Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

PIC16F716
Data Sheet

8-bit Flash-based Microcontroller with A/D Converter and Enhanced Capture/Compare/PWM

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

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

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

Printed on recycled paper.

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

MICROCHIP

PIC16F716

8-bit Flash-based Microcontroller with A/D Controller and Enhanced Capture/Compare PWM

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
- Interrupt capability
  (up to 7 internal/external interrupt sources)
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- 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
- Dual level Brown-out Reset circuitry
  - 2.5 V_{BOR} (Typical)
  - 4.0 V_{BOR} (Typical)
- Programmable code protection
- Power-Saving Sleep mode
- Selectable oscillator options
- Fully static design
- In-Circuit Serial Programming™ (ICSP™)

CMOS Technology:

- Wide operating voltage range:
  - Industrial: 2.0V to 5.5V
  - Extended: 3.0V to 5.5V
- High Sink/Source Current 25/25 mA
- Wide temperature range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C

Low-Power Features:

- Standby Current:
  - 100 nA @ 2.0V, typical
- Operating Current:
  - 14 µA @ 32 kHz, 2.0V, typical
  - 120 µA @ 1 MHz, 2.0V, typical
- Watchdog Timer Circuit:
  - 1 µA @ 2.0V, typical
- Timer1 Oscillator Current:
  - 3.0 µA @ 32 kHz, 2.0V, typical

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
- Enhanced Capture, Compare, PWM module:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM maximum resolution is 10-bit
  - Enhanced PWM:
    - Single, Half-Bridge and Full-Bridge modes
    - Digitally programmable dead-band delay
    - Auto-shutdown/restart
- 8-bit multi-channel Analog-to-Digital Converter
- 13 I/O pins with individual direction control
- Programmable weak pull-ups on PORTB

Device	Memory		I/O	8-bit A/D (ch)	Timers 8/16	PWM (outputs)	V_{DD} Range
	Flash	Data					
PIC16F716	2048 x 14	128 x 8	13	4	2/1	1/2/4	2.0V-5.5V

18-Pin Diagram

18-pin PDIP, SOIC

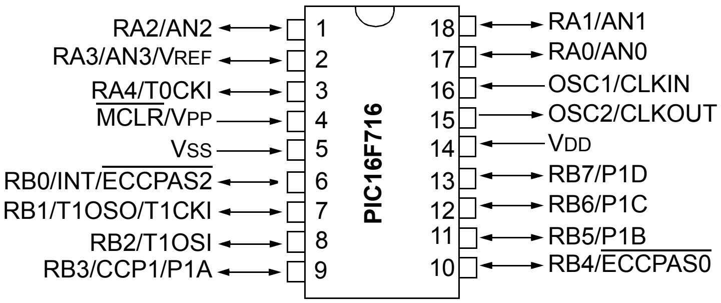


TABLE 1: 18-PIN PDIP, SOIC SUMMARY

I/O	Pin	Analog	ECCP	Timer	Interrupts	Pull-ups	Basic
RA0	17	AN0	—	—	—	—	—
RA1	18	AN1	—	—	—	—	—
RA2	1	AN2	—	—	—	—	—
RA3	2	AN3/VREF	—	—	—	—	—
RA4	3	—	—	T0CKI	—	—	—
RB0	6	—	$\overline{\text{ECCPAS2}}$	—	INT	Y	—
RB1	7	—	—	T1CKI	—	Y	—
RB2	8	—	—	T1OSI	—	Y	—
RB3	9	—	CCP1/P1A	—	—	Y	—
RB4	10	—	$\overline{\text{ECCPAS0}}$	—	IOC	Y	—
RB5	11	—	P1B	—	IOC	Y	—
RB6	12	—	P1C	—	IOC	Y	ICSPCLK
RB7	13	—	P1D	—	IOC	Y	ICSPDAT
—	14	—	—	—	—	—	VDD
—	5	—	—	—	—	—	VSS
—	4	—	—	—	—	—	$\overline{\text{MCLR}}$/VPP
—	16	—	—	—	—	—	OSC1/CLKIN
—	15	—	—	—	—	—	OSC2/CLKOUT

20-Pin Diagram

20-pin SSOP

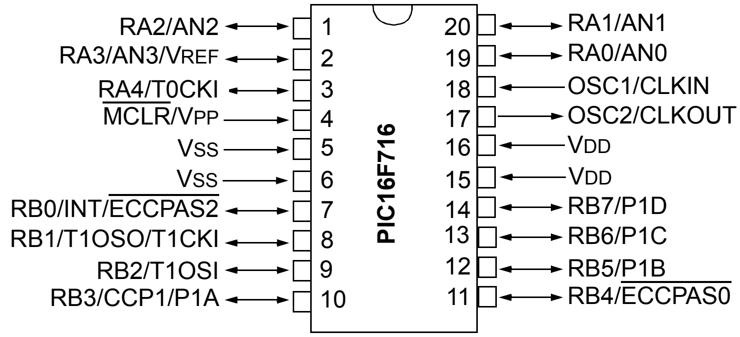


TABLE 2: 20-PIN SSOP SUMMARY

I/O	Pin	Analog	ECCP	Timer	Interrupts	Pull-ups	Basic
RA0	19	AN0	—	—	—	—	—
RA1	20	AN1	—	—	—	—	—
RA2	1	AN2	—	—	—	—	—
RA3	2	AN3/VREF	—	—	—	—	—
RA4	3	—	—	T0CKI	—	—	—
RB0	7	—	$\overline{\text{ECCPAS2}}$	—	INT	Y	—
RB1	8	—	—	T1CKI	—	Y	—
RB2	9	—	—	T1OSI	—	Y	—
RB3	10	—	CCP1/P1A	—	—	Y	—
RB4	11	—	$\overline{\text{ECCPAS0}}$	—	IOC	Y	—
RB5	12	—	P1B	—	IOC	Y	—
RB6	13	—	P1C	—	IOC	Y	ICSPCLK
RB7	14	—	P1D	—	IOC	Y	ICSPDAT
—	15	—	—	—	—	—	VDD
—	16	—	—	—	—	—	VDD
—	5	—	—	—	—	—	VSS
—	6	—	—	—	—	—	VSS
—	4	—	—	—	—	—	$\overline{\text{MCLR}}$/VPP
—	18	—	—	—	—	—	OSC1/CLKIN
—	17	—	—	—	—	—	OSC2/CLKOUT

