode control. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input 0.</p> <p>RA1 can also be analog input 1.</p> <p>RA2 can also be analog input 2 or negative analog reference voltage.</p> <p>RA3 can also be analog input 3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 module. Output is open drain type.</p> <p>RA5 can also be analog input 4.</p>
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST/OD	
RA5/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL/ST ⁽¹⁾	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ⁽²⁾	
RB7/PGD	28	28	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p>
RC1/T1OSI	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3	14	14	I/O	ST	
RC4	15	15	I/O	ST	
RC5	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
 OD = Open Drain — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt or LVP mode.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F870/871

TABLE 1-2: PIC16F871 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	I	ST/CMOS ⁽⁴⁾	Oscillator crystal input/external clock source input.
OSC2/CLKO	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input or High Voltage Test mode control. This pin is an active low RESET to the device.
RA0/AN0	2	3	19	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input 0.</p> <p>RA1 can also be analog input 1.</p> <p>RA2 can also be analog input 2 or negative analog reference voltage.</p> <p>RA3 can also be analog input 3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input 4.</p>
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/VREF-	4	5	21	I/O	TTL	
RA3/AN3/VREF+	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL/ST ⁽¹⁾	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST ⁽²⁾	
RB7/PGD	40	44	17	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or a Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p>
RC1/T1OSI	16	18	35	I/O	ST	
RC2/CCP1	17	19	36	I/O	ST	
RC3	18	20	37	I/O	ST	
RC4	23	25	42	I/O	ST	
RC5	24	26	43	I/O	ST	
RC6/TX/CK	25	27	44	I/O	ST	
RC7/RX/DT	26	29	1	I/O	ST	

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as an external interrupt or LVP mode.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
Note 4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F871 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0	19	21	38	I/O	ST/TTL ⁽³⁾	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	23	40	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	24	41	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	30	2	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	31	3	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	32	4	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	33	5	I/O	ST/TTL ⁽³⁾	
RE0/ $\overline{\text{RD}}$ /AN5	8	9	25	I/O	ST/TTL ⁽³⁾	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input 5. RE1 can also be write control for the parallel slave port, or analog input 6. RE2 can also be select control for the parallel slave port, or analog input 7.
RE1/ $\overline{\text{WR}}$ /AN6	9	10	26	I/O	ST/TTL ⁽³⁾	
RE2/ $\overline{\text{CS}}$ /AN7	10	11	27	I/O	ST/TTL ⁽³⁾	
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
Vdd	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as an external interrupt or LVP mode.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F870/871

NOTES:

2.0 MEMORY ORGANIZATION

The PIC16F870/871 devices have three memory blocks. The Program Memory and Data Memory have separate buses, so that concurrent access can occur, and is detailed in this section. The EEPROM data memory block is detailed in Section 3.0.

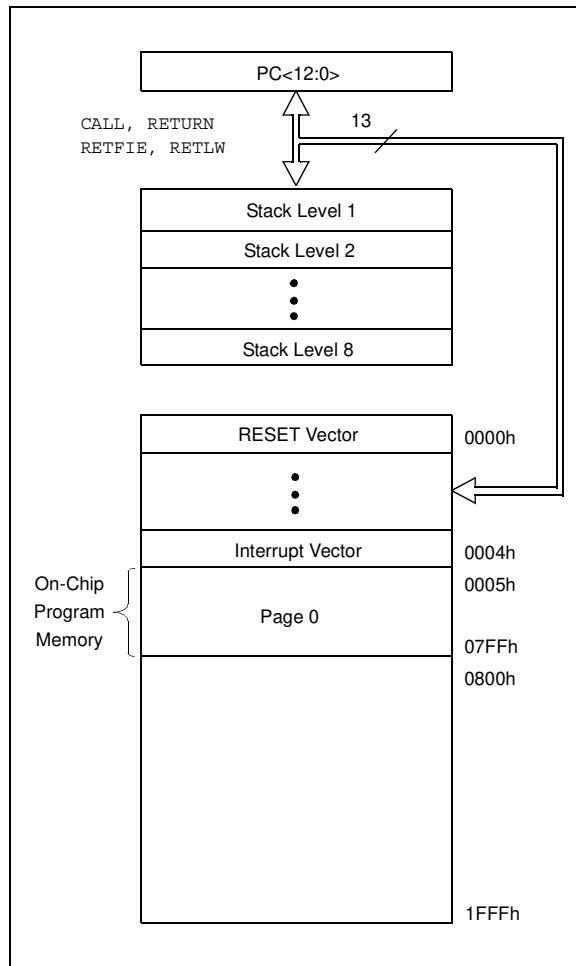
Additional information on device memory may be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