Table of Contents

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@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)

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

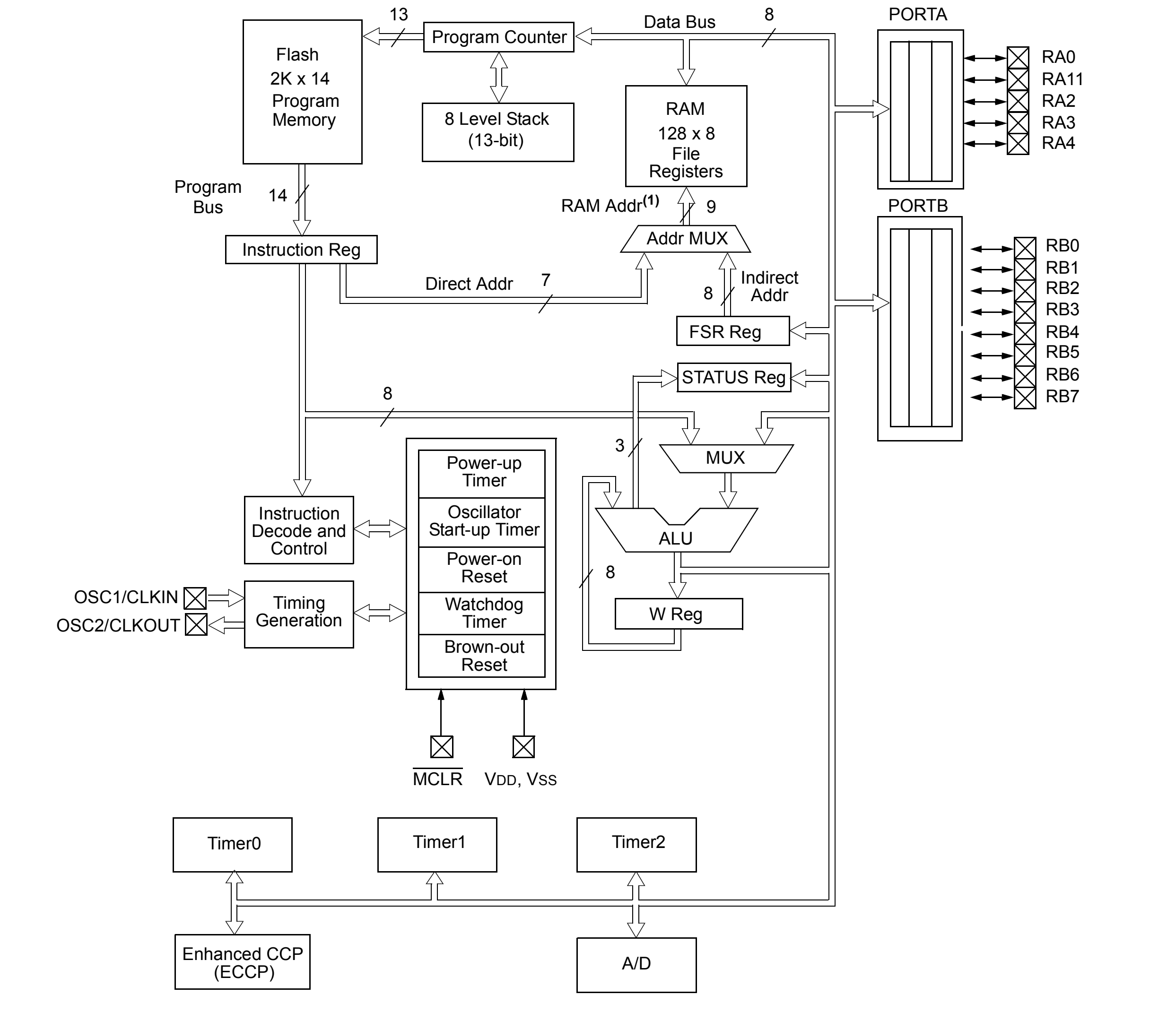
Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC16F716. Figure 1-1 is the block diagram for the PIC16F716 device. The pinouts are listed in Table 1-1.