2.1 Program Memory Organization

The PIC16F870/871 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F870/871 devices have 2K x 14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F870/871 PROGRAM MEMORY MAP AND STACK



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP<1:0>	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some “high use” Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Note: EEPROM Data Memory description can be found in Section 3.0 of this Data Sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR.

PIC16F870/871

FIGURE 2-2: PIC16F870/871 REGISTER FILE MAP

File Address		File Address		File Address		File Address		
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h	
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h	
PCL	02h	PCL	82h	PCL	102h	PCL	182h	
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h	
FSR	04h	FSR	84h	FSR	104h	FSR	184h	
PORTA	05h	TRISA	85h		105h		185h	
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h	
PORTC	07h	TRISC	87h		107h		187h	
PORTD ⁽²⁾	08h	TRISD ⁽²⁾	88h		108h		188h	
PORTE ⁽²⁾	09h	TRISE ⁽²⁾	89h		109h		189h	
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch	
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh	
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽¹⁾	18Eh	
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽¹⁾	18Fh	
T1CON	10h		90h		110h		190h	
TMR2	11h		91h					
T2CON	12h	PR2	92h					
	13h		93h					
	14h		94h					
CCPR1L	15h		95h					
CCPR1H	16h		96h					
CCP1CON	17h		97h					
RCSTA	18h	TXSTA	98h					
TXREG	19h	SPBRG	99h					
RCREG	1Ah		9Ah					
	1Bh		9Bh					
	1Ch		9Ch					
	1Dh		9Dh					
ADRESH	1Eh	ADRESL	9Eh					
ADCON0	1Fh	ADCON1	9Fh					
General Purpose Register 96 Bytes	20h	General Purpose Register 32 Bytes	A0h	accesses 20h-7Fh	120h	accesses A0h - BFh	1A0h	
			BFh					1BFh
			C0h					1C0h
			EFh				accesses 70h-7Fh	16Fh
	F0h	170h	1F0h					
	7Fh		FFh	17Fh		1FFh		

Bank 0 Bank 1 Bank 2 Bank 3

■ Unimplemented data memory locations, read as '0'.
 * Not a physical register.

Note 1: These registers are reserved; maintain these registers clear.
Note 2: These registers are not implemented on the PIC16F870.

PIC16F870/871

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS ⁽²⁾
Bank 0											
00h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
02h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h ⁽⁴⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
04h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h ⁽⁵⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h ⁽⁵⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
0Dh	PIR2	—	—	—	EEIF	—	—	—	—	---0 ----	---0 ----
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	—	Unimplemented								—	—
14h	—	Unimplemented								—	—
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ \overline{DONE}	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through \overline{MCLR} and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