FIGURE 1-1: PIC16F716 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

TABLE 1-1: PIC16F716 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
$\overline{\text{MCLR}}$/VPP	$\overline{\text{MCLR}}$	ST	—	Master clear (Reset) input. This pin is an active-low Reset to the device.
	VPP	P	—	Programming voltage input
OSC1/CLKIN	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	CMOS	—	External clock source input
	CLKIN	ST	—	RC Oscillator mode
OSC2/CLKOUT	OSC2	XTAL	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
RA0/AN0	RA0	TTL	CMOS	Bidirectional I/O
	AN0	AN	—	Analog Channel 0 input
RA1/AN1	RA1	TTL	CMOS	Bidirectional I/O
	AN1	AN	—	Analog Channel 1 input
RA2/AN2	RA2	TTL	CMOS	Bidirectional I/O
	AN2	AN	—	Analog Channel 2 input
RA3/AN3/VREF	RA3	TTL	CMOS	Bidirectional I/O
	AN3	AN	—	Analog Channel 3 input
	VREF	AN	—	A/D reference voltage input
RA4/T0CKI	RA4	ST	OD	Bidirectional I/O. Open drain when configured as output.
	T0CKI	ST	—	Timer0 external clock input
RB0/INT/$\overline{\text{ECCPAS2}}$	RB0	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up.
	INT	ST	—	External Interrupt
	$\overline{\text{ECCPAS2}}$	ST	—	ECCP Auto-Shutdown pin
RB1/T1OSO/T1CKI	RB1	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up.
	T1OSO	—	XTAL	Timer1 oscillator output. Connects to crystal in Oscillator mode.
	T1CKI	ST	—	Timer1 external clock input
RB2/T1OSI	RB2	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up.
	T1OSI	XTAL	—	Timer1 oscillator input. Connects to crystal in Oscillator mode.
RB3/CCP1/P1A	RB3	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up.
	CCP1	ST	CMOS	Capture1 input, Compare1 output, PWM1 output.
	P1A	—	CMOS	PWM P1A output
RB4/$\overline{\text{ECCPAS0}}$	RB4	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up. Interrupt-on-change.
	$\overline{\text{ECCPAS0}}$	ST	—	ECCP Auto-Shutdown pin
RB5/P1B	RB5	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up. Interrupt-on-change.
	P1B	—	CMOS	PWM P1B output
RB6/P1C	RB6	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up. Interrupt-on-change. ST input when used as ICSP programming clock.
	P1C	—	CMOS	PWM P1C output
RB7/P1D	RB7	TTL	CMOS	Bidirectional I/O. Programmable weak pull-up. Interrupt-on-change. ST input when used as ICSP programming data.
	P1D	—	CMOS	PWM P1D output
VSS	VSS	P	—	Ground reference for logic and I/O pins.
VDD	VDD	P	—	Positive supply for logic and I/O pins.

Legend:	I	= Input	AN	= Analog input or output	OD	= Open drain
	O	= Output	TTL	= TTL compatible input	ST	= Schmitt Trigger input with CMOS levels
	P	= Power	XTAL	= Crystal	CMOS	= CMOS compatible input or output

2.0 MEMORY ORGANIZATION

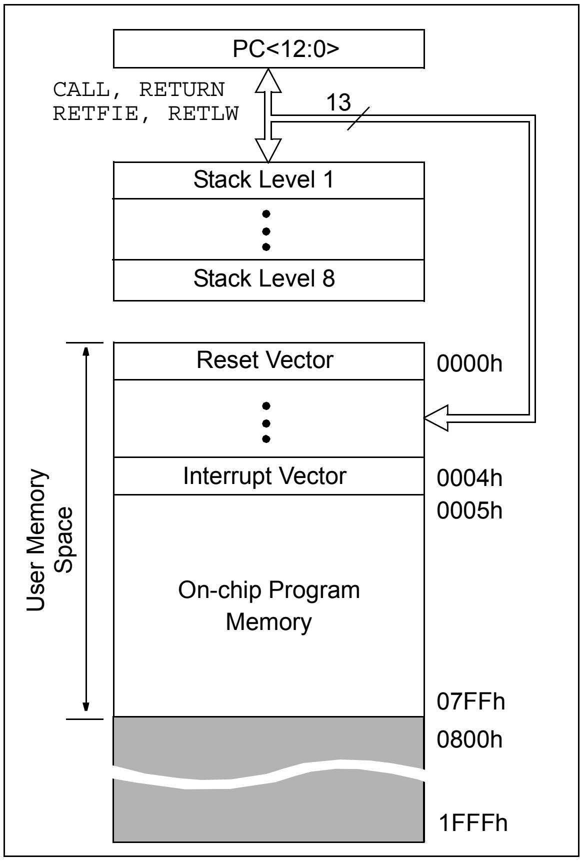
There are two memory blocks in the PIC16F716 device. Each block (program memory and data memory) has its own bus so that concurrent access can occur.

2.1 Program Memory Organization

The PIC16F716 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F716 has 2K x 14 words of program memory. Accessing a location above the physically implemented address will cause a wrap-around.

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

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK OF PIC16F716



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bits RP1 and RP0 of the STATUS register are the bank select bits.

RP<1:0>(1) **(Status<6:5>)**	**Bank**
00	0
01	1
10	2(2)
11	3(2)

Note 1: Maintain Status bit 6 clear to ensure upward compatibility with future products.

2: Not implemented

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. The upper 16 bytes of GPR space and some “high use” Special Function Registers in Bank 0 are mirrored in Bank 1 for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Select Register FSR (**Section 2.5 “Indirect Addressing, INDF and FSR Registers”**).

FIGURE 2-2: REGISTER FILE MAP

File Address			File Address
00h	INDF[(1)]	INDF[(1)]	80h
01h	TMR0	OPTION_REG	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	TMR1L	PCON	8Eh
0Fh	TMR1H	▨	8Fh
10h	T1CON	▨	90h
11h	TMR2	▨	91h
12h	T2CON	PR2	92h
13h	▨	▨	93h
14h	▨	▨	94h
15h	CCPR1L	▨	95h
16h	CCPR1H	▨	96h
17h	CCP1CON	▨	97h
18h	PWM1CON	▨	98h
19h	ECCPAS	▨	99h
1Ah	▨	▨	9Ah
1Bh	▨	▨	9Bh
1Ch	▨	▨	9Ch
1Dh	▨	▨	9Dh
1Eh	ADRES	▨	9Eh
1Fh	ADCON0	ADCON1	9Fh
20h	General Purpose Registers 80 Bytes	General Purpose Registers 32 Bytes	A0h BFh
6Fh		▨	C0h EFh
70h 7Fh	16 Bytes	Accesses 70-7Fh	F0h FFh
	Bank 0	Bank 1	

▨ Unimplemented data memory locations, read as ‘0’.