PIC16F870/871

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS ⁽²⁾	
Bank 1												
80h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000	
81h	OPTION_REG	$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
82h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000	
83h ⁽⁴⁾	STATUS	IRP	RP1	RP0	$\overline{\text{T}}\text{O}$	$\overline{\text{P}}\text{D}$	Z	DC	C	0001 1xxx	000q quuu	
84h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu	
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111	
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111	
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111	
88h ⁽⁵⁾	TRISD	PORTD Data Direction Register								1111 1111	1111 1111	
89h ⁽⁵⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits				0000 -111	0000 -111
8Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						---0 0000	---0 0000
8Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u	
8Ch	PIE1	PSPIE ⁽³⁾	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000	
8Dh	PIE2	—	—	—	EEIE	—	—	—	—	---0 ----	---0 ----	
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{P}}\text{OR}$	BOR	---- -gq	---- -uu	
8Fh	—	Unimplemented								—	—	
90h	—	Unimplemented								—	—	
91h	—	Unimplemented								—	—	
92h	PR2	Timer2 Period Register								1111 1111	1111 1111	
93h	—	Unimplemented								—	—	
94h	—	Unimplemented								—	—	
95h	—	Unimplemented								—	—	
96h	—	Unimplemented								—	—	
97h	—	Unimplemented								—	—	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010	
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000	
9Ah	—	Unimplemented								—	—	
9Bh	—	Unimplemented								—	—	
9Ch	—	Unimplemented								—	—	
9Dh	—	Unimplemented								—	—	
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu	
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000	

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through $\overline{\text{M}}\text{CLR}$ and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS ⁽²⁾
Bank 2											
100h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
102h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h ⁽⁴⁾	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM Data Register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM Address Register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM Data Register High Byte					xxxx xxxx	uuuu uuuu	
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	uuuu uuuu
Bank 3											
180h ⁽⁴⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h ⁽⁴⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h ⁽⁴⁾	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h ⁽⁴⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah ^(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh ⁽⁴⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

PIC16F870/871

2.2.2.1 STATUS Register

The STATUS register 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, as with 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 `000u u1uu` (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 the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits, see the "Instruction Set Summary".

Note 1: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS: 03h, 83h, 103h, 183h)

R/W-0	R/W-0	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
bit 7					bit 0		

- bit 7-6 **IRP:** Register Bank Select bit (used for indirect addressing)
 1 = Bank 2, 3 (100h - 1FFh)
 0 = Bank 0, 1 (00h - FFh)
- bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)
 11 = Bank 3 (180h - 1FFh)
 10 = Bank 2 (100h - 17Fh)
 11 = Bank 1 (80h - FFh)
 10 = Bank 0 (00h - 7Fh)
 Each bank is 128 bytes.
- bit 4 **\overline{TO} :** Time-out bit
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
 0 = A WDT time-out 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: 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	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

2.2.2.2 OPTION_REG Register

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2: OPTION_REG REGISTER (ADDRESS: 81h,181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	<u>RBPU</u>	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

- 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 (CLKO)
- 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 **PS2:PS0:** 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 '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

PIC16F870/871

2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

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 2-3: INTCON REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, 18Bh)

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
bit 7							bit 0

- 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 unmasked 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 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
 0 = None of the RB7:RB4 pins have changed state

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

2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

Note: Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1 REGISTER (ADDRESS: 8Ch)

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
bit 7								bit 0

- bit 7 **PSPIE⁽¹⁾:** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt
- bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt
- bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt
- bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt
- bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt
- bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Note 1: PSPIE is reserved on the PIC16F870; always maintain this bit clear.

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

PIC16F870/871

2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

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 bits are clear prior to enabling an interrupt.

REGISTER 2-5: PIR1 REGISTER (ADDRESS: 0Ch)

R/W-0	R/W-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7 **PSPIF⁽¹⁾:** Parallel Slave Port Read/Write Interrupt Flag bit
 1 = A read or a write operation has taken place (must be cleared in software)
 0 = No read or write has occurred
- bit 6 **ADIF:** A/D Converter Interrupt Flag bit
 1 = An A/D conversion completed
 0 = The A/D conversion is not complete
- bit 5 **RCIF:** USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 **TXIF:** USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
 Capture mode:
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
 Compare mode:
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
 PWM mode:
 Unused in this mode.
- bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
 1 = TMR2 to PR2 match occurred (must be cleared in software)
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
 1 = TMR1 register overflowed (must be cleared in software)
 0 = TMR1 register did not overflow

Note 1: PSPIF is reserved on the PIC16F870; always maintain this bit clear.