Note 1: Not a physical register.

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 give 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 that peripheral feature section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY BANK 0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
00h	INDF[1]	Addressing this location uses contents of FSR to address data memory (not a physical register)								`0000 0000`	18
01h	TMR0	Timer0 module's register								`xxxx xxxx`	27
02h	PCL[1]	Program Counter's (PC) Least Significant Byte								`0000 0000`	17
03h	STATUS[1]	IRP[4]	RP1[4]	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`0001 1xxx`	11
04h	FSR[1]	Indirect Data Memory Address Pointer								`xxxx xxxx`	18
05h	PORTA[5,6]	—	—	—[7]	RA4	RA3	RA2	RA1	RA0	`---x 0000`	19
06h	PORTB[5,6]	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	21
07h-09h	—	Unimplemented								—	
0Ah	PCLATH[1,2]	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					`---0 0000`	17
0Bh	INTCON[1]	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	13
0Ch	PIR1	—	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	`-0-- -000`	15
0Dh	—	Unimplemented								—	
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								`xxxx xxxx`	29
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								`xxxx xxxx`	29
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	`--00 0000`	32
11h	TMR2	Timer2 Module's Register								`0000 0000`	35
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	`-000 0000`	36
13h-14h	—	Unimplemented								—	
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								`xxxx xxxx`	48
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								`xxxx xxxx`	48
17h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	`0000 0000`	48
18h	PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	`0000 0000`	60
19h	ECCPAS	ECCPASE	ECCPAS2	—[8]	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	`00-0 0000`	57
1Ah-1Dh	—	Unimplemented								—	
1Eh	ADRES	A/D Result Register								`xxxx xxxx`	37
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—[7]	ADON	`0000 0000`	41

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

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: Other (non Power-up) Resets include: external Reset through $\overline{\text{MCLR}}$ and the Watchdog Timer Reset.

4: The IRP and RP1 bits are reserved. Always maintain these bits clear.

5: On any device Reset, these pins are configured as inputs.

6: This is the value that will be in the PORT output latch.

7: Reserved bits, do not use.

8: ECCPAS1 bit is not used on PIC16F716.

TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY BANK 1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
80h	INDF(1)	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	18
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	12
82h	PCL(1)	Program Counter's (PC) Least Significant Byte								0000 0000	17
83h	STATUS(1)	IRP(4)	RP1(4)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	11
84h	FSR(1)	Indirect Data Memory Address Pointer								xxxx xxxx	18
85h	TRISA	—	—	—(7)	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	19
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	21
87h-89h	—	Unimplemented								—	
8Ah	PCLATH(1,2)	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	17
8Bh	INTCON(1)	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	13
8Ch	PIE1	—	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	-0-- -000	14
8Dh	—	Unimplemented								—	
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	---- --qq	16
8Fh-91h	—	Unimplemented								—	
92h	PR2	Timer2 Period Register								1111 1111	35, 52
93h-9Eh	—	Unimplemented								—	
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	42

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

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: Other (non Power-up) Resets include: external Reset through $\overline{\text{MCLR}}$ and the Watchdog Timer Reset.

4: The IRP and RP1 bits are reserved. Always maintain these bits clear.

5: On any device Reset, these pins are configured as inputs.

6: This is the value that will be in the PORT output latch.

7: Reserved bits, do not use.

2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, 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, 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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ 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 PIC16F716 does not use bits IRP and RP1 of the STATUS register. Maintain these bits clear to ensure upward compatibility with future products.

2: The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ bit, respectively, in subtraction.

REGISTER 2-1: STATUS: STATUS REGISTER

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7							bit 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

bit 7 **IRP:** This bit is reserved and should be maintained as '`0`'

bit 6 **RP1:** This bit is reserved and should be maintained as '`0`'

bit 5 **RP0:** Register Bank Select bit (used for direct addressing)
`1` = Bank 1 (80h-FFh)
`0` = Bank 0 (00h-7Fh)

bit 4 **$\overline{\text{TO}}$:** Time-out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$:** 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/$\overline{\text{Borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions), For $\overline{\text{Borrow}}$, 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/$\overline{\text{Borrow}}$ bit[(1)] (`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 $\overline{\text{Borrow}}$, 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-order or low-order bit of the source register.

2.2.2.2 OPTION Register

The OPTION 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 Timer0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 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

bit 7 **$\overline{\text{RBPU}}$:** 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:** Timer0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** Timer0 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	Timer0 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

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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

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	T0IE	INTE	RBIE[1]	T0IF[2]	INTF	RBIF
bit 7							bit 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

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 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 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:** PORTB Change Interrupt Enable bit[1]
1 = Enables the PORTB change interrupt
0 = Disables the PORTB change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit[2]
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:** PORTB Change Interrupt Flag bit
1 = When at least one of the PORTB general purpose I/O pins changed state (must be cleared in software)
0 = None of the PORTB general purpose I/O pins have changed state

Note 1: IOCB register must also be enabled.

2: T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

2.2.2.4 PIE1 Register

This register contains the individual enable bits for the peripheral interrupts.

Note:	Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 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

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5-3 **Unimplemented:** Read as ‘0’

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** Timer2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

2.2.2.5 PIR1 Register

This 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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
—	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 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

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **ADIF:** A/D Interrupt Flag bit
1 = A/D conversion complete
0 = A/D conversion has not completed or has not been started

bit 5-3 **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:** Timer2 to PR2 Match Interrupt Flag bit
1 = Timer2 to PR2 match occurred (must be cleared in software)
0 = Timer2 to PR2 match has not occurred

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = Timer1 register overflowed (must be cleared in software)
0 = Timer1 has not overflowed

2.2.2.6 PCON Register

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR) to an external $\overline{\text{MCLR}}$ Reset or WDT Reset. These devices contain an additional bit to differentiate a Brown-out Reset condition from a Power-on Reset condition.

Note: If the BOREN Configuration bit is set, $\overline{\text{BOR}}$ is '1' on Power-on Reset and reset to '0' when a Brown-out condition occurs. $\overline{\text{BOR}}$ must then be set by the user and checked on subsequent Resets to see if it is clear, indicating that another Brown-out has occurred.

If the BOREN Configuration bit is clear, $\overline{\text{BOR}}$ is unknown on Power-on Reset.

REGISTER 2-6: PCON: POWER CONTROL REGISTER

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

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

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

1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

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

1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

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 high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 5 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

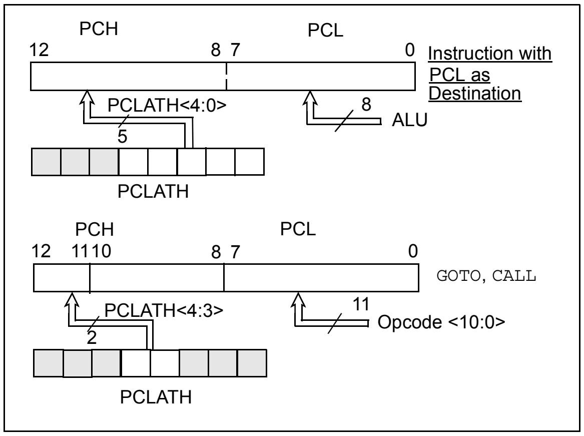
A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, "*Implementing a Table Read*" (DS00556).

2.3.2 PROGRAM MEMORY PAGING

The CALL and GOTO instructions provide 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper bit of the address is provided by PCLATH<3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bit is programmed so that the desired program memory page is addressed. If a RETURN from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> bit is not required for the RETURN instructions (which POPs the address from the stack).

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.4 Stack

The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Mid-range devices have 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 modified when the stack is PUSHed or POPed.

After the stack has been PUSHed 8 times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

EXAMPLE 2-1: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

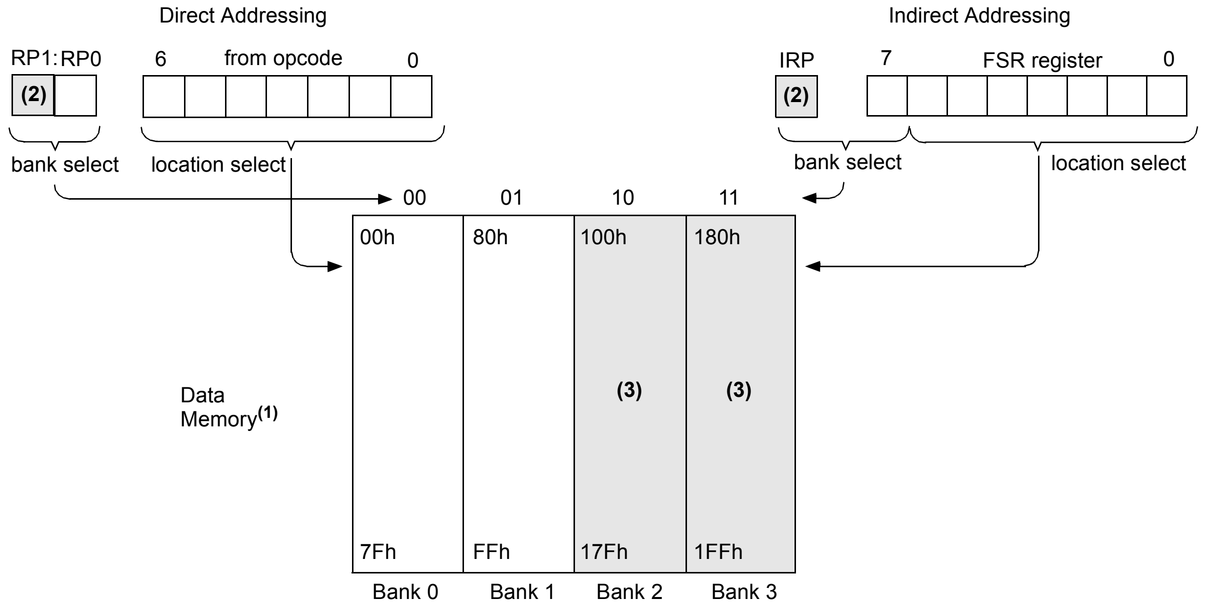
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```
          MOVLW   0x20    ;initialize pointer
          MOVWF   FSR     ;to RAM
NEXT      CLRF    INDF    ;clear RAM & FSR
          INCF    FSR     ;inc pointer
          BTFSS   FSR,4   ;all done?
          GOTO    NEXT    ;no, clear next
CONTINUE
          :               ;yes, continue
```

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit of the STATUS register, as shown in Figure 2-4. However, IRP is not used in the PIC16F716.

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING



Note 1: For register file map detail see Figure 2-2.
2: Maintain clear for upward compatibility with future products.
3: Not implemented.

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate 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.

3.1 PORTA and the TRISA Register

PORTA is a 5-bit wide bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= `1`) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISA bit (= `0`) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

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

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

PORTA pins, RA<3:0>, are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as ‘0’.

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

Note: Setting RA3:0 to output while in Analog mode will force pins to output contents of data latch.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF     STATUS, RP0    ;
CLRF    PORTA          ;Initialize PORTA by
                       ;clearing output
                       ;data latches
BSF     STATUS, RP0    ;Select Bank 1
MOVLW   0xEF           ;Value used to
                       ;initialize data
                       ;direction
MOVWF   TRISA          ;Set RA<3:0> as inputs
                       ;RA<4> as outputs
BCF     STATUS, RP0    ;Return to Bank 0
```

FIGURE 3-1: BLOCK DIAGRAM OF RA<3:0>

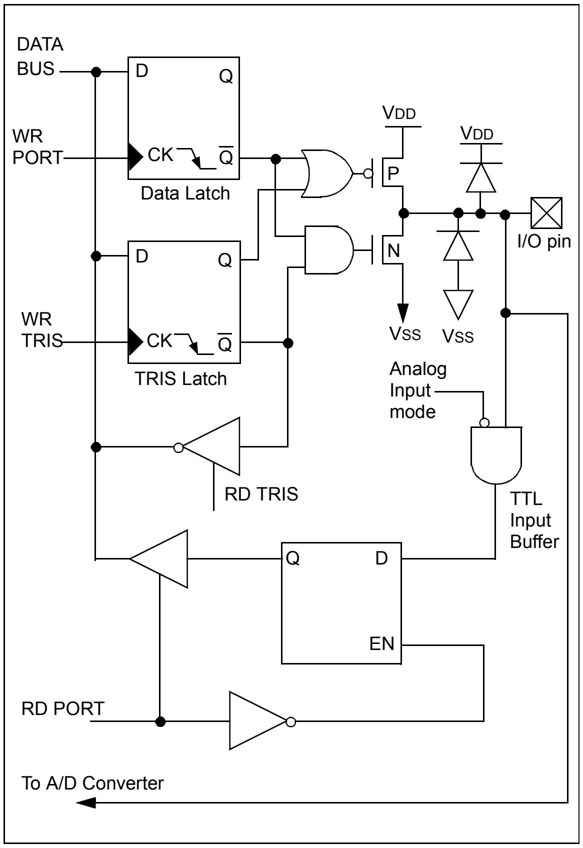


FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN

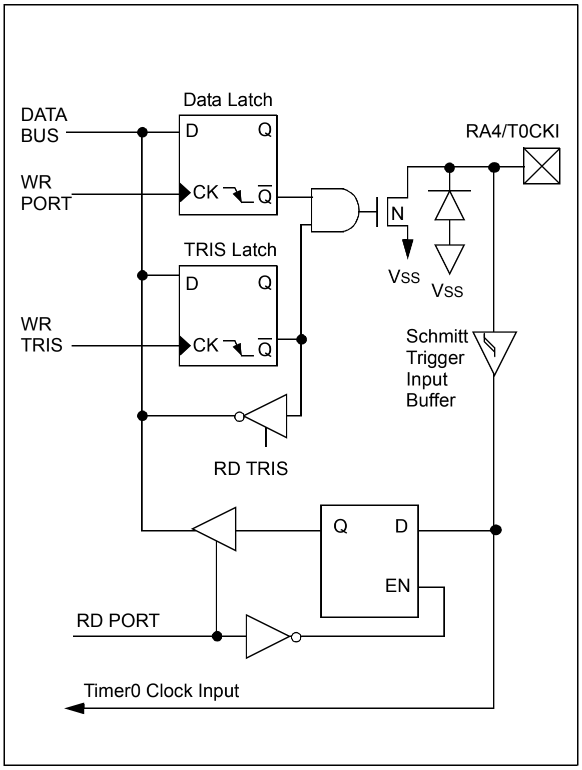


TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	`---x 0000`	`---u uuuu`
TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`
ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	`---- -000`	`---- -000`

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

3.2 PORTB and the TRISB Register

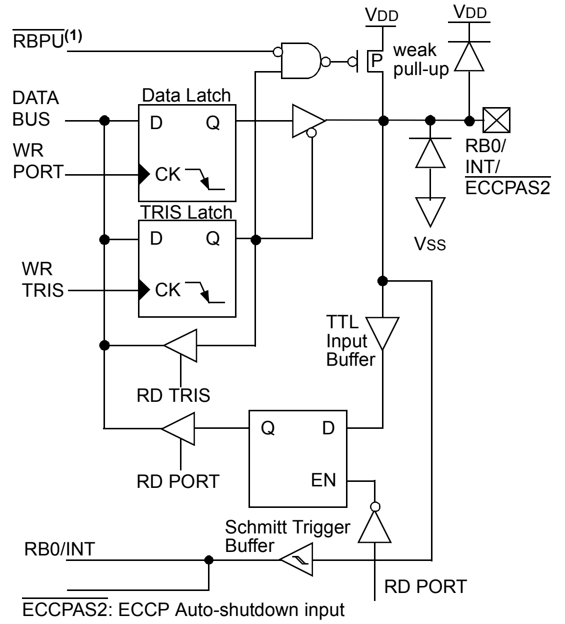
PORTB is an 8-bit wide bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= `1`) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= `0`) will make the corresponding PORTB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

EXAMPLE 3-2: INITIALIZING PORTB

```
BCF     STATUS, RP0    ;select Bank 0
CLRF    PORTB          ;Initialize PORTB by
                       ;clearing output
                       ;data latches
BSF     STATUS, RP0    ;Select Bank 1
MOVLW   0xCF           ;Value used to
                       ;initialize data
                       ;direction
MOVWF   TRISB          ;Set RB<3:0> as inputs
                       ;RB<5:4> as outputs
                       ;RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU of the OPTION register. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 3-3: BLOCK DIAGRAM OF RB0/INT/ECCPAS2 PIN



Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION register).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTB pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (such as `BSF, BCF, XORWF`) with TRISB as the destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

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, 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 RB Port Change Interrupt with flag bit RBIF of the INTCON register.

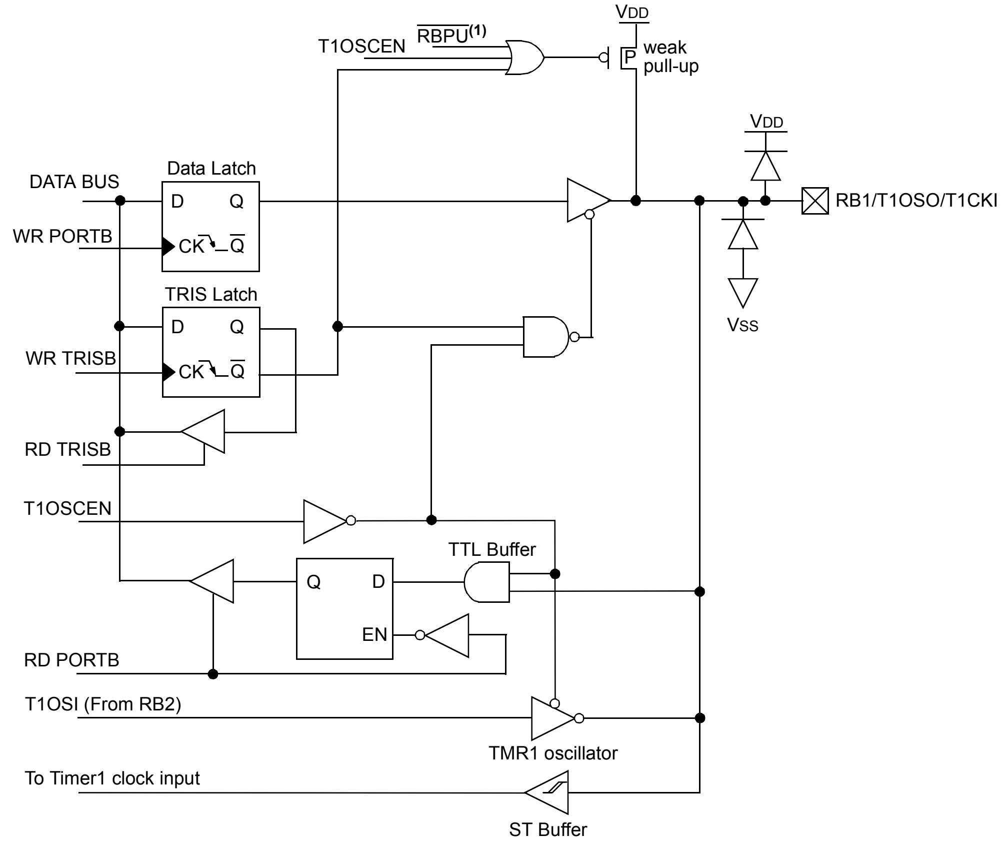
This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

1. Perform a read of PORTB to end the mismatch condition.
2. 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.

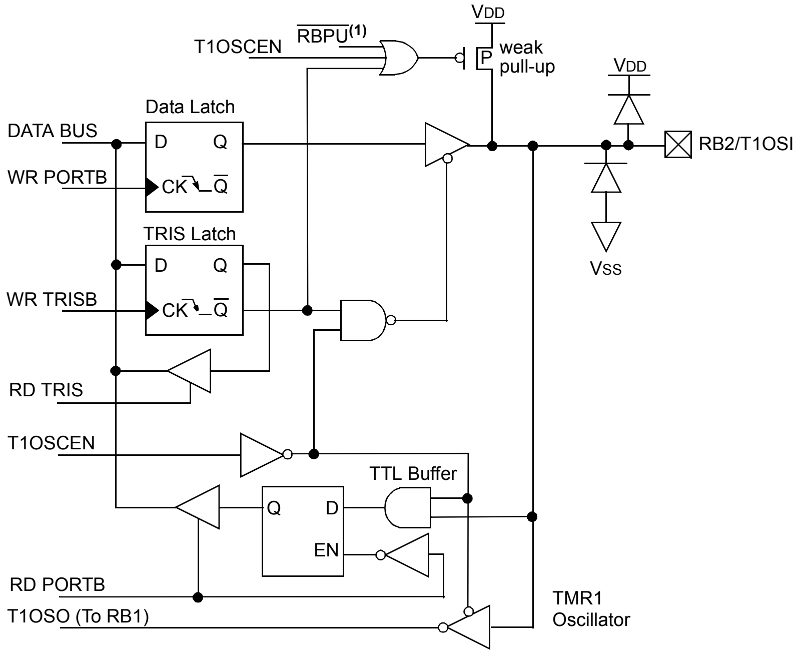
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

FIGURE 3-4: BLOCK DIAGRAM OF RB1/T1OSO/T1CKI PIN



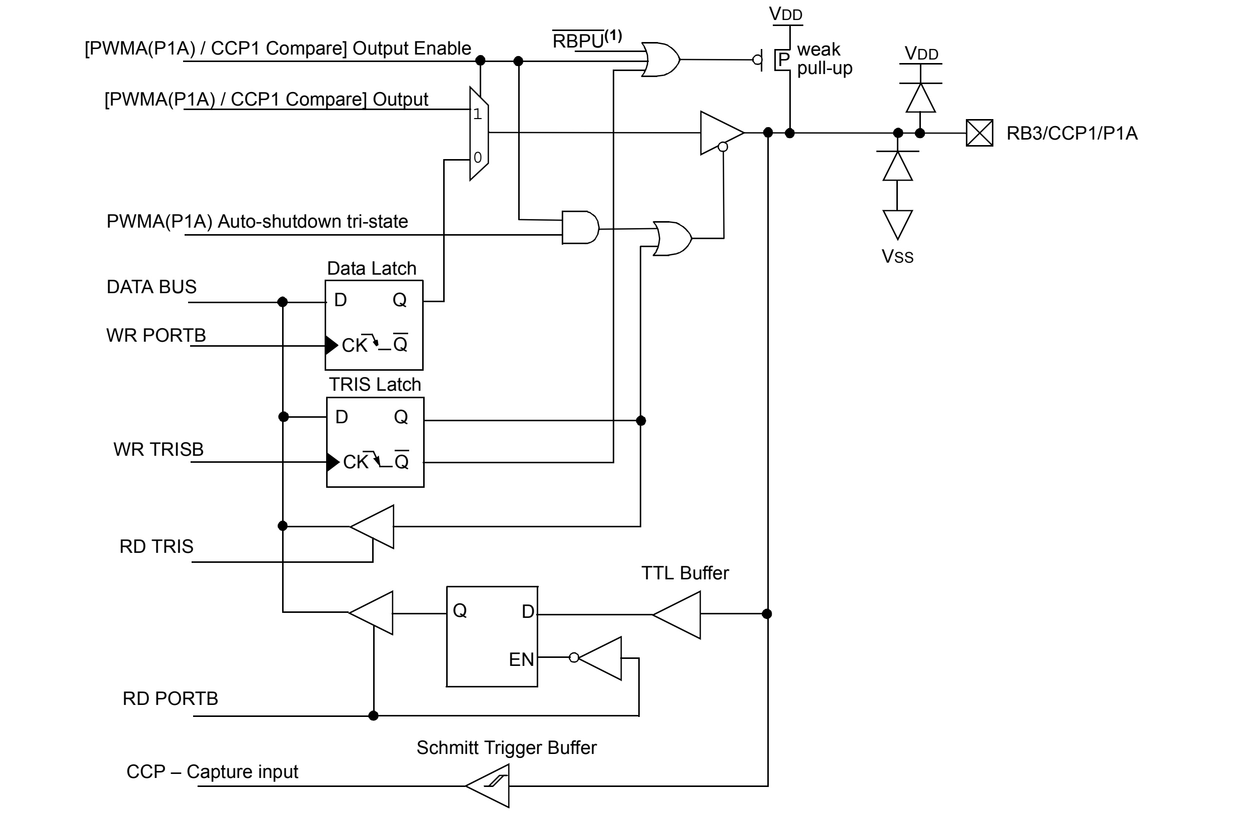
Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION register).

FIGURE 3-5: BLOCK DIAGRAM OF RB2/T1OSI PIN



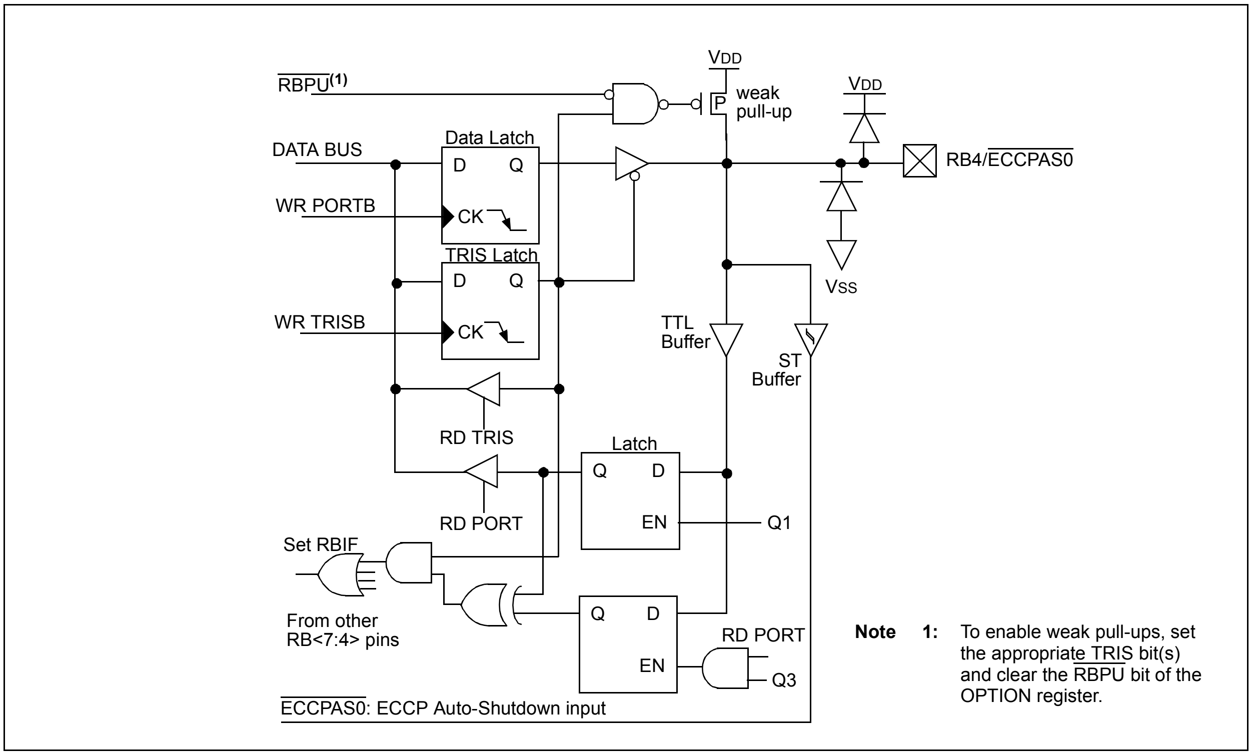
Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION register).

FIGURE 3-6: BLOCK DIAGRAM OF RB3/CCP1/P1A PIN



Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION register).

FIGURE 3-7: BLOCK DIAGRAM OF RB4/ECCPAS0 PIN



Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit of the OPTION register.