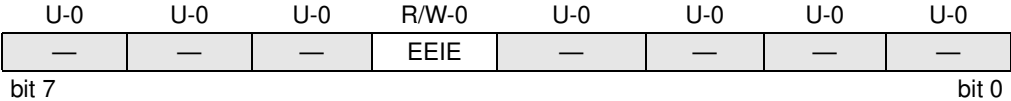
Legend:

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

2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bit for the EEPROM write operation interrupt.

REGISTER 2-6: PIE2 REGISTER (ADDRESS: 8Dh)



- bit 7-5 **Unimplemented:** Read as '0'
- bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable bit
1 = Enable EE write interrupt
0 = Disable EE write interrupt
- bit 3-0 **Unimplemented:** Read as '0'

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

PIC16F870/871

2.2.2.7 PIR2 Register

The PIR2 register contains the flag bit for the EEPROM write operation 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 2-7: PIR2 REGISTER (ADDRESS: 0Dh)

U-0	U-0	U-0	R/W-0	U-0	U-0	U-0	U-0
—	—	—	EEIF	—	—	—	—
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

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

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

2.2.2.8 PCON Register

The Power Control (PCON) register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset (BOR), a Watchdog Reset (WDT) and an external MCLR Reset.

Note: $\overline{\text{BOR}}$ is unknown on POR. It must be set by the user and checked on subsequent RESETS to see if BOR is clear, indicating a brown-out has occurred. The BOR status bit is a don't care and is not predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the configuration word).

REGISTER 2-8: PCON REGISTER (ADDRESS: 8Eh)

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

- 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 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
 1 = No Brown-out Reset occurred
 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend:

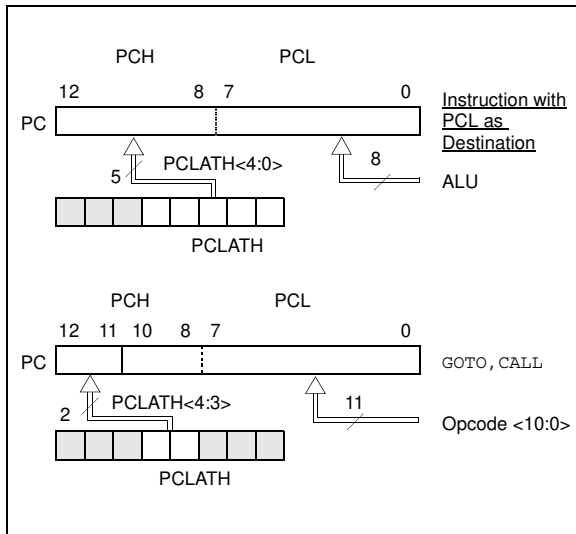
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F870/871

2.3 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 upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any RESET, the upper bits of the PC will be cleared. Figure 2-3 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> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 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).

2.3.2 STACK

The PIC16FXXX family has an 8-level deep x 13-bit wide hardware stack. 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 instructions/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.

2.4 Program Memory Paging

The PIC16FXXX architecture is capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide 11 bits of the address, which allows branches within any 2K program memory page. Therefore, the 8K words of program memory are broken into four pages. Since the PIC16F872 has only 2K words of program memory or one page, additional code is not required to ensure that the correct page is selected before a CALL or GOTO instruction is executed. The PCLATH<4:3> bits should always be maintained as zeros. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Manipulation of the PCLATH is not required for the return instructions.

2.5 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 the register pointed to by the File Select register, FSR. Reading the INDF register itself indirectly (FSR = 0) will read 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 2-4.

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

EXAMPLE 2-1: INDIRECT ADDRESSING

```

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

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING

