PIC18F6585/8585/6680/8680

64/68/80-Pin High-Performance, 64-Kbyte Enhanced Flash Microcontrollers with ECAN Module

High-Performance RISC CPU:

- Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 2 Mbytes
- Linear data memory addressing to 4096 bytes
- 1 Kbyte of data EEPROM
- Up to 10 MIPs operation:
  - DC – 40 MHz osc./clock input
  - 4 MHz-10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 31-level, software accessible hardware stack
- 8 x 8 Single-Cycle Hardware Multiplier

External Memory Interface (PIC18F8X8X Devices Only):

- Address capability of up to 2 Mbytes
- 16-bit interface

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- One Capture/Compare/PWM (CCP) module:
  - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
  - Compare is 16-bit, max. resolution 100 ns (TCY)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Same Capture/Compare features as CCP
  - One, two or four PWM outputs
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown on external event
  - Auto-restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
  - 3-wire SPI (supports all 4 SPI modes)
  - I^2C™ Master and Slave mode
- Enhanced Addressable USART module:
  - Supports RS-232, RS-485 and LIN 1.2
  - Programmable wake-up on Start bit
  - Auto-baud detect
- Parallel Slave Port (PSP) module

Analog Features:

- Up to 16-channel, 10-bit Analog-to-Digital Converter module (A/D) with:
  - Fast sampling rate
  - Programmable acquisition time
  - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
  - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (BOR)
- Dual analog comparators:
  - Programmable input/output configuration

ECAN Module Features:

- Message bit rates up to 1 Mbps
- Conforms to CAN 2.0B ACTIVE Specification
- Fully backward compatible with PIC18XXX8 CAN modules
- Three modes of operation:
  - Legacy, Enhanced Legacy, FIFO
- Three dedicated transmit buffers with prioritization
- Two dedicated receive buffers
- Six programmable receive/transmit buffers
- Three full 29-bit acceptance masks
- 16 full 29-bit acceptance filters with dynamic association
- DeviceNet™ data byte filter support
- Automatic remote frame handling
- Advanced Error Management features

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- 1-second programming time
- Flash/Data EEPROM Retention: > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options including:
  - Software enabled 4x Phase Lock Loop (of primary oscillator)
  - Secondary Oscillator (32 kHz) clock input
- In-Circuit Serial Programming™ (ICSP™) via two pins
- MPLAB® In-Circuit Debug (ICD) via two pins

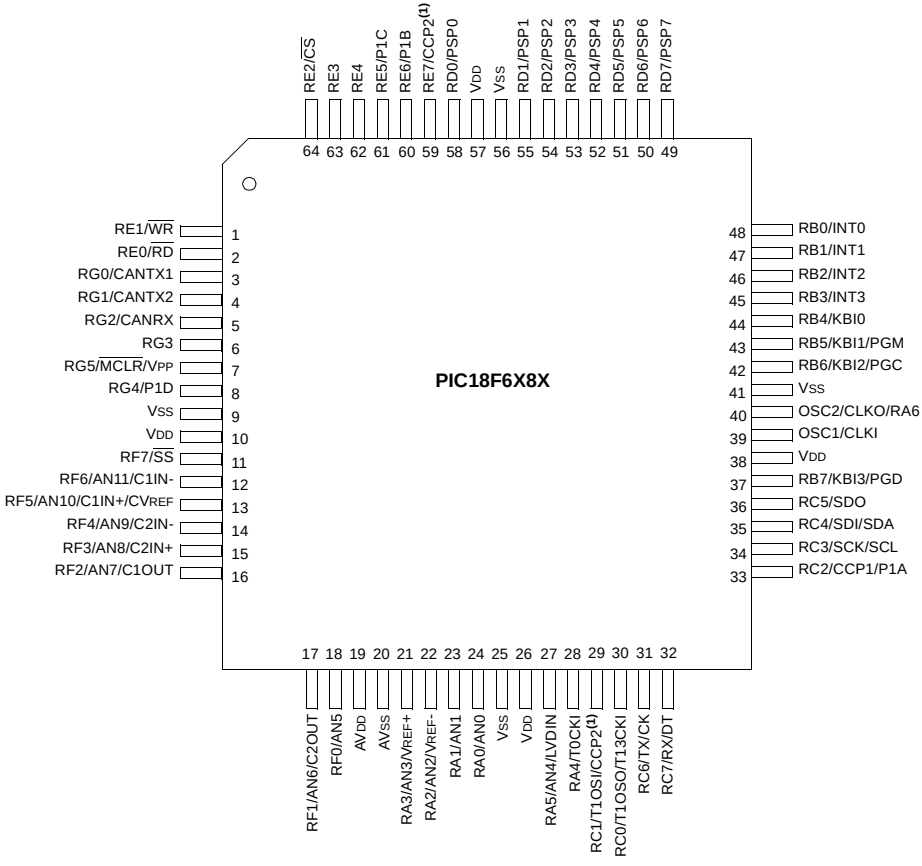
CMOS Technology:

- Low-power, high-speed Flash technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		ECAN/ AUSART	Timers 8-bit/16-bit	EMA
	Bytes	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I^2C			
PIC18F6585	48K	24576	3328	1024	53	12	1/1	Y	Y	Y/Y	2/3	N
PIC18F6680	64K	32768	3328	1024	53	12	1/1	Y	Y	Y/Y	2/3	N
PIC18F8585	48K	24576	3328	1024	69	16	1/1	Y	Y	Y/Y	2/3	Y
PIC18F8680	64K	32768	3328	1024	69	16	1/1	Y	Y	Y/Y	2/3	Y

Pin Diagrams

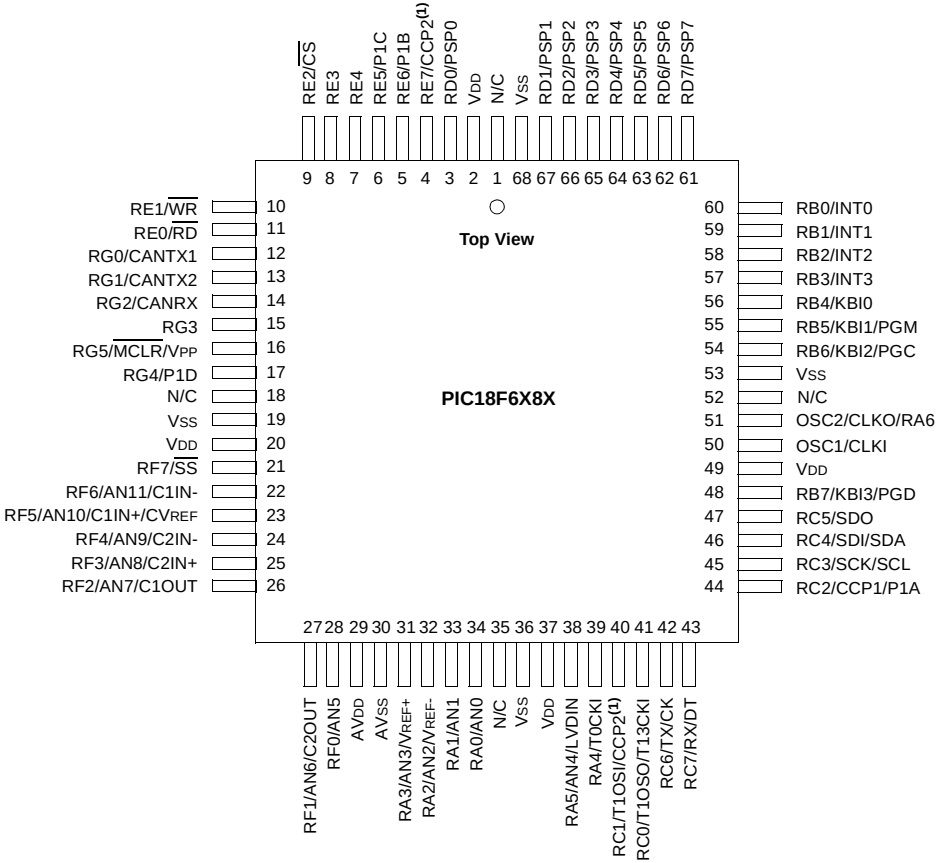
64-Pin TQFP



Note 1: CCP2 pin placement depends on CCP2MX setting.

Pin Diagrams (Continued)

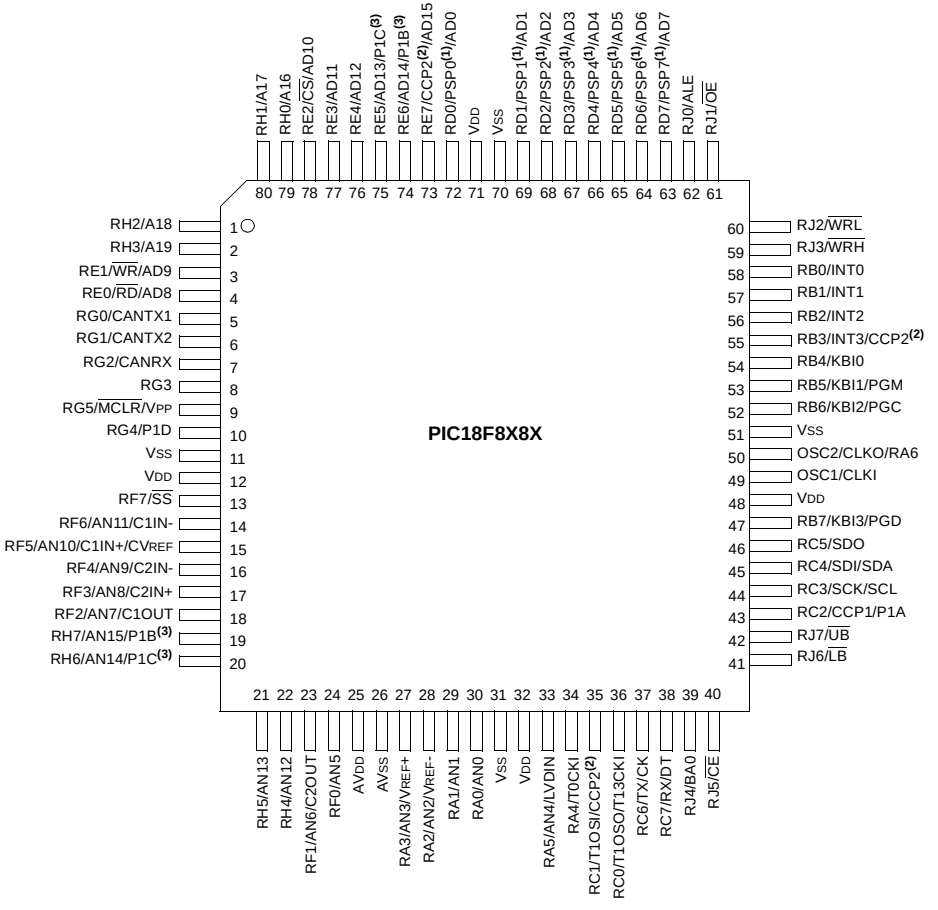
68-Pin PLCC



Note 1: CCP2 pin placement depends on CCP2MX setting.

Pin Diagrams (Continued)

80-Pin TQFP



Note 1: PSP is available only in Microcontroller mode.

2: CCP2 pin placement depends on CCP2MX and Processor mode settings.

3: P1B and P1C pin placement depends on ECCPMX setting.

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@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.

NOTES:

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F6585
- PIC18F8585
- PIC18F6680
- PIC18F8680

PIC18F6X8X devices are available in 64-pin TQFP and 68-pin PLCC packages. PIC18F8X8X devices are available in the 80-pin TQFP package. They are differentiated from each other in four ways:

1. Flash program memory (48 Kbytes for PIC18FX585 devices, 64 Kbytes for PIC18FX680)
2. A/D channels (12 for PIC18F6X8X devices, 16 for PIC18F8X8X)
3. I/O ports (7 on PIC18F6X8X devices, 9 on PIC18F8X8X)
4. External program memory interface (present only on PIC18F8X8X devices)

All other features for devices in the PIC18F6585/8585/6680/8680 family are identical. These are summarized in Table 1-1.

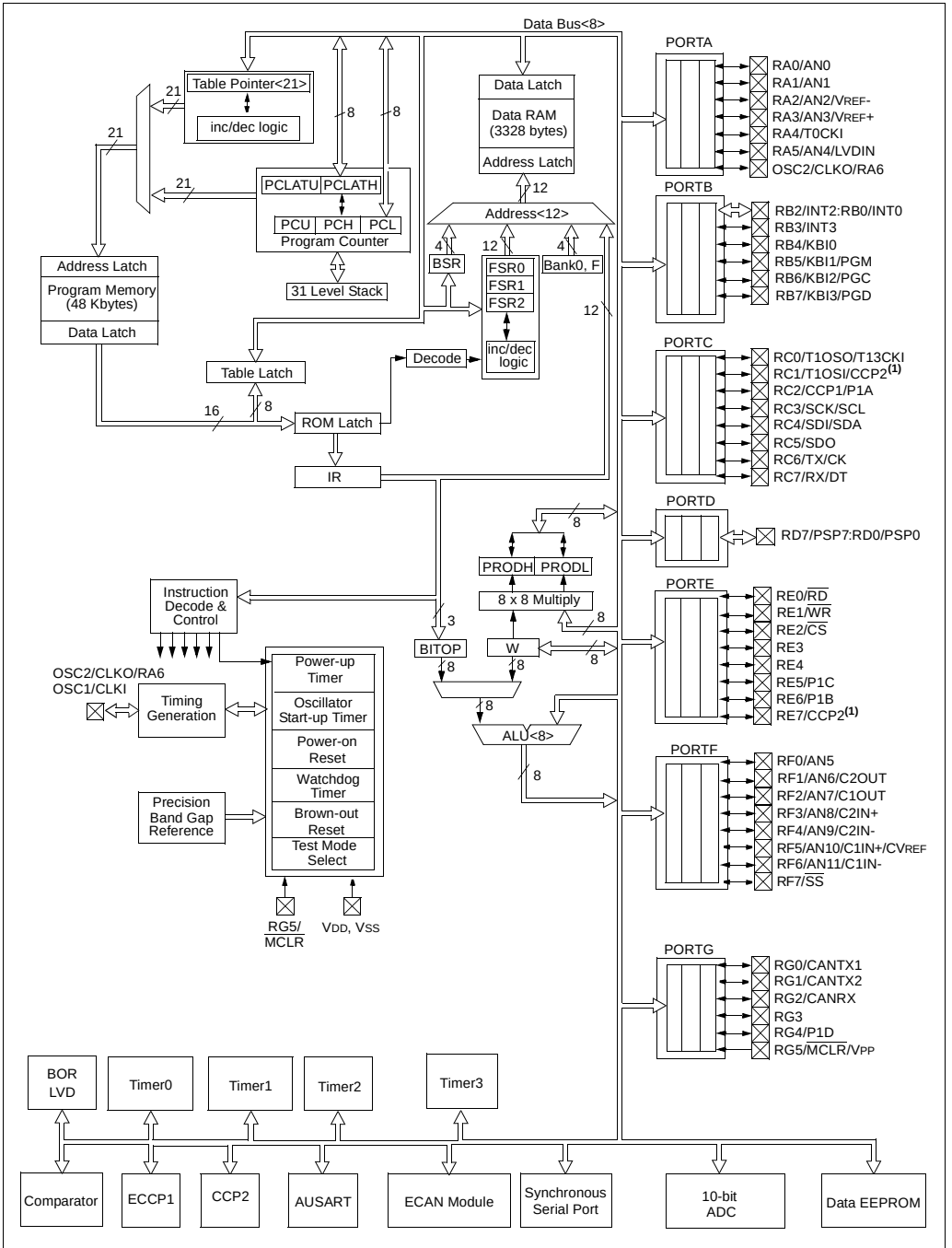
Block diagrams of the PIC18F6X8X and PIC18F8X8X devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2.

TABLE 1-1: PIC18F6585/8585/6680/8680 DEVICE FEATURES

Features	PIC18F6585	PIC18F6680	PIC18F8585	PIC18F8680
Operating Frequency	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz DC – 25 MHz w/EMA	DC – 40 MHz DC – 25 MHz w/EMA
Program Memory (Bytes)	48K	64K	48K (2 MB EMA)	64K (2 MB EMA)
Program Memory (Instructions)	24576	32768	24576	32768
Data Memory (Bytes)	3328	3328	3328	3328
Data EEPROM Memory (Bytes)	1024	1024	1024	1024
External Memory Interface	No	No	Yes	Yes
Interrupt Sources	29	29	29	29
I/O Ports	Ports A-G	Ports A-G	Ports A-H, J	Ports A-H, J
Timers	4	4	4	4
Capture/Compare/PWM Module	1	1	1	1
Enhanced Capture/Compare/PWM Module	1	1	1	1
Serial Communications	MSSP, Enhanced AUSART, ECAN	MSSP, Enhanced AUSART, ECAN	MSSP, Enhanced AUSART, ECAN	MSSP, Enhanced AUSART, ECAN
Parallel Communications	PSP	PSP	PSP(1)	PSP(1)
10-bit Analog-to-Digital Module	12 input channels	12 input channels	16 input channels	16 input channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions	75 Instructions	75 Instructions	75 Instructions
Package	64-pin TQFP, 68-pin PLCC	64-pin TQFP, 68-pin PLCC	80-pin TQFP	80-pin TQFP

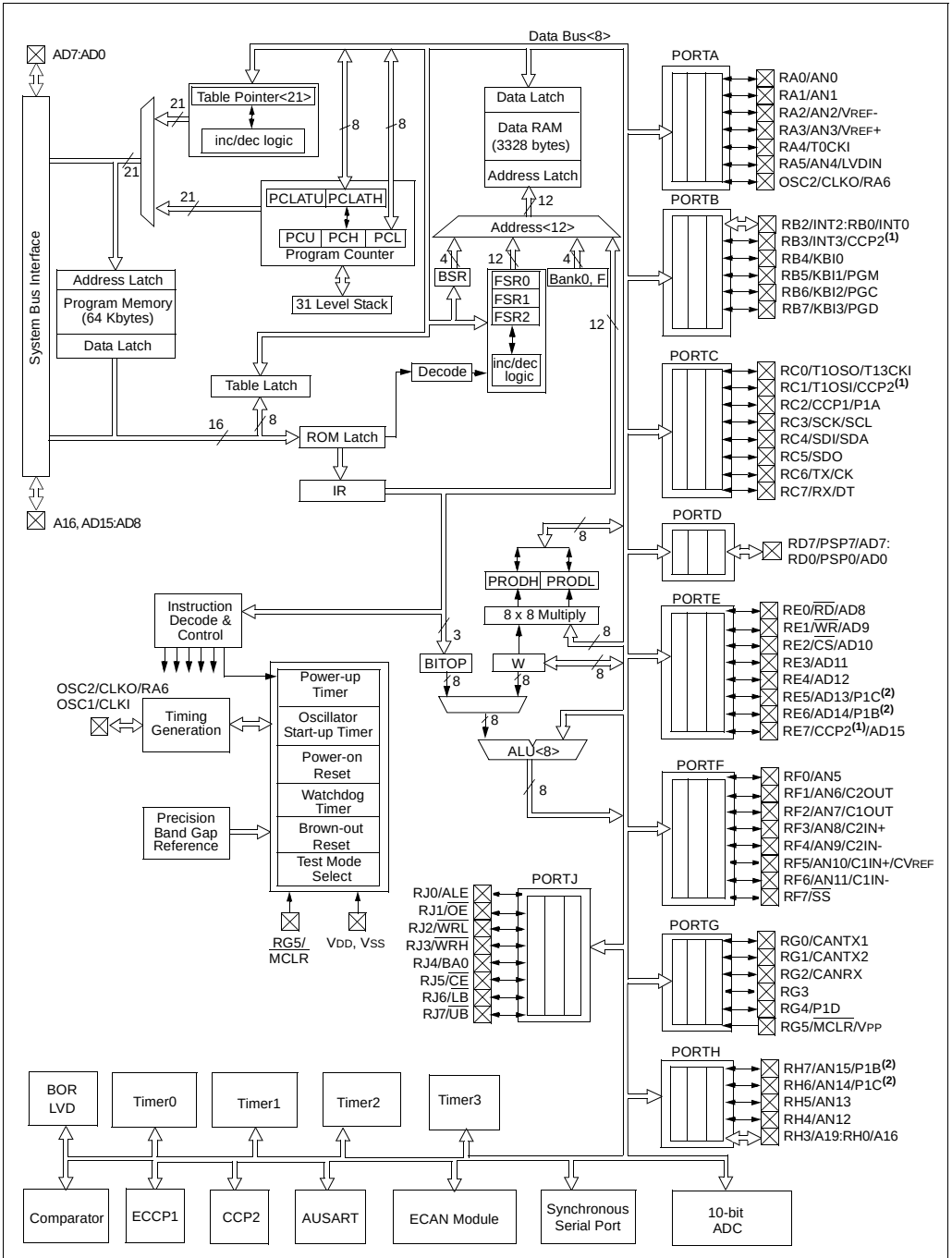
Note 1: PSP is only available in Microcontroller mode.

FIGURE 1-1: PIC18F6X8X BLOCK DIAGRAM



Note 1: The CCP2 pin placement depends on the CCP2MX and Processor mode settings.

FIGURE 1-2: PIC18F8X8X BLOCK DIAGRAM



Note 1: The CCP2 pin placement depends on the CCP2MX and Processor mode settings.
2: P1B and P1C pin placement depends on the ECCPMX setting.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
RG5/$\overline{MCLR}$/VPP	7	16	9			Master Clear (input) or programming voltage (input).
RG5				I	ST	General purpose input pin.
$\overline{MCLR}$				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P		Programming voltage input.
OSC1/CLKI	39	50	49			Oscillator crystal or external clock input.
OSC1				I	CMOS/ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI				I	CMOS	External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO/RA6	40	51	50			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6				I/O	TTL	General purpose I/O pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTA is a bidirectional I/O port.
RA0/AN0	24	34	30			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	23	33	29			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-	22	32	28			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (Low) input.
RA3/AN3/VREF+	21	31	27			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (High) input.
RA4/T0CKI	28	39	34			
RA4				I/O	ST/OD	Digital I/O – Open-drain when configured as output.
T0CKI				I	ST	Timer0 external clock input.
RA5/AN4/LVDIN	27	38	33			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
LVDIN				I	Analog	Low-voltage detect input.
RA6						See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT0	48	60	58			
RB0				I/O	TTL	Digital I/O.
INT0				I	ST	External interrupt 0.
RB1/INT1	47	59	57			
RB1				I/O	TTL	Digital I/O.
INT1				I	ST	External interrupt 1.
RB2/INT2	46	58	56			
RB2				I/O	TTL	Digital I/O.
INT2				I	ST	External interrupt 2.
RB3/INT3/CCP2	45	57	55			
RB3				I/O	TTL	Digital I/O.
INT3				I/O	ST	External interrupt 3.
CCP2(1)				I/O	ST	Capture 2 input/Compare 2 output/ PWM 2 output.
RB4/KBI0	44	56	54			
RB4				I/O	TTL	Digital I/O.
KBI0				I	ST	Interrupt-on-change pin.
RB5/KBI1/PGM	43	55	53			
RB5				I/O	TTL	Digital I/O.
KBI1				I	ST	Interrupt-on-change pin.
PGM				I/O	ST	Low-Voltage ICSP Programming enable pin.
RB6/KBI2/PGC	42	54	52			
RB6				I/O	TTL	Digital I/O.
KBI2				I	ST	Interrupt-on-change pin.
PGC				I/O	ST	In-circuit debugger and ICSP programming clock.
RB7/KBI3/PGD	37	48	47			
RB7				I/O	TTL	Digital I/O.
KBI3				I/O	ST	Interrupt-on-change pin.
PGD						In-circuit debugger and ICSP programming data.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	30	41	36			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2	29	40	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2(1, 4)				I/O	ST	CCP2 Capture input/Compare output/ PWM 2 output.
RC2/CCP1/P1A	33	44	43			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	CCP1 Capture input/Compare output.
P1A				I/O	ST	CCP1 PWM output A.
RC3/SCK/SCL	34	45	44			
RC3				I/O	ST	Digital I/O.
SCK				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL				I/O	ST	Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	35	46	45			
RC4				I/O	ST	Digital I/O.
SDI				I	ST	SPI data in.
SDA				I/O	ST	I^2C data I/O.
RC5/SDO	36	47	46			
RC5				I/O	ST	Digital I/O.
SDO				O	—	SPI data out.
RC6/TX/CK	31	42	37			
RC6				I/O	ST	Digital I/O.
TX				O	—	USART asynchronous transmit.
CK				I/O	ST	USART synchronous clock (see RX/DT).
RC7/RX/DT	32	43	38			
RC7				I/O	ST	Digital I/O.
RX				I	ST	USART 1 asynchronous receive.
DT				I/O	ST	USART 1 synchronous data (see TX/CK).

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTD is a bidirectional I/O port. These pins have TTL input buffers when external memory is enabled.
RD0/PSP0/AD0	58	3	72			
RD0				I/O	ST	Digital I/O.
PSP0(6)				I/O	TTL	Parallel Slave Port data.
AD0(3)				I/O	TTL	External memory address/data 0.
RD1/PSP1/AD1	55	67	69			
RD1				I/O	ST	Digital I/O.
PSP1(6)				I/O	TTL	Parallel Slave Port data.
AD1(3)				I/O	TTL	External memory address/data 1.
RD2/PSP2/AD2	54	66	68			
RD2				I/O	ST	Digital I/O.
PSP2(6)				I/O	TTL	Parallel Slave Port data.
AD2(3)				I/O	TTL	External memory address/data 2.
RD3/PSP3/AD3	53	65	67			
RD3				I/O	ST	Digital I/O.
PSP3(6)				I/O	TTL	Parallel Slave Port data.
AD3(3)				I/O	TTL	External memory address/data 3.
RD4/PSP4/AD4	52	64	66			
RD4				I/O	ST	Digital I/O.
PSP4(6)				I/O	TTL	Parallel Slave Port data.
AD4(3)				I/O	TTL	External memory address/data 4.
RD5/PSP5/AD5	51	63	65			
RD5				I/O	ST	Digital I/O.
PSP5(6)				I/O	TTL	Parallel Slave Port data.
AD5(3)				I/O	TTL	External memory address/data 5.
RD6/PSP6/AD6	50	62	64			
RD6				I/O	ST	Digital I/O.
PSP6(6)				I/O	TTL	Parallel Slave Port data.
AD6(3)				I/O	TTL	External memory address/data 6.
RD7/PSP7/AD7	49	61	63			
RD7				I/O	ST	Digital I/O.
PSP7(6)				I/O	TTL	Parallel Slave Port data.
AD7(3)				I/O	TTL	External memory address/data 7.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AD8	2	11	4			
RE0				I/O	ST	Digital I/O.
$\overline{RD}$(6)				I	TTL	Read control for Parallel Slave Port (see $\overline{WR}$ and $\overline{CS}$ pins).
AD8(3)				I/O	TTL	External memory address/data 8.
RE1/$\overline{WR}$/AD9	1	10	3			
RE1				I/O	ST	Digital I/O.
$\overline{WR}$(6)				I	TTL	Write control for Parallel Slave Port (see $\overline{CS}$ and $\overline{RD}$ pins).
AD9(3)				I/O	TTL	External memory address/data 9.
RE2/$\overline{CS}$/AD10	64	9	78			
RE2				I/O	ST	Digital I/O.
$\overline{CS}$(6)				I	TTL	Chip select control for Parallel Slave Port (see $\overline{RD}$ and $\overline{WR}$).
AD10(3)				I/O	TTL	External memory address/data 10.
RE3/AD11	63	8	77			
RE3				I/O	ST	Digital I/O.
AD11(3)				I/O	TTL	External memory address/data 11.
RE4/AD12	62	7	76			
RE4				I/O	ST	Digital I/O.
AD12(3)				I/O	TTL	External memory address/data 12.
RE5/AD13/P1C	61	6	75			
RE5				I/O	ST	Digital I/O.
AD13(3)				I/O	TTL	External memory address/data 13.
P1C(7)				I/O	ST	ECCP1 PWM output C.
RE6/AD14/P1B	60	5	74			
RE6				I/O	ST	Digital I/O.
AD14(3)				I/O	TTL	External memory address/data 14.
P1B(7)				I/O	ST	ECCP1 PWM output B.
RE7/CCP2/AD15	59	4	73			
RE7				I/O	ST	Digital I/O.
CCP2(1,4)				I/O	ST	Capture 2 input/Compare 2 output/ PWM 2 output.
AD15(3)				I/O	TTL	External memory address/data 15.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTF is a bidirectional I/O port.
RF0/AN5	18	28	24			
RF0				I/O	ST	Digital I/O.
AN5				I	Analog	Analog input 5.
RF1/AN6/C2OUT	17	27	23			
RF1				I/O	ST	Digital I/O.
AN6				I	Analog	Analog input 6.
C2OUT				O	ST	Comparator 2 output.
RF2/AN7/C1OUT	16	26	18			
RF2				I/O	ST	Digital I/O.
AN7				I	Analog	Analog input 7.
C1OUT				O	ST	Comparator 1 output.
RF3/AN8/C2IN+	15	25	17			
RF1				I/O	ST	Digital I/O.
AN8				I	Analog	Analog input 8.
C2IN+				I	Analog	Comparator 2 input (+).
RF4/AN9/C2IN-	14	24	16			
RF1				I/O	ST	Digital I/O.
AN9				I	Analog	Analog input 9.
C2IN-				I	Analog	Comparator 2 input (-).
RF5/AN10/C1IN+/CVREF	13	23	15			
RF1						
AN10				I/O	ST	Digital I/O.
C1IN+				I	Analog	Analog input 10.
CVREF				I	Analog	Comparator 1 input (+).
				O	Analog	Comparator VREF output.
RF6/AN11/C1IN-	12	22	14			
RF6				I/O	ST	Digital I/O.
AN11				I	Analog	Analog input 11.
C1IN-				I	Analog	Comparator 1 input (-)
RF7/$\overline{SS}$	11	21	13			
RF7				I/O	ST	Digital I/O.
$\overline{SS}$				I	TTL	SPI slave select input.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTG is a bidirectional I/O port.
RG0/CANTX1	3	12	5			
RG0				I/O	ST	Digital I/O.
CANTX1				O	TTL	CAN bus transmit 1.
RG1/CANTX2	4	13	6			
RG1				I/O	ST	Digital I/O.
CANTX2				O	TTL	CAN bus transmit 2.
RG2/CANRX	5	14	7			
RG2				I/O	ST	Digital I/O.
CANRX				I	TTL	CAN bus receive.
RG3	6	15	8			
RG3				I/O	ST	Digital I/O.
RG4/P1D	8	17	10			
RG4				I/O	ST	Digital I/O.
P1D				O	TTL	ECCP1 PWM output D.
RG5	7	16	9	I	ST	General purpose input pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTH is a bidirectional I/O port[5].
RH0/A16	—	—	79			
RH0				I/O	ST	Digital I/O.
A16				O	TTL	External memory address 16.
RH1/A17	—	—	80			
RH1				I/O	ST	Digital I/O.
A17				O	TTL	External memory address 17.
RH2/A18	—	—	1			
RH2				I/O	ST	Digital I/O.
A18				O	TTL	External memory address 18.
RH3/A19	—	—	2			
RH3				I/O	ST	Digital I/O.
A19				O	TTL	External memory address 19.
RH4/AN12	—	—	22			
RH4				I/O	ST	Digital I/O.
AN12				I	Analog	Analog input 12.
RH5/AN13	—	—	21			
RH5				I/O	ST	Digital I/O.
AN13				I	Analog	Analog input 13.
RH6/AN14/P1C	—	—	20			
RH6				I/O	ST	Digital I/O.
AN14				I	Analog	Analog input 14.
P1C[7]				I/O	ST	Alternate CCP1 PWM output C.
RH7/AN15/P1B	—	—	19			
RH7				I/O	ST	Digital I/O.
AN15				I	Analog	Analog input 15.
P1B[7]						Alternate CCP1 PWM output B.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
						PORTJ is a bidirectional I/O port(5).
RJ0/ALE	—	—	62			
RJ0				I/O	ST	Digital I/O.
ALE				O	TTL	External memory address latch enable.
RJ1/$\overline{OE}$	—	—	61			
RJ1				I/O	ST	Digital I/O.
$\overline{OE}$				O	TTL	External memory output enable.
RJ2/$\overline{WRL}$	—	—	60			
RJ2				I/O	ST	Digital I/O.
$\overline{WRL}$				O	TTL	External memory write low control.
RJ3/$\overline{WRH}$	—	—	59			
RJ3				I/O	ST	Digital I/O.
$\overline{WRH}$				O	TTL	External memory write high control.
RJ4/BA0	—	—	39			
RJ4				I/O	ST	Digital I/O.
BA0				O	TTL	System bus byte address 0 control.
RJ5/$\overline{CE}$	—	—	40	I/O	ST	Digital I/O
$\overline{CE}$				O	TTL	External memory chip enable.
RJ6/$\overline{LB}$	—	—	42			
RJ6				I/O	ST	Digital I/O.
$\overline{LB}$				O	TTL	External memory low byte select.
RJ7/$\overline{UB}$	—	—	41			
RJ7				I/O	ST	Digital I/O.
$\overline{UB}$				O	TTL	External memory high byte select.
VSS	9, 25, 41, 56	19, 36, 53, 68	11, 31, 51, 70	P	—	Ground reference for logic and I/O pins.
VDD	10, 26, 38, 57	2, 20, 37, 49	12, 32, 48, 71	P	—	Positive supply for logic and I/O pins.
AVSS	20	30	26	P	—	Ground reference for analog modules.
AVDD	19	29	25	P	—	Positive supply for analog modules.
NC	—	1, 18, 35, 52	—	—	—	No connect.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.

2: Default assignment when CCP2MX is set.

3: External memory interface functions are only available on PIC18F8X8X devices.

4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.

5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.

6: PSP is available in Microcontroller mode only.

7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

NOTES:

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

The PIC18F6585/8585/6680/8680 devices can be operated in eleven different oscillator modes. The user can program four configuration bits (FOSC3, FOSC2, FOSC1 and FOSC0) to select one of these eleven modes:

1. LP Low-Power Crystal
2. XT Crystal/Resonator
3. HS High-Speed Crystal/Resonator
4. RC External Resistor/Capacitor
5. EC External Clock
6. ECIO External Clock with I/O pin enabled
7. HS+PLL High-Speed Crystal/Resonator with PLL enabled
8. RCIO External Resistor/Capacitor with I/O pin enabled
9. ECIO+SPLL External Clock with software controlled PLL
10. ECIO+PLL External Clock with PLL and I/O pin enabled
11. HS+SPLL High-Speed Crystal/Resonator with software control

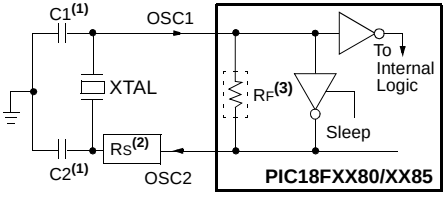
2.2 Crystal Oscillator/Ceramic Resonators

In XT, LP, HS, HS+PLL or HS+SPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-1 shows the pin connections.

The PIC18F6585/8585/6680/8680 oscillator design requires the use of a parallel cut crystal.

Note:	Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

FIGURE 2-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for recommended values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the oscillator mode chosen.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq**	**C1**	**C2**
XT	455 kHz	68-100 pF	68-100 pF
	2.0 MHz	15-68 pF	15-68 pF
	4.0 MHz	15-68 pF	15-68 pF
HS	8.0 MHz	10-68 pF	10-68 pF
	16.0 MHz	10-22 pF	10-22 pF
These values are for design guidance only. See notes following this table.			
Resonators Used:			
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

Note 1:	Higher capacitance increases the stability of the oscillator, but also increases the start-up time.
2:	When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use high gain HS mode, try a lower frequency resonator, or switch to a crystal oscillator.
3:	Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components, or verify oscillator performance.

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Ranges Tested:			
Mode	**Freq**	**C1**	**C2**
LP	32.0 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1.0 MHz	15 pF	15 pF
	4.0 MHz	15 pF	15 pF
HS	4.0 MHz	15 pF	15 pF
	8.0 MHz	15-33 pF	15-33 pF
	20.0 MHz	15-33 pF	15-33 pF
	25.0 MHz	TBD	TBD
These values are for design guidance only. See notes following this table.			
Crystals Used			
32.0 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1.0 MHz	ECS ECS-10-13-1	± 50 PPM	
4.0 MHz	ECS ECS-40-20-1	± 50 PPM	
8.0 MHz	Epson CA-301 8.000M-C	± 30 PPM	
20.0 MHz	Epson CA-301 20.000M-C	± 30 PPM	

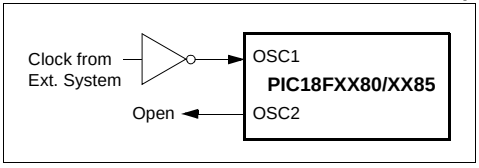
Note 1: Higher capacitance increases the stability of the oscillator, but also increases the start-up time.

2: Rs (see Figure 2-1) may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specifications.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components, or verify oscillator performance.

An external clock source may also be connected to the OSC1 pin in the HS, XT and LP modes, as shown in Figure 2-2.

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

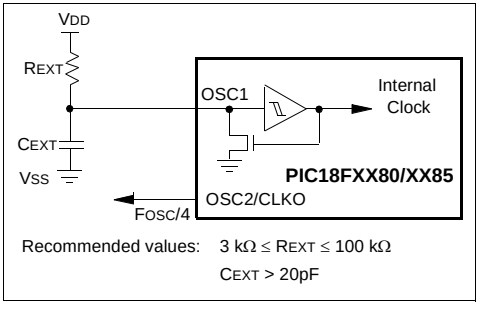


2.3 RC Oscillator

For timing insensitive applications, the “RC” and “RCIO” device options offer additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit, due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 2-3 shows how the R/C combination is connected.

In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic.

FIGURE 2-3: RC OSCILLATOR MODE



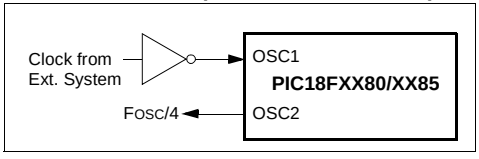
The RCIO Oscillator mode functions like the RC mode except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

2.4 External Clock Input

The EC, ECIO, EC+PLL and EC+SPLL Oscillator modes require an external clock source to be connected to the OSC1 pin. The feedback device between OSC1 and OSC2 is turned off in these modes to save current. There is a maximum 1.5 μs start-up required after a Power-on Reset, or wake-up from Sleep mode.

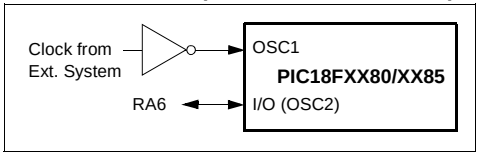
In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)



The ECIO Oscillator mode functions like the EC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)



2.5 Phase Locked Loop (PLL)

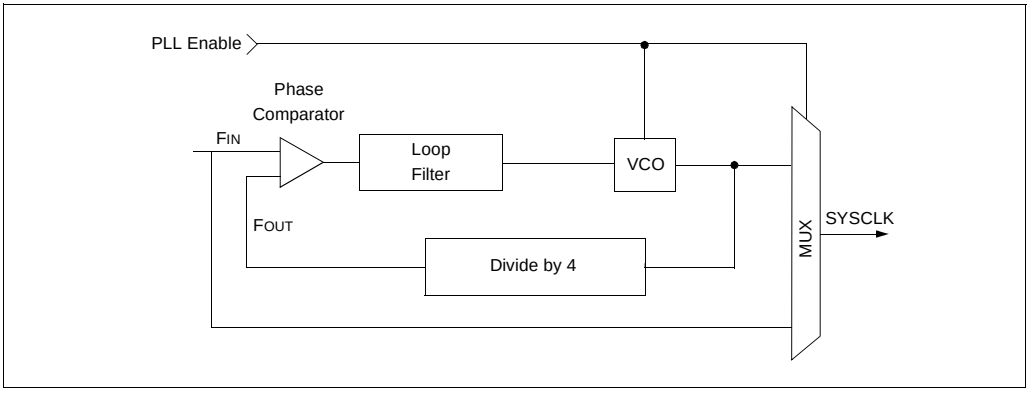
A Phase Locked Loop circuit is provided as a programmable option for users that want to multiply the frequency of the incoming oscillator signal by 4. For an input clock frequency of 10 MHz, the internal clock frequency will be multiplied to 40 MHz. This is useful for customers who are concerned with EMI due to high-frequency crystals.

The PLL can only be enabled when the oscillator configuration bits are programmed for High-Speed Oscillator or External Clock mode. If they are programmed for any other mode, the PLL is not enabled and the system clock will come directly from OSC1. There are two types of PLL modes: Software Controlled PLL and Configuration bits Controlled PLL. In Software Controlled PLL mode, PIC18F6585/8585/6680/8680 executes at regular clock frequency after all Reset conditions. During execution, application can enable PLL and switch to 4x clock frequency operation by setting the PLLEN bit in the OSCCON register. In Configuration bits Controlled PLL mode, PIC18F6585/8585/6680/8680 always executes with 4x clock frequency.

The type of PLL is selected by programming the FOSC<3:0> configuration bits in the CONFIG1H Configuration register. The oscillator mode is specified during device programming.

A PLL lock timer is used to ensure that the PLL has locked before device execution starts. The PLL lock timer has a time-out that is called TPLL.

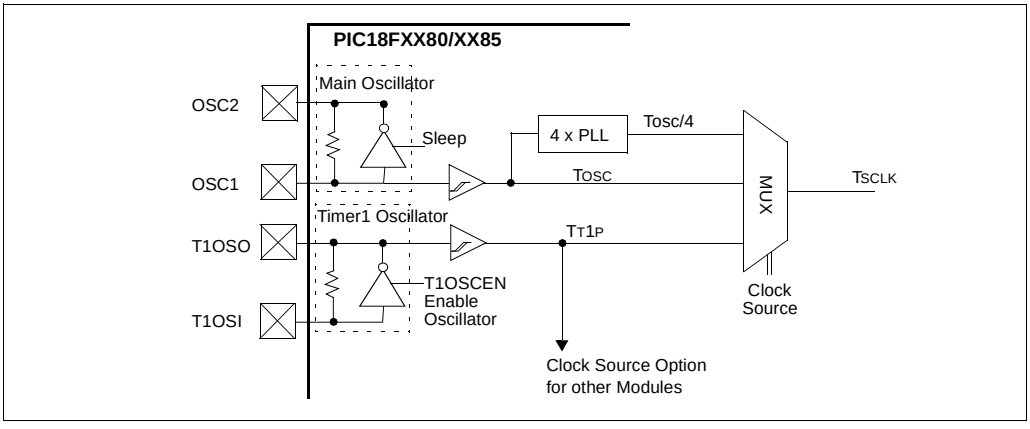
FIGURE 2-6: PLL BLOCK DIAGRAM



2.6 Oscillator Switching Feature

The PIC18F6585/8585/6680/8680 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low-frequency clock source. For the PIC18F6585/8585/6680/8680 devices, this alternate clock source is the Timer1 oscillator. If a low-frequency crystal (32 kHz, for example) has been attached to the Timer1 oscillator pins and the Timer1 oscillator has been enabled, the device can switch to a low-power execution mode. Figure 2-7 shows a block diagram of the system clock sources. The clock switching feature is enabled by programming the Oscillator Switching Enable ($\overline{\text{OSCSEN}}$) bit in configuration register, CONFIG1H, to a '0'. Clock switching is disabled in an erased device. See **Section 12.0 "Timer1 Module"** for further details of the Timer1 oscillator. See **Section 24.0 "Special Features of the CPU"** for configuration register details.

FIGURE 2-7: DEVICE CLOCK SOURCES



2.6.1 SYSTEM CLOCK SWITCH BIT

The system clock source switching is performed under software control. The System Clock Switch bits, SCS1:SCS0 (OSCCON<1:0>), control the clock switching. When the SCS0 bit is '0', the system clock source comes from the main oscillator that is selected by the FOSC configuration bits in configuration register, CONFIG1H. When the SCS0 bit is set, the system clock source will come from the Timer1 oscillator. The SCS0 bit is cleared on all forms of Reset.

When FOSC bits are programmed for software PLL mode, the SCS1 bit can be used to select between primary oscillator/clock and PLL output. The SCS1 bit will only have an effect on the system clock if the PLL is enabled (PLLEN = 1) and locked (LOCK = 1), else it will be forced clear. When programmed with Configuration Controlled PLL mode, the SCS1 bit will be forced clear.

Note:	The Timer1 oscillator must be enabled and operating to switch the system clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON). If the Timer1 oscillator is not enabled, then any write to the SCS0 bit will be ignored (SCS0 bit forced cleared) and the main oscillator will continue to be the system clock source.

REGISTER 2-1: OSCCON REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	LOCK	PLLEN	SCS1	SCS0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **LOCK:** Phase Lock Loop Lock Status bit
1 = Phase Lock Loop output is stable as system clock
0 = Phase Lock Loop output is not stable and output cannot be used as system clock

bit 2 **PLLEN(1):** Phase Lock Loop Enable bit
1 = Enable Phase Lock Loop output as system clock
0 = Disable Phase Lock Loop

bit 1 **SCS1:** System Clock Switch bit 1
When PLLEN and LOCK bits are set:
1 = Use PLL output
0 = Use primary oscillator/clock input pin
When PLLEN or LOCK bit is cleared:
Bit is forced clear.

bit 0 **SCS0(2):** System Clock Switch bit 0
When $\overline{\text{OSCSEN}}$ configuration bit = 0 and T1OSCEN bit = 1:
1 = Switch to Timer1 oscillator/clock pin
0 = Use primary oscillator/clock input pin
When $\overline{\text{OSCSEN}}$ and T1OSCEN are in other states:
Bit is forced clear.

Note 1: PLLEN bit is ignored when configured for ECIO+PLL and HS+PLL. This bit is used in ECIO+SPLL and HS+SPLL modes only.

2: The setting of SCS0 = 1 supersedes SCS1 = 1.

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

2.6.2 OSCILLATOR TRANSITIONS

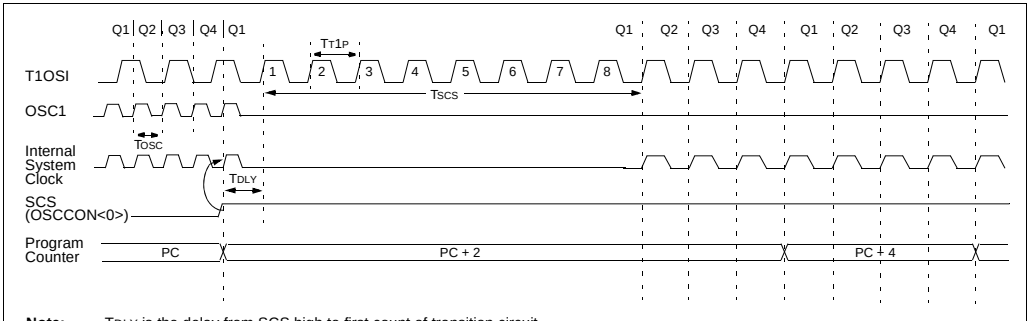
PIC18F6585/8585/6680/8680 devices contain circuitry to prevent “glitches” when switching between oscillator sources. Essentially, the circuitry waits for eight rising edges of the clock source that the processor is switching to. This ensures that the new clock source is stable and that its pulse width will not be less than the shortest pulse width of the two clock sources.

A timing diagram, indicating the transition from the main oscillator to the Timer1 oscillator, is shown in Figure 2-8. The Timer1 oscillator is assumed to be running all the time. After the SCS0 bit is set, the processor is frozen at the next occurring Q1 cycle. After eight synchronization cycles are counted from the Timer1 oscillator, operation resumes. No additional delays are required after the synchronization cycles.

The sequence of events that takes place when switching from the Timer1 oscillator to the main oscillator will depend on the mode of the main oscillator. In addition to eight clock cycles of the main oscillator, additional delays may take place.

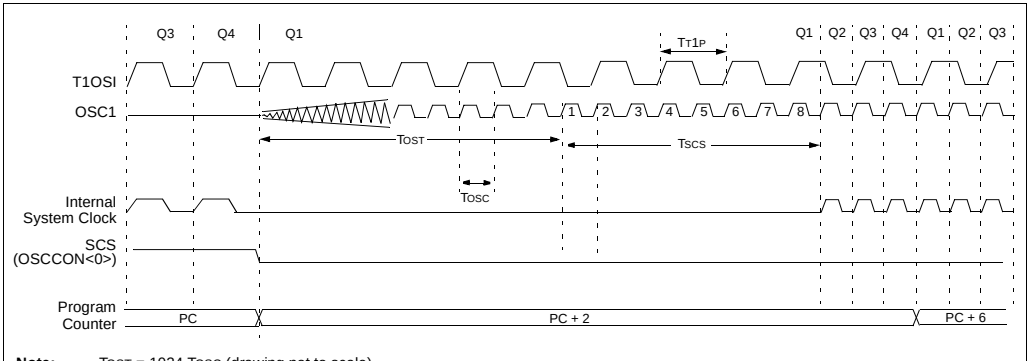
If the main oscillator is configured for an external crystal (HS, XT, LP), then the transition will take place after an oscillator start-up time (TOST) has occurred. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for HS, XT and LP modes, is shown in Figure 2-9.

FIGURE 2-8: TIMING DIAGRAM FOR TRANSITION FROM OSC1 TO TIMER1 OSCILLATOR



Note: TDLY is the delay from SCS high to first count of transition circuit.

FIGURE 2-9: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (HS, XT, LP)

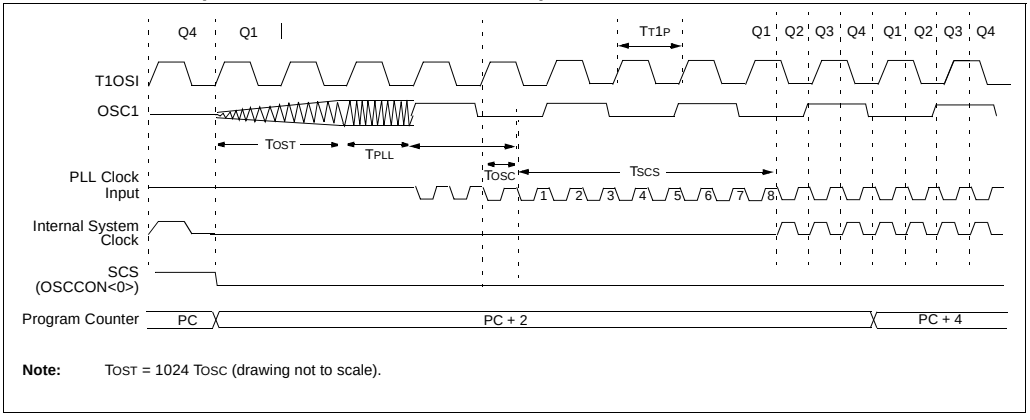


Note: TOST = 1024 TOSC (drawing not to scale).

If the main oscillator is configured for HS mode with PLL active, an oscillator start-up time (TOST) plus an additional PLL time-out (TPLL) will occur. The PLL time-out is typically 2 ms and allows the PLL to lock to the main oscillator frequency. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for HS-PLL mode, is shown in Figure 2-10.

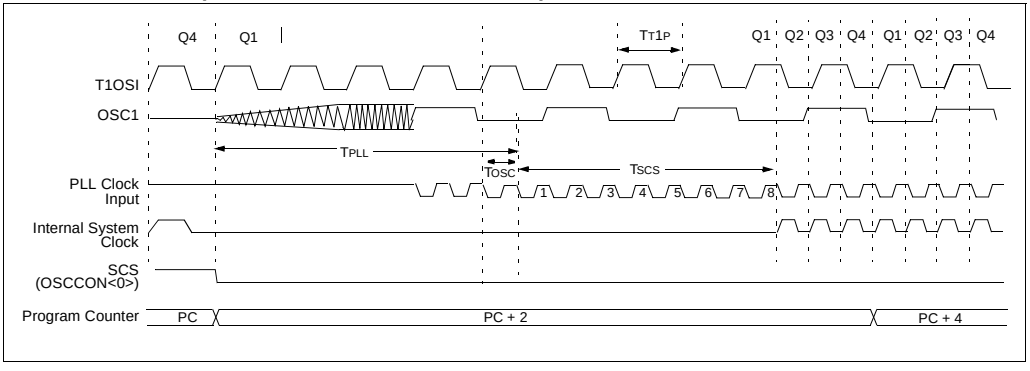
If the main oscillator is configured for EC mode with PLL active, only the PLL time-out (TPLL) will occur. The PLL time-out is typically 2 ms and allows the PLL to lock to the main oscillator frequency. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for EC with PLL active, is shown in Figure 2-11.

FIGURE 2-10: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (HS WITH PLL ACTIVE, SCS1 = 1)



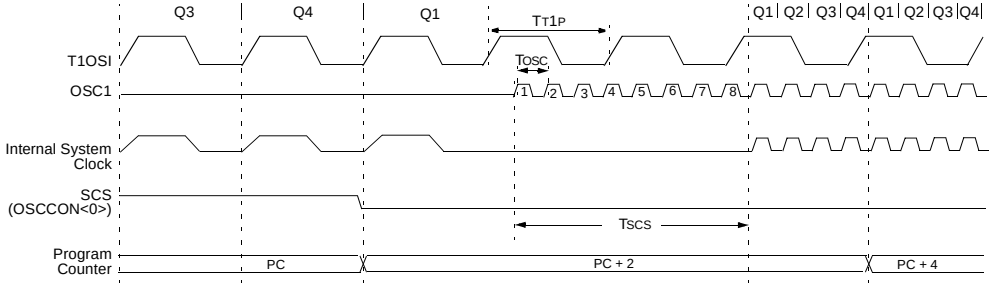
Note: TOST = 1024 TOSC (drawing not to scale).

FIGURE 2-11: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (EC WITH PLL ACTIVE, SCS1 = 1)



If the main oscillator is configured in the RC, RCIO, EC or ECIO modes, there is no oscillator start-up time-out. Operation will resume after eight cycles of the main oscillator have been counted. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for RC, RCIO, EC and ECIO modes, is shown in Figure 2-12.

FIGURE 2-12: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (RC, EC)



Note: RC Oscillator mode assumed.

2.7 Effects of Sleep Mode on the On-Chip Oscillator

When the device executes a SLEEP instruction, the on-chip clocks and oscillator are turned off and the device is held at the beginning of an instruction cycle (Q1 state). With the oscillator off, the OSC1 and OSC2 signals will stop oscillating. Since all the transistor switching currents have been removed, Sleep mode achieves the lowest current consumption of the device (only leakage currents). Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The user can wake from Sleep through external Reset, Watchdog Timer Reset, or through an interrupt.

TABLE 2-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

OSC Mode	OSC1 Pin	OSC2 Pin
RC	Floating, external resistor should pull high	At logic low
RCIO	Floating, external resistor should pull high	Configured as PORTA, bit 6
ECIO	Floating	Configured as PORTA, bit 6
EC	Floating	At logic low
LP, XT, and HS	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level

Note: See Table 3-1 in **Section 3.0 “Reset”**, for time-outs due to Sleep and $\overline{\text{MCLR}}$ Reset.

2.8 Power-up Delays

Power-up delays are controlled by two timers so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply and clock are stable. For additional information on Reset operation, see **Section 3.0 “Reset”**.

The first timer is the Power-up Timer (PWRT) which optionally provides a fixed delay of 72 ms (nominal) on power-up only (POR and BOR). The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable.

With the PLL enabled (HS+PLL and EC+PLL Oscillator mode), the time-out sequence following a Power-on Reset is different from other oscillator modes. The time-out sequence is as follows: First, the PWRT time-out is invoked after a POR time delay has expired. Then, the Oscillator Start-up Timer (OST) is invoked. However, this is still not a sufficient amount of time to allow the PLL to lock at high frequencies. The PWRT timer is used to provide an additional fixed 2 ms (nominal) time-out to allow the PLL ample time to lock to the incoming clock frequency.

NOTES:

3.0 RESET

The PIC18F6585/8585/6680/8680 devices differentiate between various kinds of Reset:

a) Power-on Reset (POR)
b) $\overline{\text{MCLR}}$ Reset during normal operation
c) $\overline{\text{MCLR}}$ Reset during Sleep
d) Watchdog Timer (WDT) Reset (during normal operation)
e) Programmable Brown-out Reset (BOR)
f) `RESET` Instruction
g) Stack Full Reset
h) Stack Underflow Reset

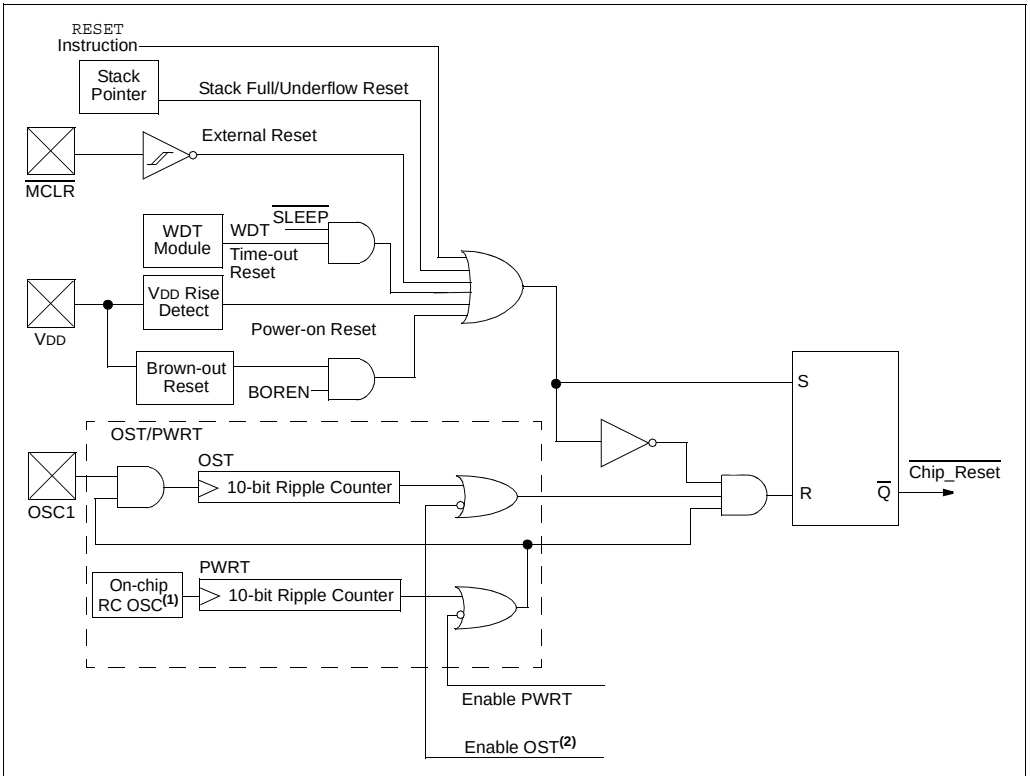
Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a "Reset state" on Power-on Reset, $\overline{\text{MCLR}}$, WDT Reset, Brown-out Reset, $\overline{\text{MCLR}}$ Reset during Sleep and by the `RESET` instruction.

Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. Status bits from the RCON register, $\overline{\text{RI}}$, $\overline{\text{TO}}$, $\overline{\text{PD}}$, $\overline{\text{POR}}$ and $\overline{\text{BOR}}$, are set or cleared differently in different Reset situations, as indicated in Table 3-2. These bits are used in software to determine the nature of the Reset. See Table 3-3 for a full description of the Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The Enhanced MCU devices have a $\overline{\text{MCLR}}$ noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses. The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



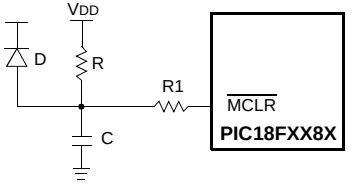
Note 1: This is a separate oscillator from the RC oscillator of the CLKI pin.
2: See Table 3-1 for time-out situations.

3.1 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected. To take advantage of the POR circuitry, tie the MCLR pin through a 1 kΩ to 10 kΩ resistor to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004). For a slow rise time, see Figure 3-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

FIGURE 3-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 = 1 kΩ to 10 kΩ will limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

3.2 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out (parameter #33) only on power-up from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip due to VDD, temperature and process variation. See DC parameter #33 for details.

3.3 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over (parameter #32). This ensures that the crystal oscillator or resonator has started and stabilized.

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

3.4 PLL Lock Time-out

With the PLL enabled, the time-out sequence following a Power-on Reset is different from other oscillator modes. A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the oscillator start-up time-out (OST).

3.5 Brown-out Reset (BOR)

A configuration bit, BOREN, can disable (if clear/programmed), or enable (if set) the Brown-out Reset circuitry. If VDD falls below parameter D005 for greater than parameter #35, the brown-out situation will reset the chip. A Reset may not occur if VDD falls below parameter D005 for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. If the Power-up Timer is enabled, it will be invoked after VDD rises above BVDD; it then will keep the chip in Reset for an additional time delay (parameter #33). If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute the additional time delay.

3.6 Time-out Sequence

On power-up, the time-out sequence is as follows: First, PWRT time-out is invoked after the POR time delay has expired. Then, OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 3-3, Figure 3-4, Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, the time-outs will expire if MCLR is kept low long enough. Bringing MCLR high will begin execution immediately (Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC18FXX8X device operating in parallel.

Table 3-2 shows the Reset conditions for some Special Function Registers while Table 3-3 shows the Reset conditions for all of the registers.

TABLE 3-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up[2]		Brown-out	Wake-up from Sleep or Oscillator Switch
	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1		
HS with PLL enabled[1]	72 ms + 1024 TOSC + 2ms	1024 TOSC + 2 ms	1024 TOSC + 2 ms	1024 TOSC + 2 ms
EC with PLL enabled[1]	72 ms + 2ms	1.5 μs + 2 ms	2 ms	1.5 μs + 2 ms
HS, XT, LP	72 ms + 1024 TOSC	1024 TOSC	1024 TOSC	1024 TOSC
EC	72 ms	1.5 μs	1.5 μs	1.5 μs[3]
External RC	72 ms	1.5 μs	1.5 μs	1.5 μs

Note 1: 2 ms is the nominal time required for the 4x PLL to lock.

2: 72 ms is the nominal power-up timer delay if implemented.

3: 1.5 μs is the recovery time from Sleep. There is no recovery time from oscillator switch.

REGISTER 3-1: RCON REGISTER BITS AND POSITIONS

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

Note: Refer to **Section 4.14 "RCON Register"** for bit definitions.

TABLE 3-2: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter	RCON Register	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	STKFUL	STKUNF
Power-on Reset	0000h	0--1 1100	1	1	1	0	0	u	u
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	0--u uuuu	u	u	u	u	u	u	u
Software Reset during normal operation	0000h	0--0 uuuu	0	u	u	u	u	u	u
Stack Full Reset during normal operation	0000h	0--u uu11	u	u	u	u	u	u	1
Stack Underflow Reset during normal operation	0000h	0--u uu11	u	u	u	u	u	1	u
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	0--u 10uu	u	1	0	u	u	u	u
WDT Reset	0000h	0--u 01uu	1	0	1	u	u	u	u
WDT Wake-up	PC + 2	u--u 00uu	u	0	0	u	u	u	u
Brown-out Reset	0000h	0--1 11u0	1	1	1	1	0	u	u
Interrupt wake-up from Sleep	PC + 2[1]	u--u 00uu	u	1	0	u	u	u	u

Legend: u = unchanged, x = unknown, – = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bits are set, the PC is loaded with the interrupt vector (000008h or 000018h).

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TOSU	PIC18F6X8X	PIC18F8X8X	---0 0000	---0 0000	---0 uuuu(3)
TOSH	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu(3)
TOSL	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu(3)
STKPTR	PIC18F6X8X	PIC18F8X8X	00-0 0000	uu-0 0000	uu-u uuuu(3)
PCLATU	PIC18F6X8X	PIC18F8X8X	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	PIC18F6X8X	PIC18F8X8X	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6X8X	PIC18F8X8X	0000 000x	0000 000x	uuuu uuuu(1)
INTCON2	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu(1)
INTCON3	PIC18F6X8X	PIC18F8X8X	1100 0000	1100 0000	uuuu uuuu(1)
INDF0	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTINC0	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTDEC0	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PREINC0	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PLUSW0	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
FSR0H	PIC18F6X8X	PIC18F8X8X	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTINC1	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTDEC1	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PREINC1	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PLUSW1	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
FSR1H	PIC18F6X8X	PIC18F8X8X	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6X8X	PIC18F8X8X	---- 0000	---- 0000	---- uuuu
INDF2	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTINC2	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
POSTDEC2	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PREINC2	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
PLUSW2	PIC18F6X8X	PIC18F8X8X	N/A	N/A	N/A
FSR2H	PIC18F6X8X	PIC18F8X8X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F6X8X	PIC18F8X8X	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F6X8X	PIC18F8X8X	0000 0000	uuuu uuuu	uuuu uuuu
TMR0L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6X8X	PIC18F8X8X	---- 0000	---- 0000	---- uuuu
LVDCON	PIC18F6X8X	PIC18F8X8X	--00 0101	--00 0101	--uu uuuu
WDTCON	PIC18F6X8X	PIC18F8X8X	---- ---0	---- ---0	---- ---u
RCON(4)	PIC18F6X8X	PIC18F8X8X	0--q 11qq	0--q qquu	u--u qquu
TMR1H	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	PIC18F6X8X	PIC18F8X8X	0-00 0000	u-uu uuuu	u-uu uuuu
TMR2	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
PR2	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	1111 1111
T2CON	PIC18F6X8X	PIC18F8X8X	-000 0000	-000 0000	-uuu uuuu
SSPBUF	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
SSPCON1	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
SSPCON2	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
ADRESH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F6X8X	PIC18F8X8X	--00 0000	--00 0000	--uu uuuu
ADCON1	PIC18F6X8X	PIC18F8X8X	--00 0000	--00 0000	--uu uuuu
ADCON2	PIC18F6X8X	PIC18F8X8X	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F6X8X	PIC18F8X8X	--00 0000	--00 0000	--uu uuuu
CCPAS1	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
CVRCON	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
CMCON	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TMR3H	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F6X8X	PIC18F8X8X	0000 0000	uuuu uuuu	uuuu uuuu
PSPCON	PIC18F6X8X	PIC18F8X8X	0000 ----	0000 ----	uuuu ----
SPBRG	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RCREG	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TXREG	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TXSTA	PIC18F6X8X	PIC18F8X8X	0000 0010	0000 0010	uuuu uuuu
RCSTA	PIC18F6X8X	PIC18F8X8X	0000 000x	0000 000x	uuuu uuuu
EEADRH	PIC18F6X8X	PIC18F8X8X	---- --00	---- --00	---- --uu
EEADR	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
EEDATA	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
EECON2	PIC18F6X8X	PIC18F8X8X	xx-0 x000	uu-0 u000	uu-0 u000
EECON1	PIC18F6X8X	PIC18F8X8X	00-0 x000	00-0 u000	uu-u uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
IPR3	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
PIE3	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F6X8X	PIC18F8X8X	-1-1 1111	-1-1 1111	-u-u uuuu
PIR2	PIC18F6X8X	PIC18F8X8X	-0-0 0000	-0-0 0000	-u-u uuuu(1)
PIE2	PIC18F6X8X	PIC18F8X8X	-0-0 0000	-0-0 0000	-u-u uuuu
IPR1	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu(1)
PIE1	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
MEMCON	PIC18F6X8X	PIC18F8X8X	0-00 --00	0-00 --00	u-uu --uu
TRISJ	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISH	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISG	PIC18F6X8X	PIC18F8X8X	---1 1111	---1 1111	---u uuuu
TRISF	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISE	PIC18F6X8X	PIC18F8X8X	0000 -111	0000 -111	uuuu -uuu
TRISD	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F6X8X	PIC18F8X8X	1111 1111	1111 1111	uuuu uuuu
TRISA(5,6)	PIC18F6X8X	PIC18F8X8X	-111 1111(5)	-111 1111(5)	-uuu uuuu(5)
LATJ	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATG	PIC18F6X8X	PIC18F8X8X	---x xxxx	---u uuuu	---u uuuu
LATF	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATE	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATD	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA(5,6)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx(5)	-uuu uuuu(5)	-uuu uuuu(5)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
PORTJ	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTH	PIC18F6X8X	PIC18F8X8X	0000 xxxx	0000 uuuu	uuuu uuuu
PORTG	PIC18F6X8X	PIC18F8X8X	--xx xxxx	--uu uuuu	--uu uuuu
PORTF	PIC18F6X8X	PIC18F8X8X	x000 0000	u000 0000	u000 0000
PORTE	PIC18F6X8X	PIC18F8X8X	---- -000	---- -000	---- -uuu
PORTD	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(5,6)	PIC18F6X8X	PIC18F8X8X	-x0x 0000(5)	-u0u 0000(5)	-uuu uuuu(5)
SPBRGH	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
BAUDCON	PIC18F6X8X	PIC18F8X8X	-1-0 0-00	-1-0 0-00	-u-u u-uu
ECCP1DEL	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
ECANCON	PIC18F6X8X	PIC18F8X8X	0001 0000	0001 0000	uuuu uuuu
TXERRCNT	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXERRCNT	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
COMSTAT	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
CIOCON	PIC18F6X8X	PIC18F8X8X	0000 ----	0000 ----	uuuu ----
BRGCON3	PIC18F6X8X	PIC18F8X8X	00-- -000	00-- -000	uu-- -uuu
BRGCON2	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
BRGCON1	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
CANCON	PIC18F6X8X	PIC18F8X8X	1000 000-	1000 000-	uuuu uuu-
CANSTAT	PIC18F6X8X	PIC18F8X8X	100- 000-	100- 000-	uuu- uuu-
RXB0D7	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D6	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D5	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D4	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D3	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D2	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D1	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D0	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0DLC	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
RXB0EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0SIDL	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB0SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0CON	PIC18F6X8X	PIC18F8X8X	000- 0000	000- 0000	uuu- uuuu
RXB1D7	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D6	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D5	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D4	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D3	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D2	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D1	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D0	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1DLC	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
RXB1EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1SIDL	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB1SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1CON	PIC18F6X8X	PIC18F8X8X	000- 0000	000- 0000	uuu- uuuu
TXB0D7	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D6	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D5	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D4	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D3	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D2	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D1	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D0	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0DLC	PIC18F6X8X	PIC18F8X8X	-x-- xxxx	-u-- uuuu	-u-- uuuu
TXB0EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
TXB0SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TXB0SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0CON	PIC18F6X8X	PIC18F8X8X	0000 0-00	0000 0-00	uuuu u-uu
TXB1D7	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D6	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D5	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D4	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D3	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D2	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D1	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D0	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1DLC	PIC18F6X8X	PIC18F8X8X	-x-- xxxx	-u-- uuuu	-u-- uuuu
TXB1EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- uu-u
TXB1SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
TXB1CON	PIC18F6X8X	PIC18F8X8X	0000 0-00	0000 0-00	uuuu u-uu
TXB2D7	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D6	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D5	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D4	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D3	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D2	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D1	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2D0	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	0uuu uuuu
TXB2DLC	PIC18F6X8X	PIC18F8X8X	-x-- xxxx	-u-- uuuu	-u-- uuuu
TXB2EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
TXB2SIDH	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
TXB2CON	PIC18F6X8X	PIC18F8X8X	0000 0-00	0000 0-00	uuuu u-uu
RXM1EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
RXM1EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXM1SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXM1SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXM0EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXM0EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXM0SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXM0SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF5EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF5EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF5SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF5SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF4EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF4EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF4SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF4SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF3EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF3EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF3SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF3SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF2EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF2EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF2SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF2SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF1EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF1EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF1SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF1SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF0EIDL	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF0EIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF0SIDL	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF0SIDH	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
B5D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B5EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B5SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B5SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B5CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
B4D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B4EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B4SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B4CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
B3D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
B3D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B3EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B3SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B3CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
B2D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B2EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B2SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
B1D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
B1D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B1EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B1SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
B0D7(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D6(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D5(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D4(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D3(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D2(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D1(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D0(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0DLC(7)	PIC18F6X8X	PIC18F8X8X	-xxx xxxx	-uuu uuuu	-uuu uuuu
B0EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx x-xx	uuuu u-uu	uuuu u-uu
B0SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0CON(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
TXBIE(7)	PIC18F6X8X	PIC18F8X8X	---0 00--	---u uu--	---u uu--
BIE0(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
BSEL0(7)	PIC18F6X8X	PIC18F8X8X	0000 00--	0000 00--	uuuu uu--
MSEL3(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
MSEL2(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
MSEL1(7)	PIC18F6X8X	PIC18F8X8X	0000 0101	0000 0101	uuuu uuuu
MSEL0(7)	PIC18F6X8X	PIC18F8X8X	0101 0000	0101 0000	uuuu uuuu
SDFLC(7)	PIC18F6X8X	PIC18F8X8X	---0 0000	---0 0000	-u-- uuuu
RXFCON1(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
RXFCON0(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON7(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON6(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON5(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON4(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON3(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON2(7)	PIC18F6X8X	PIC18F8X8X	0001 0001	0001 0001	uuuu uuuu
RXFBCON1(7)	PIC18F6X8X	PIC18F8X8X	0001 0001	0001 0001	uuuu uuuu
RXFBCON0(7)	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXF15EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF15EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF15SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF15SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF14SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF13SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF12SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF11SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF10EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF10EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
RXF10SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	-uuu uuuu
RXF10SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF9EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF9EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF9SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	-uuu uuuu
RXF9SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF8EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF8EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF8SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	-uuu uuuu
RXF8SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF7EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF7EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF7SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	-uuu uuuu
RXF7SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF6EIDL(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF6EIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF6SIDL(7)	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	-uuu uuuu
RXF6SIDH(7)	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.

7: This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

FIGURE 3-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD} VIA 1 kΩ RESISTOR)

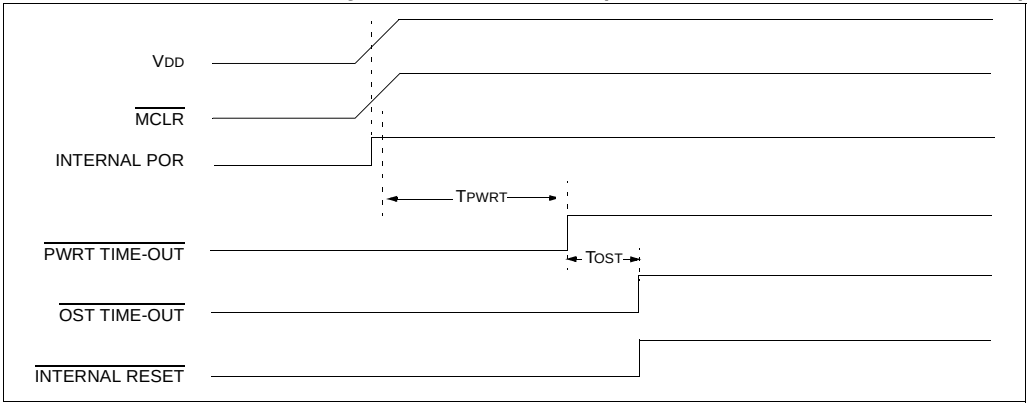


FIGURE 3-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 1

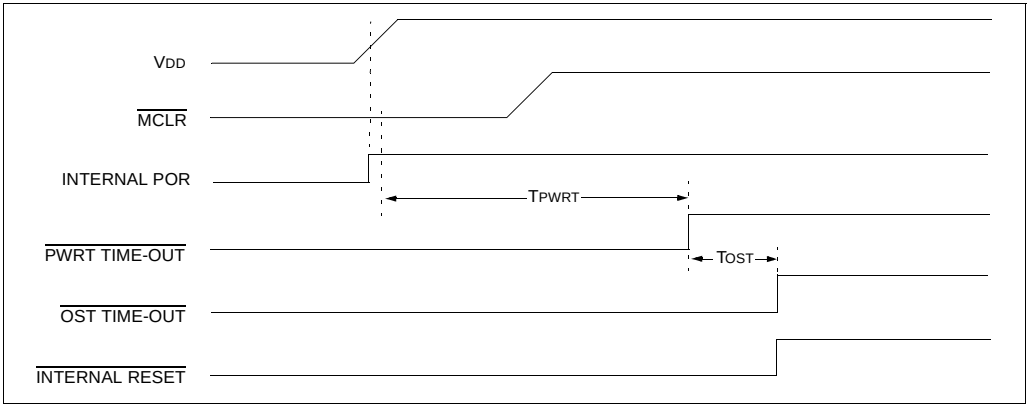


FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 2

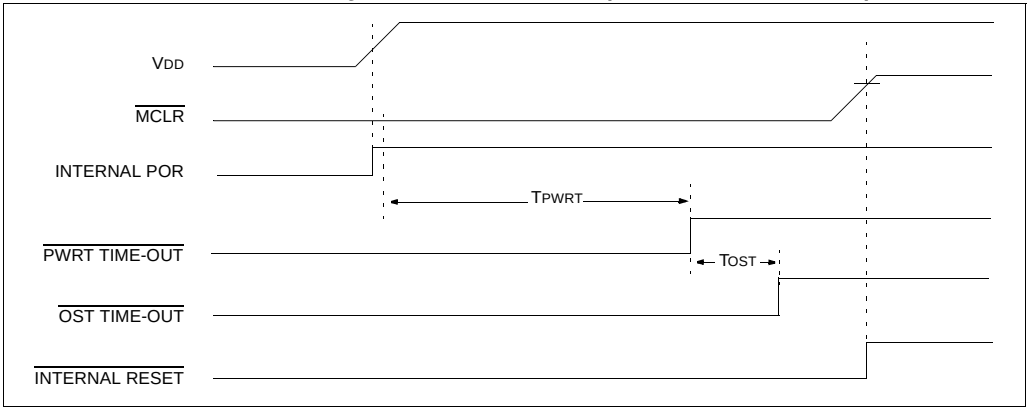


FIGURE 3-6: SLOW RISE TIME (MCLR TIED TO VDD VIA 1 kΩ RESISTOR)

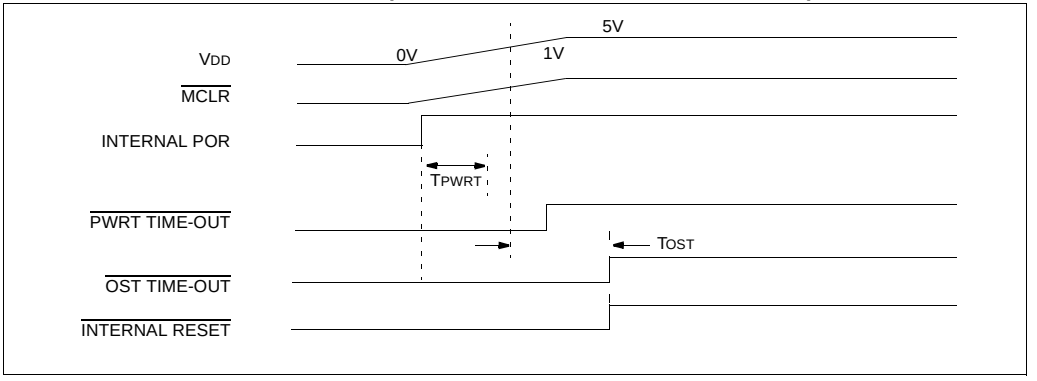
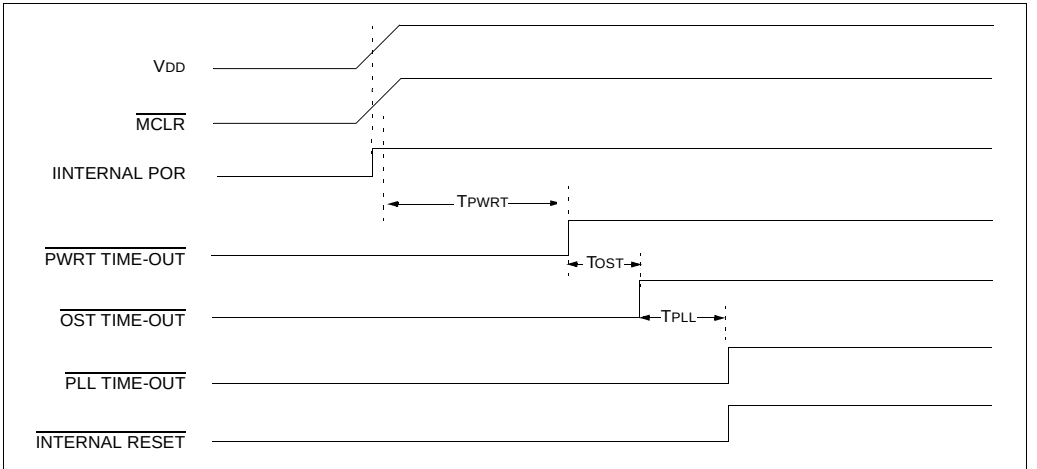


FIGURE 3-7: TIME-OUT SEQUENCE ON POR W/ PLL ENABLED (MCLR TIED TO VDD VIA 1 kΩ RESISTOR)



Note: TOST = 1024 clock cycles.
TPLL ≈ 2 ms max. First three stages of the PWRT timer.

4.0 MEMORY ORGANIZATION

There are three memory blocks in PIC18F6585/8585/6680/8680 devices. They are:

- Program Memory
- Data RAM
- Data EEPROM

Data and program memory use separate busses which allows for concurrent access of these blocks. Additional detailed information for Flash program memory and data EEPROM is provided in **Section 5.0 "Flash Program Memory"** and **Section 7.0 "Data EEPROM Memory"**, respectively.

In addition to on-chip Flash, the PIC18F8X8X devices are also capable of accessing external program memory through an external memory bus. Depending on the selected operating mode (discussed in **Section 4.1.1 "PIC18F8X8X Program Memory Modes"**), the controllers may access either internal or external program memory exclusively, or both internal and external memory in selected blocks. Additional information on the external memory interface is provided in **Section 6.0 "External Memory Interface"**.

4.1 Program Memory Organization

A 21-bit program counter is capable of addressing the 2-Mbyte program memory space. Accessing a location between the physically implemented memory and the 2-Mbyte address will cause a read of all '0's (a NOP instruction).

The PIC18F6585 and PIC18F8585 each have 48 Kbytes of on-chip Flash memory, while the PIC18F6680 and PIC18F8680 have 64 Kbytes of Flash. This means that PIC18FX585 devices can store internally up to 24,576 single-word instructions and PIC18FX680 devices can store up to 32,768 single-word instructions.

The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

Figure 4-1 shows the program memory map for PIC18F6585/8585 devices while Figure 4-2 shows the program memory map for PIC18F6680/8680 devices.

4.1.1 PIC18F8X8X PROGRAM MEMORY MODES

PIC18F8X8X devices differ significantly from their PIC18 predecessors in their utilization of program memory. In addition to available on-chip Flash program memory, these controllers can also address up to 2 Mbytes of external program memory through the external memory interface. There are four distinct operating modes available to the controllers:

- Microprocessor (MP)
- Microprocessor with Boot Block (MPBB)
- Extended Microcontroller (EMC)
- Microcontroller (MC)

The Program Memory mode is determined by setting the two Least Significant bits of the CONFIG3L configuration byte, as shown in Register 4-1. (See also **Section 24.1 "Configuration Bits"** for additional details on the device configuration bits.)

The Program Memory modes operate as follows:

- The **Microprocessor Mode** permits access only to external program memory; the contents of the on-chip Flash memory are ignored. The 21-bit program counter permits access to a 2-MByte linear program memory space.
- The **Microprocessor with Boot Block Mode** accesses on-chip Flash memory from addresses 000000h to 0007FFh. Above this, external program memory is accessed all the way up to the 2-MByte limit. Program execution automatically switches between the two memories as required.
- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the physical limit of the on-chip Flash (0BFFFh for the PIC18F8585, 0FFFFh for the PIC18F8680) causes a read of all '0's (a NOP instruction). The Microcontroller mode is the only operating mode available to PIC18F6X8X devices.
- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip Flash memory; above this, the device accesses external program memory up to the 2-MByte program space limit. As with Boot Block mode, execution automatically switches between the two memories as required.

In all modes, the microcontroller has complete access to data RAM and EEPROM.

Figure 4-3 compares the memory maps of the different Program Memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 4-1.

FIGURE 4-1: INTERNAL PROGRAM MEMORY MAP AND STACK FOR PIC18F6585/8585

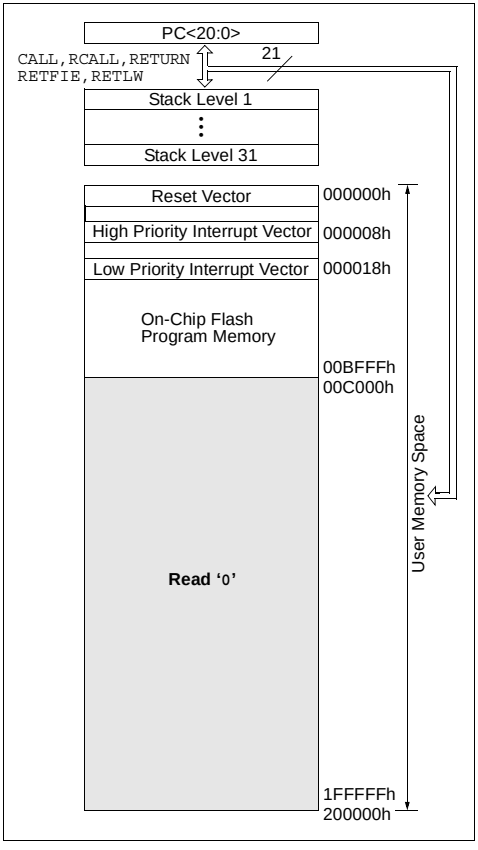


FIGURE 4-2: INTERNAL PROGRAM MEMORY MAP AND STACK FOR PIC18F6680/8680

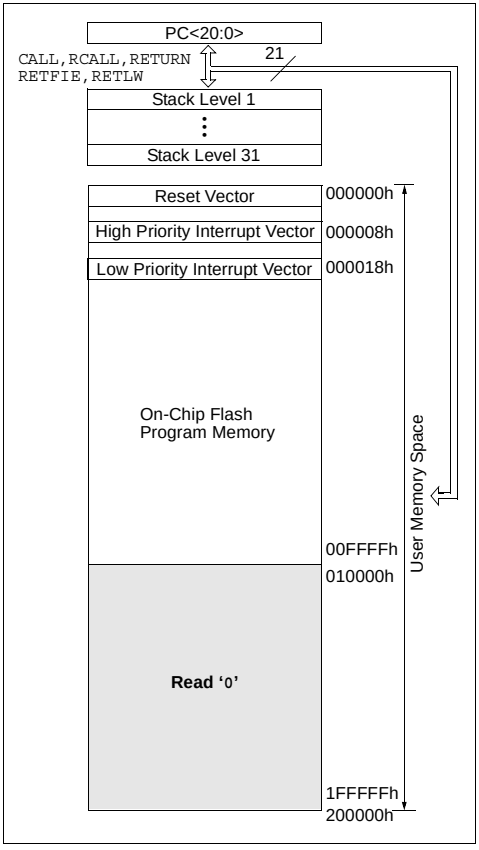


TABLE 4-1: MEMORY ACCESS FOR PIC18F8X8X PROGRAM MEMORY MODES

Operating Mode	Internal Program Memory			External Program Memory		
	Execution From	Table Read From	Table Write To	Execution From	Table Read From	Table Write To
Microprocessor	No Access	No Access	No Access	Yes	Yes	Yes
Microprocessor w/ Boot Block	Yes	Yes	Yes	Yes	Yes	Yes
Microcontroller	Yes	Yes	Yes	No Access	No Access	No Access
Extended Microcontroller	Yes	Yes	Yes	Yes	Yes	Yes

REGISTER 4-1: CONFIG3L CONFIGURATION BYTE

R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
WAIT	—	—	—	—	—	PM1	PM0
bit 7							bit 0

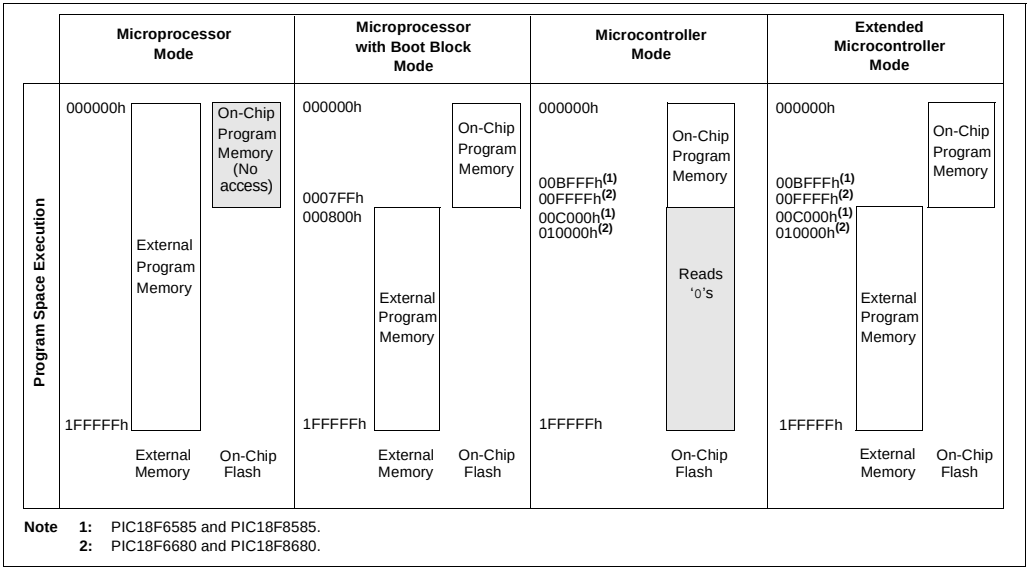
bit 7 **WAIT:** External Bus Data Wait Enable bit
1 = Wait selections unavailable, device will not wait
0 = Wait programmed by WAIT1 and WAIT0 bits of MEMCOM register (MEMCOM<5:4>)

bit 6-2 **Unimplemented:** Read as '0'

bit 1-0 **PM1:PM0:** Processor Data Memory Mode Select bits
11 = Microcontroller mode
10 = Microprocessor mode
01 = Microcontroller with Boot Block mode
00 = Extended Microcontroller mode

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'	
- n = Value after erase	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

FIGURE 4-3: MEMORY MAPS FOR PIC18F8X8X PROGRAM MEMORY MODES



Note 1: PIC18F6585 and PIC18F8585.
2: PIC18F6680 and PIC18F8680.

4.2 Return Address Stack

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a CALL or RCALL instruction is executed or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW, or a RETFIE instruction. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit stack pointer, with the stack pointer initialized to 00000b after all Resets. There is no RAM associated with stack pointer 00000b. This is only a Reset value. During a CALL type instruction causing a push onto the stack, the stack pointer is first incremented and the RAM location pointed to by the stack pointer is written with the contents of the PC. During a RETURN type instruction causing a pop from the stack, the contents of the RAM location pointed to by the STKPTR are transferred to the PC and then the stack pointer is decremented.

The stack space is not part of either program or data space. The stack pointer is readable and writable and the address on the top of the stack is readable and writable through SFR registers. Data can also be pushed to or popped from the stack, using the top-of-stack SFRs. Status bits indicate if the stack pointer is at or beyond the 31 levels provided.

4.2.1 TOP-OF-STACK ACCESS

The top of the stack is readable and writable. Three register locations, TOSU, TOSH and TOSL, hold the contents of the stack location pointed to by the STKPTR register. This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU, TOSH and TOSL registers. These values can be placed on a user defined software stack. At return time, the software can replace the TOSU, TOSH and TOSL and do a return.

The user must disable the global interrupt enable bits during this time to prevent inadvertent stack operations.

4.2.2 RETURN STACK POINTER (STKPTR)

The STKPTR register contains the stack pointer value, the STKFUL (Stack Full) status bit, and the STKUNF (Stack Underflow) status bits. Register 4-2 shows the STKPTR register. The value of the stack pointer can be 0 through 31. The stack pointer increments when values are pushed onto the stack and decrements when values are popped off the stack. At Reset, the stack pointer value will be '0'. The user may read and write the stack pointer value. This feature can be used by a Real-Time Operating System for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit can only be cleared in software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) configuration bit. Refer to **Section 25.0 "Instruction Set Summary"** for a description of the device configuration bits. If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the stack pointer will be set to '0'.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the stack pointer will increment to 31. Any additional pushes will not overwrite the 31st push and STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and sets the STKUNF bit while the stack pointer remains at '0'. The STKUNF bit will remain set until cleared in software or a POR occurs.

Note:	Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken.

REGISTER 4-2: STKPTR REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

bit 7 **STKFUL:** Stack Full Flag bit
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = Stack underflow occurred
0 = Stack underflow did not occur

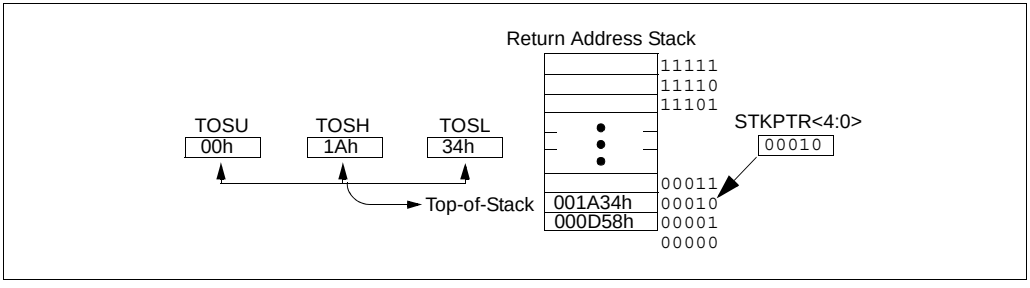
bit 5 **Unimplemented:** Read as '0'

bit 4-0 **SP4:SP0:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 can only be cleared in user software or by a POR.

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

FIGURE 4-4: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



4.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable option. To push the current PC value onto the stack, a PUSH instruction can be executed. This will increment the stack pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place a return address on the stack.

The ability to pull the TOS value off of the stack and replace it with the value that was previously pushed onto the stack, without disturbing normal execution, is achieved by using the POP instruction. The POP instruction discards the current TOS by decrementing the stack pointer. The previous value pushed onto the stack then becomes the TOS value.

4.2.4 STACK FULL/UNDERFLOW RESETS

These Resets are enabled by programming the STVREN configuration bit. When the STVREN bit is disabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. When the STVREN bit is enabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. The STKFUL or STKUNF bits are only cleared by the user software or a POR Reset.

4.3 Fast Register Stack

A "fast interrupt return" option is available for interrupts. A fast register stack is provided for the Status, WREG and BSR registers and is only one in depth. The stack is not readable or writable and is loaded with the current value of the corresponding register when the processor vectors for an interrupt. The values in the registers are then loaded back into the working registers if the FAST RETURN instruction is used to return from the interrupt.

A low or high priority interrupt source will push values into the stack registers. If both low and high priority interrupts are enabled, the stack registers cannot be used reliably for low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten.

If high priority interrupts are not disabled during low priority interrupts, users must save the key registers in software during a low priority interrupt.

If no interrupts are used, the fast register stack can be used to restore the Status, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a FAST CALL instruction must be executed.

Example 4-1 shows a source code example that uses the fast register stack.

EXAMPLE 4-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL  SUB1, FAST    ;STATUS, WREG, BSR
                    ;SAVED IN FAST REGISTER
                    ;STACK
        •
        •
SUB1    •
        •
        •
      RETURN FAST   ;RESTORE VALUES SAVED
                    ;IN FAST REGISTER STACK
```

4.4 PCL, PCLATH and PCLATU

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide. The low byte is called the PCL register; this register is readable and writable. The high byte is called the PCH register. This register contains the PC<15:8> bits and is not directly readable or writable; updates to the PCH register may be performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits and is not directly readable or writable; updates to the PCU register may be performed through the PCLATU register.

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the LSB of the PCL is fixed to a value of '0'. The PC increments by 2 to address sequential instructions in the program memory.

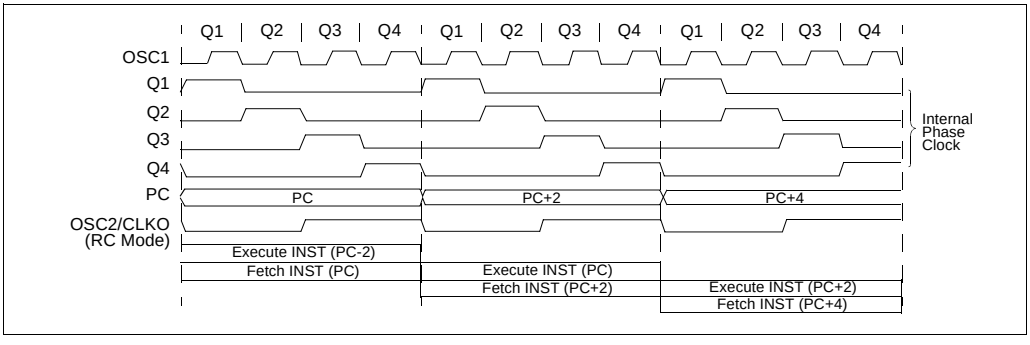
The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

The contents of PCLATH and PCLATU will be transferred to the program counter by an operation that writes PCL. Similarly, the upper two bytes of the program counter will be transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 4.8.1 "Computed GOTO"**).

4.5 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 4-5.

FIGURE 4-5: CLOCK/INSTRUCTION CYCLE



4.6 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 4-2).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

EXAMPLE 4-2: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
1. MOVLW 55h	Fetch 1	Execute 1				
2. MOVWF PORTB		Fetch 2	Execute 2			
3. BRA SUB_1			Fetch 3	Execute 3		
4. BSF PORTA, 3 (Forced NOP)				Fetch 4	Flush (NOP)	
5. Instruction @ address SUB_1					Fetch SUB_1	Execute SUB_1

All instructions are single cycle except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

4.7 Instructions in Program Memory

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte (LSB) of an instruction word is always stored in a program memory location with an even address (LSB = 0). Figure 4-6 shows an example of how instruction words are stored in the program memory. To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 4.4 "PCL, PCLATH and PCLATU"**).

The CALL and GOTO instructions have an absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1> which accesses the desired byte address in program memory. Instruction #2 in Figure 4-6 shows how the instruction "GOTO 000006h" is encoded in the program memory. Program branch instructions which encode a relative address offset operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 25.0 "Instruction Set Summary"** provides further details of the instruction set.

FIGURE 4-6: INSTRUCTIONS IN PROGRAM MEMORY

			Program Memory Byte Locations →	LSB = 1	LSB = 0	Word Address ↓
						000000h
						000002h
						000004h
						000006h
Instruction 1:	MOVLW	055h		0Fh	55h	000008h
Instruction 2:	GOTO	000006h		0EFh	03h	00000Ah
				0F0h	00h	00000Ch
Instruction 3:	MOVFF	123h, 456h		0C1h	23h	00000Eh
				0F4h	56h	000010h
						000012h
						000014h

4.7.1 TWO-WORD INSTRUCTIONS

The PIC18F6585/8585/6680/8680 devices have four two-word instructions: MOVFF, CALL, GOTO and LFSR. The second word of these instructions has the 4 MSBs set to '1's and is a special kind of NOP instruction. The lower 12 bits of the second word contain data to be used by the instruction. If the first word of the instruction is executed, the data in the second word is accessed. If the second word of the instruction is executed by itself (first word was skipped), it will execute as a NOP. This action is necessary when the two-word instruction is preceded by a conditional instruction that changes the PC. A program example that demonstrates this concept is shown in Example 4-3. Refer to **Section 25.0 "Instruction Set Summary"** for further details of the instruction set.

EXAMPLE 4-3: TWO-WORD INSTRUCTIONS

CASE 1:

Object Code	Source Code	
0110 0110 0000 0000	TSTFSZ REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2	; No, execute 2-word instruction
1111 0100 0101 0110		; 2nd operand holds address of REG2
0010 0100 0000 0000	ADDWF REG3	; continue code

CASE 2:

Object Code	Source Code	
0110 0110 0000 0000	TSTFSZ REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2	; Yes
1111 0100 0101 0110		; 2nd operand becomes NOP
0010 0100 0000 0000	ADDWF REG3	; continue code

4.8 Look-up Tables

Look-up tables are implemented two ways. These are:

- Computed GOTO
- Table Reads

4.8.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL).

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW 0xnn instructions. WREG is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW 0xnn instructions that returns the value 0xnn to the calling function.

The offset value (value in WREG) specifies the number of bytes that the program counter should advance.

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

4.8.2 TABLE READS/TABLE WRITES

A better method of storing data in program memory allows 2 bytes of data to be stored in each instruction location.

Look-up table data may be stored 2 bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from, or written to program memory. Data is transferred to/from program memory, one byte at a time.

A description of the table read/table write operation is shown in **Section 5.0 "Flash Program Memory"**.

4.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. Figure 4-7 shows the data memory organization for the PIC18F6585/8585/6680/8680 devices.

The data memory map is divided into 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper 4 bits for the BSR are not implemented.

The data memory contains Special Function Registers (SFR) and General Purpose Registers (GPR). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratch pad operations in the user's application. The SFRs start at the last location of Bank 15 (0FFFh) and extend downwards. Any remaining space beyond the SFRs in the Bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as '0's.

The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of a File Select Register (FSRn) and a corresponding Indirect File Operand (INDFn). Each FSR holds a 12-bit address value that can be used to access any location in the data memory map without banking.

The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction that moves a value from one register to another.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM. **Section 4.10 "Access Bank"** provides a detailed description of the Access RAM.

4.9.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly. Indirect addressing operates using a File Select Register and corresponding Indirect File Operand. The operation of indirect addressing is shown in **Section 4.12 "Indirect Addressing, INDF and FSR Registers"**.

Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

Data RAM is available for use as general purpose registers by all instructions. The top section of Bank 15 (0F60h to 0FFFh) contains SFRs. All other banks of data memory contain GPR registers, starting with Bank 0.

4.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) 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 4-2 and Table 4-3.

The SFRs can be classified into two sets: those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature. The SFRs are typically distributed among the peripherals whose functions they control.

The unused SFR locations are unimplemented and read as '0's. The addresses for the SFRs are listed in Table 4-2.

FIGURE 4-7: DATA MEMORY MAP FOR PIC18FXX80/XX85 DEVICES

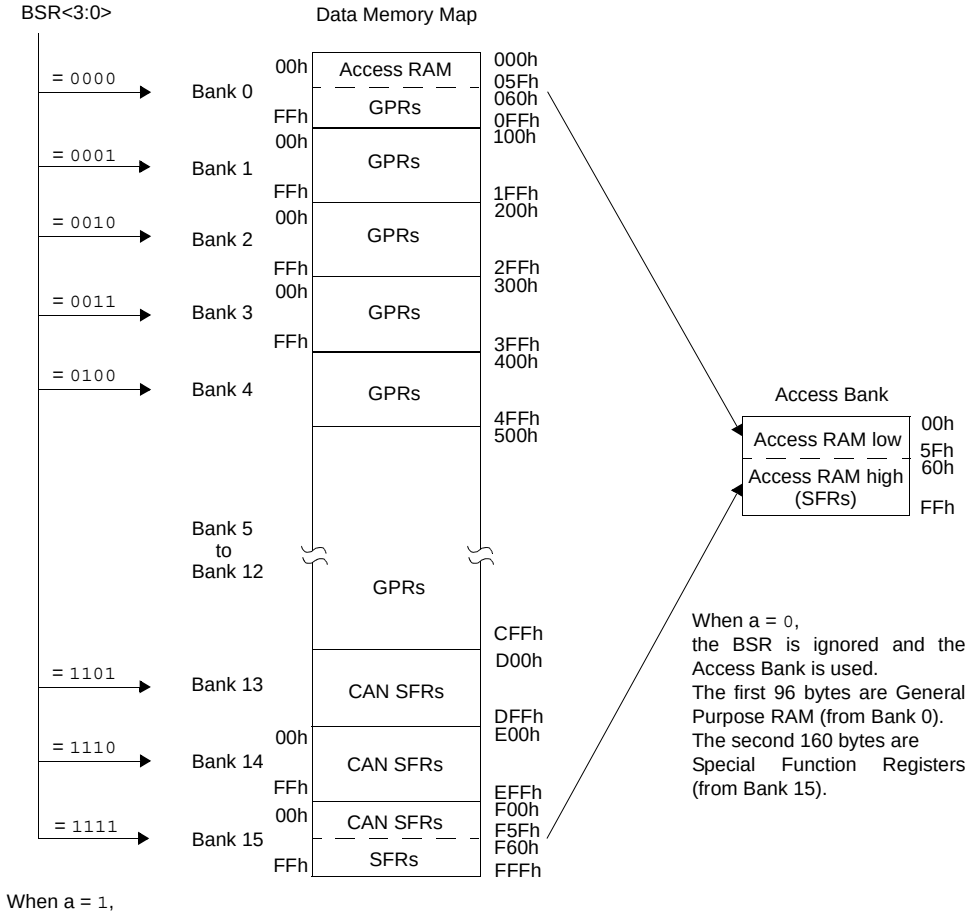


TABLE 4-2: SPECIAL FUNCTION REGISTER MAP

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(3)	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2(3)	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(3)	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2(3)	FBCh	CCPR2H	F9Ch	MEMCON(2)
FFBh	PCLATU	FDBh	PLUSW2(3)	FBBh	CCPR2L	F9Bh	—(1)
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	TRISJ(2)
FF9h	PCL	FD9h	FSR2L	FB9h	—(1)	F99h	TRISH(2)
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—(1)	F98h	TRISG
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—(1)	F97h	TRISF
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD
FF4h	PRODH	FD4h	—(1)	FB4h	CMCON	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	LATJ(2)
FF0h	INTCON3	FD0h	RCON	FB0h	PSPCON	F90h	LATH(2)
FEFh	INDF0(3)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	LATG
FEEh	POSTINC0(3)	FCEh	TMR1L	FAEh	RCREG	F8Eh	LATF
FEDh	POSTDEC0(3)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE
FECh	PREINC0(3)	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD
FEBh	PLUSW0(3)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	EEADRH	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	PORTJ(2)
FE7h	INDF1(3)	FC7h	SSPSTAT	FA7h	EECON2	F87h	PORTH(2)
FE6h	POSTINC1(3)	FC6h	SSPCON1	FA6h	EECON1	F86h	PORTG
FE5h	POSTDEC1(3)	FC5h	SSPCON2	FA5h	IPR3	F85h	PORTF
FE4h	PREINC1(3)	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE
FE3h	PLUSW1(3)	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
F7Fh	SPBRGH	F5Fh	CANCON_RO0	F3Fh	CANCON_RO2	F1Fh	RXM1EIDL
F7Eh	BAUDCON	F5Eh	CANSTAT_RO0	F3Eh	CANSTAT_RO2	F1Eh	RXM1EIDH
F7Dh	—(1)	F5Dh	RXB1D7	F3Dh	TXB1D7	F1Dh	RXM1SIDL
F7Ch	—(1)	F5Ch	RXB1D6	F3Ch	TXB1D6	F1Ch	RXM1SIDH
F7Bh	—(1)	F5Bh	RXB1D5	F3Bh	TXB1D5	F1Bh	RXM0EIDL
F7Ah	—(1)	F5Ah	RXB1D4	F3Ah	TXB1D4	F1Ah	RXM0EIDH
F79h	ECCP1DEL	F59h	RXB1D3	F39h	TXB1D3	F19h	RXM0SIDL
F78h	—(1)	F58h	RXB1D2	F38h	TXB1D2	F18h	RXM0SIDH
F77h	ECANCON	F57h	RXB1D1	F37h	TXB1D1	F17h	RXF5EIDL
F76h	TXERRCNT	F56h	RXB1D0	F36h	TXB1D0	F16h	RXF5EIDH
F75h	RXERRCNT	F55h	RXB1DLC	F35h	TXB1DLC	F15h	RXF5SIDL
F74h	COMSTAT	F54h	RXB1EIDL	F34h	TXB1EIDL	F14h	RXF5SIDH
F73h	CIOCON	F53h	RXB1EIDH	F33h	TXB1EIDH	F13h	RXF4EIDL
F72h	BRGCON3	F52h	RXB1SIDL	F32h	TXB1SIDL	F12h	RXF4EIDH
F71h	BRGCON2	F51h	RXB1SIDH	F31h	TXB1SIDH	F11h	RXF4SIDL
F70h	BRGCON1	F50h	RXB1CON	F30h	TXB1CON	F10h	RXF4SIDH
F6Fh	CANCON	F4Fh	CANCON_RO1	F2Fh	CANCON_RO3	F0Fh	RXF3EIDL
F6Eh	CANSTAT	F4Eh	CANSTAT_RO1	F2Eh	CANSTAT_RO3	F0Eh	RXF3EIDH
F6Dh	RXB0D7	F4Dh	TXB0D7	F2Dh	TXB2D7	F0Dh	RXF3SIDL
F6Ch	RXB0D6	F4Ch	TXB0D6	F2Ch	TXB2D6	F0Ch	RXF3SIDH
F6Bh	RXB0D5	F4Bh	TXB0D5	F2Bh	TXB2D5	F0Bh	RXF2EIDL
F6Ah	RXB0D4	F4Ah	TXB0D4	F2Ah	TXB2D4	F0Ah	RXF2EIDH
F69h	RXB0D3	F49h	TXB0D3	F29h	TXB2D3	F09h	RXF2SIDL
F68h	RXB0D2	F48h	TXB0D2	F28h	TXB2D2	F08h	RXF2SIDH
F67h	RXB0D1	F47h	TXB0D1	F27h	TXB2D1	F07h	RXF1EIDL
F66h	RXB0D0	F46h	TXB0D0	F26h	TXB2D0	F06h	RXF1EIDH
F65h	RXB0DLC	F45h	TXB0DLC	F25h	TXB2DLC	F05h	RXF1SIDL
F64h	RXB0EIDL	F44h	TXB0EIDL	F24h	TXB2EIDL	F04h	RXF1SIDH
F63h	RXB0EIDH	F43h	TXB0EIDH	F23h	TXB2EIDH	F03h	RXF0EIDL
F62h	RXB0SIDL	F42h	TXB0SIDL	F22h	TXB2SIDL	F02h	RXF0EIDH
F61h	RXB0SIDH	F41h	TXB0SIDH	F21h	TXB2SIDH	F01h	RXF0SIDL
F60h	RXB0CON	F40h	TXB0CON	F20h	TXB2CON	F00h	RXF0SIDH

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
EFFh	—(1)	EDFh	—(1)	EBFh	—(1)	E9Fh	—(1)
EFEh	—(1)	EDEh	—(1)	EBEh	—(1)	E9Eh	—(1)
EFDh	—(1)	EDDh	—(1)	EBDh	—(1)	E9Dh	—(1)
EFCh	—(1)	EDCh	—(1)	EBCh	—(1)	E9Ch	—(1)
EFBh	—(1)	EDBh	—(1)	EBBh	—(1)	E9Bh	—(1)
EFAh	—(1)	EDAh	—(1)	EBAh	—(1)	E9Ah	—(1)
EF9h	—(1)	ED9h	—(1)	EB9h	—(1)	E99h	—(1)
EF8h	—(1)	ED8h	—(1)	EB8h	—(1)	E98h	—(1)
EF7h	—(1)	ED7h	—(1)	EB7h	—(1)	E97h	—(1)
EF6h	—(1)	ED6h	—(1)	EB6h	—(1)	E96h	—(1)
EF5h	—(1)	ED5h	—(1)	EB5h	—(1)	E95h	—(1)
EF4h	—(1)	ED4h	—(1)	EB4h	—(1)	E94h	—(1)
EF3h	—(1)	ED3h	—(1)	EB3h	—(1)	E93h	—(1)
EF2h	—(1)	ED2h	—(1)	EB2h	—(1)	E92h	—(1)
EF1h	—(1)	ED1h	—(1)	EB1h	—(1)	E91h	—(1)
EF0h	—(1)	ED0h	—(1)	EB0h	—(1)	E90h	—(1)
EEFh	—(1)	ECFh	—(1)	EAFh	—(1)	E8Fh	—(1)
EEEh	—(1)	ECEh	—(1)	EAEh	—(1)	E8Eh	—(1)
EEDh	—(1)	ECDh	—(1)	EADh	—(1)	E8Dh	—(1)
EECh	—(1)	ECCh	—(1)	EACh	—(1)	E8Ch	—(1)
EEBh	—(1)	ECBh	—(1)	EABh	—(1)	E8Bh	—(1)
EEAh	—(1)	ECAh	—(1)	EAAh	—(1)	E8Ah	—(1)
EE9h	—(1)	EC9h	—(1)	EA9h	—(1)	E89h	—(1)
EE8h	—(1)	EC8h	—(1)	EA8h	—(1)	E88h	—(1)
EE7h	—(1)	EC7h	—(1)	EA7h	—(1)	E87h	—(1)
EE6h	—(1)	EC6h	—(1)	EA6h	—(1)	E86h	—(1)
EE5h	—(1)	EC5h	—(1)	EA5h	—(1)	E85h	—(1)
EE4h	—(1)	EC4h	—(1)	EA4h	—(1)	E84h	—(1)
EE3h	—(1)	EC3h	—(1)	EA3h	—(1)	E83h	—(1)
EE2h	—(1)	EC2h	—(1)	EA2h	—(1)	E82h	—(1)
EE1h	—(1)	EC1h	—(1)	EA1h	—(1)	E81h	—(1)
EE0h	—(1)	EC0h	—(1)	EA0h	—(1)	E80h	—(1)

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
E7Fh	CANCON_RO4	E5Fh	CANCON_RO6	E3Fh	CANCON_RO8	E1Fh	—(1)
E7Eh	CANSTAT_RO4	E5Eh	CANSTAT_RO6	E3Eh	CANSTAT_RO8	E1Eh	—(1)
E7Dh	B5D7	E5Dh	B3D7	E3Dh	B1D7	E1Dh	—(1)
E7Ch	B5D6	E5Ch	B3D6	E3Ch	B1D6	E1Ch	—(1)
E7Bh	B5D5	E5Bh	B3D5	E3Bh	B1D5	E1Bh	—(1)
E7Ah	B5D4	E5Ah	B3D4	E3Ah	B1D4	E1Ah	—(1)
E79h	B5D3	E59h	B3D3	E39h	B1D3	E19h	—(1)
E78h	B5D2	E58h	B3D2	E38h	B1D2	E18h	—(1)
E77h	B5D1	E57h	B3D1	E37h	B1D1	E17h	—(1)
E76h	B5D0	E56h	B3D0	E36h	B1D0	E16h	—(1)
E75h	B5DLC	E55h	B3DLC	E35h	B1DLC	E15h	—(1)
E74h	B5EIDL	E54h	B3EIDL	E34h	B1EIDL	E14h	—(1)
E73h	B5EIDH	E53h	B3EIDH	E33h	B1EIDH	E13h	—(1)
E72h	B5SIDL	E52h	B3SIDL	E32h	B1SIDL	E12h	—(1)
E71h	B5SIDH	E51h	B3SIDH	E31h	B1SIDH	E11h	—(1)
E70h	B5CON	E50h	B3CON	E30h	B1CON	E10h	—(1)
E6Fh	CANCON_RO5	E4Fh	CANCON_RO7	E2Fh	CANCON_RO9	E0Fh	—(1)
E6Eh	CANSTAT_RO5	E4Eh	CANSTAT_RO7	E2Eh	CANSTAT_RO9	E0Eh	—(1)
E6Dh	B4D7	E4Dh	B2D7	E2Dh	B0D7	E0Dh	—(1)
E6Ch	B4D6	E4Ch	B2D6	E2Ch	B0D6	E0Ch	—(1)
E6Bh	B4D5	E4Bh	B2D5	E2Bh	B0D5	E0Bh	—(1)
E6Ah	B4D4	E4Ah	B2D4	E2Ah	B0D4	E0Ah	—(1)
E69h	B4D3	E49h	B2D3	E29h	B0D3	E09h	—(1)
E68h	B4D2	E48h	B2D2	E28h	B0D2	E08h	—(1)
E67h	B4D1	E47h	B2D1	E27h	B0D1	E07h	—(1)
E66h	B4D0	E46h	B2D0	E26h	B0D0	E06h	—(1)
E65h	B4DLC	E45h	B2DLC	E25h	B0DLC	E05h	—(1)
E64h	B4EIDL	E44h	B2EIDL	E24h	B0EIDL	E04h	—(1)
E63h	B4EIDH	E43h	B2EIDH	E23h	B0EIDH	E03h	—(1)
E62h	B4SIDL	E42h	B2SIDL	E22h	B0SIDL	E02h	—(1)
E61h	B4SIDH	E41h	B2SIDH	E21h	B0SIDH	E01h	—(1)
E60h	B4CON	E40h	B2CON	E20h	B0CON	E00h	—(1)

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
DFFh	—(1)	DDFh	—(1)	DBFh	—(1)	D9Fh	—(1)
DFEh	—(1)	DDEh	—(1)	DBEh	—(1)	D9Eh	—(1)
DFDh	—(1)	DDDh	—(1)	DBDh	—(1)	D9Dh	—(1)
DFCh	TXBIE	DDCh	—(1)	DBCh	—(1)	D9Ch	—(1)
DFBh	—(1)	DDBh	—(1)	DBBh	—(1)	D9Bh	—(1)
DFAh	BIE0	DDAh	—(1)	DBAh	—(1)	D9Ah	—(1)
DF9h	—(1)	DD9h	—(1)	DB9h	—(1)	D99h	—(1)
DF8h	BSEL0	DD8h	SDFLC	DB8h	—(1)	D98h	—(1)
DF7h	—(1)	DD7h	—(1)	DB7h	—(1)	D97h	—(1)
DF6h	—(1)	DD6h	—(1)	DB6h	—(1)	D96h	—(1)
DF5h	—(1)	DD5h	RXFCON1	DB5h	—(1)	D95h	—(1)
DF4h	—(1)	DD4h	RXFCON0	DB4h	—(1)	D94h	—(1)
DF3h	MSEL3	DD3h	—(1)	DB3h	—(1)	D93h	RXF15EIDL
DF2h	MSEL2	DD2h	—(1)	DB2h	—(1)	D92h	RXF15EIDH
DF1h	MSEL1	DD1h	—(1)	DB1h	—(1)	D91h	RXF15SIDL
DF0h	MSEL0	DD0h	—(1)	DB0h	—(1)	D90h	RXF15SIDH
DEFh	—(1)	DCFh	—(1)	DAFh	—(1)	D8Fh	—(1)
DEEh	—(1)	DCEh	—(1)	DAEh	—(1)	D8Eh	—(1)
DEDh	—(1)	DCDh	—(1)	DADh	—(1)	D8Dh	—(1)
DECh	—(1)	DCCh	—(1)	DACh	—(1)	D8Ch	—(1)
DEBh	—(1)	DCBh	—(1)	DABh	—(1)	D8Bh	RXF14EIDL
DEAh	—(1)	DCAh	—(1)	DAAh	—(1)	D8Ah	RXF14EIDH
DE9h	—(1)	DC9h	—(1)	DA9h	—(1)	D89h	RXF14SIDL
DE8h	—(1)	DC8h	—(1)	DA8h	—(1)	D88h	RXF14SIDH
DE7h	RXFBCON7	DC7h	—(1)	DA7h	—(1)	D87h	RXF13EIDL
DE6h	RXFBCON6	DC6h	—(1)	DA6h	—(1)	D86h	RXF13EIDH
DE5h	RXFBCON5	DC5h	—(1)	DA5h	—(1)	D85h	RXF13SIDL
DE4h	RXFBCON4	DC4h	—(1)	DA4h	—(1)	D84h	RXF13SIDH
DE3h	RXFBCON3	DC3h	—(1)	DA3h	—(1)	D83h	RXF12EIDL
DE2h	RXFBCON2	DC2h	—(1)	DA2h	—(1)	D82h	RXF12EIDH
DE1h	RXFBCON1	DC1h	—(1)	DA1h	—(1)	D81h	RXF12SIDL
DE0h	RXFBCON0	DC0h	—(1)	DA0h	—(1)	D80h	RXF12SIDH

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name
D7Fh	—(1)
D7Eh	—(1)
D7Dh	—(1)
D7Ch	—(1)
D7Bh	RXF11EIDL
D7Ah	RXF11EIDH
D79h	RXF11SIDL
D78h	RXF11SIDH
D77h	RXF10EIDL
D76h	RXF10EIDH
D75h	RXF10SIDL
D74h	RXF10SIDH
D73h	RXF9EIDL
D72h	RXF9EIDH
D71h	RXF9SIDL
D70h	RXF9SIDH
D6Fh	—(1)
D6Eh	—(1)
D6Dh	—(1)
D6Ch	—(1)
D6Bh	RXF8EIDL
D6Ah	RXF8EIDH
D69h	RXF8SIDL
D68h	RXF8SIDH
D67h	RXF7EIDL
D66h	RXF7EIDH
D65h	RXF7SIDL
D64h	RXF7SIDH
D63h	RXF6EIDL
D62h	RXF6EIDH
D61h	RXF6SIDL
D60h	RXF6SIDH

Note 1: Unimplemented registers are read as '0'.
2: This register is not available on PIC18F6X8X devices.
3: This is not a physical register.

TABLE 4-3: REGISTER FILE SUMMARY

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
TOSU	—	—	—	Top-of-Stack Upper Byte (TOS<20:16>)					---0 0000	36, 54
TOSH	Top-of-Stack High Byte (TOS<15:8>)								0000 0000	36, 54
TOSL	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000	36, 54
STKPTR	STKFUL	STKUNF	—	Return Stack Pointer					00-0 0000	36, 55
PCLATU	—	—	bit 21	Holding Register for PC<20:16>					--00 0000	36, 56
PCLATH	Holding Register for PC<15:8>								0000 0000	36, 56
PCL	PC Low Byte (PC<7:0>)								0000 0000	36, 56
TBLPTRU	—	—	bit 21(2)	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000	36, 86
TBLPTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000	36, 86
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000	36, 86
TABLAT	Program Memory Table Latch								0000 0000	36, 86
PRODH	Product Register High Byte								xxxx xxxx	36, 107
PRODL	Product Register Low Byte								xxxx xxxx	36, 107
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	36, 111
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	36, 112
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	36, 113
INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								n/a	79
POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								n/a	79
POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								n/a	79
PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								n/a	79
PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by value in WREG								n/a	79
FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High Byte				---- 0000	36, 79
FSR0L	Indirect Data Memory Address Pointer 0 Low Byte								xxxx xxxx	36, 79
WREG	Working Register								xxxx xxxx	36
INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)								n/a	79
POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)								n/a	79
POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)								n/a	79
PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)								n/a	79
PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by value in WREG								n/a	79
FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1 High Byte				---- 0000	37, 79
FSR1L	Indirect Data Memory Address Pointer 1 Low Byte								xxxx xxxx	37, 79
BSR	—	—	—	—	Bank Select Register				---- 0000	37, 78
INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)								n/a	79
POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)								n/a	79
POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)								n/a	79
PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)								n/a	79
PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by value in WREG								n/a	79
FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High Byte				---- 0000	37, 79
FSR2L	Indirect Data Memory Address Pointer 2 Low Byte								xxxx xxxx	37, 79

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	37, 81
TMR0H	Timer0 Register High Byte								0000 0000	37, 157
TMR0L	Timer0 Register Low Byte								xxxx xxxx	37, 157
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	37, 155
OSCCON	—	—	—	—	LOCK	PLLEN	SCS1	SCS	---- 0000	27, 37
LVDCON	—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	37, 271
WDTCON	—	—	—	—	—	—	—	SWDTE	---- ---0	37, 355
RCON	IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0--1 11qq	37, 82, 123
TMR1H	Timer1 Register High Byte								xxxx xxxx	37, 159
TMR1L	Timer1 Register Low Byte								xxxx xxxx	37, 159
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0-00 0000	37, 159
TMR2	Timer2 Register								0000 0000	37, 162
PR2	Timer2 Period Register								1111 1111	37, 163
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	37, 162
SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	37, 189
SSPADD	SSP Address Register in I^2C Slave mode. SSP Baud Rate Reload Register in I^2C Master mode.								0000 0000	37, 198
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	37, 199
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	37, 191
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	37, 201
ADRESH	A/D Result Register High Byte								xxxx xxxx	38, 257
ADRESL	A/D Result Register Low Byte								xxxx xxxx	38, 257
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	38, 249
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	38, 257
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	38, 251
CCPR1H	Enhanced Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	38, 173
CCPR1L	Enhanced Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	38, 172
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	38, 172
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	38, 172
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	38, 172
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	38, 172
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	38, 172
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	38, 265
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	38, 259
TMR3H	Timer3 Register High Byte								xxxx xxxx	38, 164
TMR3L	Timer3 Register Low Byte								xxxx xxxx	38, 164
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	0000 0000	38, 164
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	38, 153

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{MCLR}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
SPBRG	USART Baud Rate Generator								0000 0000	38, 239
RCREG	USART Receive Register								0000 0000	38, 241
TXREG	USART Transmit Register								0000 0000	38, 239
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	38, 230
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	38, 231
EEADRH	—	—	—	—	—	—	EE Adr Register High		---- --00	38, 105
EEADR	Data EEPROM Address Register								0000 0000	38, 105
EEDATA	Data EEPROM Data Register								0000 0000	38, 105
EECON2	Data EEPROM Control Register 2 (not a physical register)								---- ----	38, 105
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	00-0 x000	38, 102
IPR3	IRXIP	WAKIP	ERRIP	TXB2IP/ TXBnIP	TXB1IP	TXB0IP	RXB1IP/ RXBnIP	RXB0IP/ FIFOWMIP	1111 1111	39, 122
PIR3	IRXIF	WAKIF	ERRIF	TXB2IF/ TXBnIF	TXB1IF	TXB0IF	RXB1IF/ RXBnIF	RXB0IF/ FIFOWMIF	0000 0000	39, 116
PIE3	IRXIE	WAKIE	ERRIE	TXB2IE/ TXBnIE	TXB1IE	TXB0IE	RXB1IE/ RXBnIE	RXB0IE/ FIFOWMIE	0000 0000	39, 119
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-1 1111	39, 121
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-0 0000	39, 115
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-0 0000	39, 118
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0111 1111	39, 120
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	39, 114
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	39, 117
MEMCON(3)	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	39, 94
TRISJ(3)	Data Direction Control Register for PORTJ								1111 1111	39, 151
TRISH(3)	Data Direction Control Register for PORTH								1111 1111	39, 148
TRISG	—	—	—	Data Direction Control Register for PORTG					---1 1111	39, 145
TRISF	Data Direction Control Register for PORTF								1111 1111	39, 141
TRISE	Data Direction Control Register for PORTE								1111 1111	39, 138
TRISD	Data Direction Control Register for PORTD								1111 1111	39, 135
TRISC	Data Direction Control Register for PORTC								1111 1111	39, 131
TRISB	Data Direction Control Register for PORTB								1111 1111	39, 128
TRISA	—	TRISA6(1)	Data Direction Control Register for PORTA						-111 1111	39, 125
LATJ(3)	Read PORTJ Data Latch, Write PORTJ Data Latch								xxxx xxxx	39, 151
LATH(3)	Read PORTH Data Latch, Write PORTH Data Latch								xxxx xxxx	39, 148
LATG	—	—	—	Read PORTG Data Latch, Write PORTG Data Latch					---x xxxx	39, 145
LATF	Read PORTF Data Latch, Write PORTF Data Latch								xxxx xxxx	39, 141
LATE	Read PORTE Data Latch, Write PORTE Data Latch								xxxx xxxx	39, 138
LATD	Read PORTD Data Latch, Write PORTD Data Latch								xxxx xxxx	39, 133
LATC	Read PORTC Data Latch, Write PORTC Data Latch								xxxx xxxx	39, 131
LATB	Read PORTB Data Latch, Write PORTB Data Latch								xxxx xxxx	39, 128
LATA	—	LATA6(1)	Read PORTA Data Latch, Write PORTA Data Latch(1)						-xxx xxxx	39, 125

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
PORTJ(3)	Read PORTJ pins, Write PORTJ Data Latch								xxxx xxxx	40, 151
PORTH(3)	Read PORTH pins, Write PORTH Data Latch								xxxx xxxx	40, 148
PORTG	—	—	RG5(6)	Read PORTG pins, Write PORTG Data Latch					--0x xxxx	40, 145
PORTF	Read PORTF pins, Write PORTF Data Latch								xxxx xxxx	40, 141
PORTE	Read PORTE pins, Write PORTE Data Latch								xxxx xxxx	40, 136
PORTD	Read PORTD pins, Write PORTD Data Latch								xxxx xxxx	40, 133
PORTC	Read PORTC pins, Write PORTC Data Latch								xxxx xxxx	40, 131
PORTB	Read PORTB pins, Write PORTB Data Latch								xxxx xxxx	40, 128
PORTA	—	RA6(1)	Read PORTA pins, Write PORTA Data Latch(1)						-x0x 0000	40, 125
SPBRGH	Enhanced USART Baud Rate Generator High Byte								0000 0000	40, 233
BAUDCON	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-0 0-00	40, 233
ECCP1DEL	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	40, 187
TXERRCNT	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	0000 0000	40, 288
RXERRCNT	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	0000 0000	40, 296
COMSTAT Mode 0	RXB0OVFL	RXB1OVFL	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN	0000 0000	40, 284
COMSTAT Mode 1	—	RXBnOVFL	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN	-000 0000	40, 284
COMSTAT Mode 2	$\overline{\text{FIFOEMPTY}}$	RXBnOVFL	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN	0000 0000	40, 284
CIOCON	TX2SRC	TX2EN	ENDRHI	CANCAP	—	—	—	—	0000 ----	40, 318
BRGCON3	WAKDIS	WAKFIL	—	—	—	SEG2PH2	SEG2PH1	SEG2PH0	00-- -000	40, 317
BRGCON2	SEG2PHT	SAM	SEG1PH2	SEG1PH1	SEG1PH0	PRSEG2	PRSEG1	PRSEG0	0000 0000	40, 317
BRGCON1	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	0000 0000	40, 317
CANCON Mode 0	REQOP2	REQOP1	REQOP0	ABAT	WIN2	WIN1	WIN0	—	1000 000-	40, 239
CANCON Mode 1	REQOP2	REQOP1	REQOP0	ABAT	—	—	—	—	1000 ----	40, 239
CANCON Mode 2	REQOP2	REQOP1	REQOP0	ABAT	FP3	FP2	FP1	FP0	1000 0000	40, 239
CANSTAT Mode 0	OPMODE2	OPMODE1	OPMODE0	—	ICODE2	ICODE1	ICODE0	—	000- 0000	40, 239
CANSTAT Modes 0, 1	OPMODE2	OPMODE1	OPMODE0	EICODE4	EICODE3	EICODE2	EICODE1	EICODE0	0000 0000	40, 239
ECANCON	MDSEL1	MDSEL0	FIFOWM	EWIN4	EWIN3	EWIN2	EWIN1	EWIN0	0001 0000	40, 323
RXB0D7	RXB0D77	RXB0D76	RXB0D75	RXB0D74	RXB0D73	RXB0D72	RXB0D71	RXB0D70	xxxx xxxx	40, 230
RXB0D6	RXB0D67	RXB0D66	RXB0D65	RXB0D64	RXB0D63	RXB0D62	RXB0D61	RXB0D60	xxxx xxxx	40, 230
RXB0D5	RXB0D57	RXB0D56	RXB0D55	RXB0D54	RXB0D53	RXB0D52	RXB0D51	RXB0D50	xxxx xxxx	40, 230
RXB0D4	RXB0D47	RXB0D46	RXB0D45	RXB0D44	RXB0D43	RXB0D42	RXB0D41	RXB0D40	xxxx xxxx	40, 230
RXB0D3	RXB0D37	RXB0D36	RXB0D35	RXB0D34	RXB0D33	RXB0D32	RXB0D31	RXB0D30	xxxx xxxx	40, 230
RXB0D2	RXB0D27	RXB0D26	RXB0D25	RXB0D24	RXB0D23	RXB0D22	RXB0D21	RXB0D20	xxxx xxxx	40, 230
RXB0D1	RXB0D17	RXB0D16	RXB0D15	RXB0D14	RXB0D13	RXB0D12	RXB0D11	RXB0D10	xxxx xxxx	40, 230
RXB0D0	RXB0D07	RXB0D06	RXB0D05	RXB0D04	RXB0D03	RXB0D02	RXB0D01	RXB0D00	xxxx xxxx	40, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
RXB0DLC	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	40, 230
RXB0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	41, 230
RXB0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	41, 230
RXB0SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	41, 230
RXB0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	41, 230
RXB0CON Mode 0	RXFUL	RXM1	RXM0(4)	—(4)	RXRTRR0(4)	RXB0DBEN(4)	JTOFF(4)	FILHIT0(4)	000- 0000	41, 230
RXB0CON Mode 1, 2	RXFUL	RXM1	RTRR0(4)	FILHIT4(4)	FILHIT3(4)	FILHIT2(4)	FILHIT1(4)	FILHIT0(4)	0000 0000	41, 230
RXB1D7	RXB1D77	RXB1D76	RXB1D75	RXB1D74	RXB1D73	RXB1D72	RXB1D71	RXB1D70	xxxx xxxx	41, 230
RXB1D6	RXB1D67	RXB1D66	RXB1D65	RXB1D64	RXB1D63	RXB1D62	RXB1D61	RXB1D60	xxxx xxxx	41, 230
RXB1D5	RXB1D57	RXB1D56	RXB1D55	RXB1D54	RXB1D53	RXB1D52	RXB1D51	RXB1D50	xxxx xxxx	41, 230
RXB1D4	RXB1D47	RXB1D46	RXB1D45	RXB1D44	RXB1D43	RXB1D42	RXB1D41	RXB1D40	xxxx xxxx	41, 230
RXB1D3	RXB1D37	RXB1D36	RXB1D35	RXB1D34	RXB1D33	RXB1D32	RXB1D31	RXB1D30	xxxx xxxx	41, 230
RXB1D2	RXB1D27	RXB1D26	RXB1D25	RXB1D24	RXB1D23	RXB1D22	RXB1D21	RXB1D20	xxxx xxxx	41, 230
RXB1D1	RXB1D17	RXB1D16	RXB1D15	RXB1D14	RXB1D13	RXB1D12	RXB1D11	RXB1D10	xxxx xxxx	41, 230
RXB1D0	RXB1D07	RXB1D06	RXB1D05	RXB1D04	RXB1D03	RXB1D02	RXB1D01	RXB1D00	xxxx xxxx	41, 230
RXB1DLC	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	41, 230
RXB1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	41, 230
RXB1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	41, 230
RXB1SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	41, 230
RXB1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	41, 230
RXB1CON Mode 0	RXFUL	RXM1	RXM0(4)	—(4)	RXRTRR0(4)	FILHIT2(4)	FILHIT1(4)	FILHIT0(4)	000- 0000	41, 230
RXB1CON Mode 1, 2	RXFUL	RXM1	RTRRO(4)	FILHIT4(4)	FILHIT3(4)	FILHIT2(4)	FILHIT1(4)	FILHIT0(4)	0000 0000	41, 230
TXB0D7	TXB0D77	TXB0D76	TXB0D75	TXB0D74	TXB0D73	TXB0D72	TXB0D71	TXB0D70	xxxx xxxx	41, 230
TXB0D6	TXB0D67	TXB0D66	TXB0D65	TXB0D64	TXB0D63	TXB0D62	TXB0D61	TXB0D60	xxxx xxxx	41, 230
TXB0D5	TXB0D57	TXB0D56	TXB0D55	TXB0D54	TXB0D53	TXB0D52	TXB0D51	TXB0D50	xxxx xxxx	41, 230
TXB0D4	TXB0D47	TXB0D46	TXB0D45	TXB0D44	TXB0D43	TXB0D42	TXB0D41	TXB0D40	xxxx xxxx	41, 230
TXB0D3	TXB0D37	TXB0D36	TXB0D35	TXB0D34	TXB0D33	TXB0D32	TXB0D31	TXB0D30	xxxx xxxx	41, 230
TXB0D2	TXB0D27	TXB0D26	TXB0D25	TXB0D24	TXB0D23	TXB0D22	TXB0D21	TXB0D20	xxxx xxxx	41, 230
TXB0D1	TXB0D17	TXB0D16	TXB0D15	TXB0D14	TXB0D13	TXB0D12	TXB0D11	TXB0D10	xxxx xxxx	41, 230
TXB0D0	TXB0D07	TXB0D06	TXB0D05	TXB0D04	TXB0D03	TXB0D02	TXB0D01	TXB0D00	xxxx xxxx	41, 230
TXB0DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	41, 230
TXB0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	41, 230
TXB0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	41, 230
TXB0SIDL	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xx-x x-xx	41, 230
TXB0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	42, 230
TXB0CON Mode 0	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	-000 0-00	42, 230
TXB0CON Mode 1, 2	TXBIF	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	0000 0-00	42, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
TXB1D7	TXB1D77	TXB1D76	TXB1D75	TXB1D74	TXB1D73	TXB1D72	TXB1D71	TXB1D70	xxxx xxxx	42, 230
TXB1D6	TXB1D67	TXB1D66	TXB1D65	TXB1D64	TXB1D63	TXB1D62	TXB1D61	TXB1D60	xxxx xxxx	42, 230
TXB1D5	TXB1D57	TXB1D56	TXB1D55	TXB1D54	TXB1D53	TXB1D52	TXB1D51	TXB1D50	xxxx xxxx	42, 230
TXB1D4	TXB1D47	TXB1D46	TXB1D45	TXB1D44	TXB1D43	TXB1D42	TXB1D41	TXB1D40	xxxx xxxx	42, 230
TXB1D3	TXB1D37	TXB1D36	TXB1D35	TXB1D34	TXB1D33	TXB1D32	TXB1D31	TXB1D30	xxxx xxxx	42, 230
TXB1D2	TXB1D27	TXB1D26	TXB1D25	TXB1D24	TXB1D23	TXB1D22	TXB1D21	TXB1D20	xxxx xxxx	42, 230
TXB1D1	TXB1D17	TXB1D16	TXB1D15	TXB1D14	TXB1D13	TXB1D12	TXB1D11	TXB1D10	xxxx xxxx	42, 230
TXB1D0	TXB1D07	TXB1D06	TXB1D05	TXB1D04	TXB1D03	TXB1D02	TXB1D01	TXB1D00	xxxx xxxx	42, 230
TXB1DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	42, 230
TXB1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	42, 230
TXB1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	42, 230
TXB1SIDL	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xx-x x-xx	42, 230
TXB1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	42, 230
TXB1CON Mode 0	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	-000 0-00	42, 230
TXB1CON Mode 1, 2	TXBIF	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	0000 0-00	42, 230
TXB2D7	TXB2D77	TXB2D76	TXB2D75	TXB2D74	TXB2D73	TXB2D72	TXB2D71	TXB2D70	xxxx xxxx	42, 230
TXB2D6	TXB2D67	TXB2D66	TXB2D65	TXB2D64	TXB2D63	TXB2D62	TXB2D61	TXB2D60	xxxx xxxx	42, 230
TXB2D5	TXB2D57	TXB2D56	TXB2D55	TXB2D54	TXB2D53	TXB2D52	TXB2D51	TXB2D50	xxxx xxxx	42, 230
TXB2D4	TXB2D47	TXB2D46	TXB2D45	TXB2D44	TXB2D43	TXB2D42	TXB2D41	TXB2D40	xxxx xxxx	42, 230
TXB2D3	TXB2D37	TXB2D36	TXB2D35	TXB2D34	TXB2D33	TXB2D32	TXB2D31	TXB2D30	xxxx xxxx	42, 230
TXB2D2	TXB2D27	TXB2D26	TXB2D25	TXB2D24	TXB2D23	TXB2D22	TXB2D21	TXB2D20	xxxx xxxx	42, 230
TXB2D1	TXB2D17	TXB2D16	TXB2D15	TXB2D14	TXB2D13	TXB2D12	TXB2D11	TXB2D10	xxxx xxxx	42, 230
TXB2D0	TXB2D07	TXB2D06	TXB2D05	TXB2D04	TXB2D03	TXB2D02	TXB2D01	TXB2D00	xxxx xxxx	42, 230
TXB2DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	42, 230
TXB2EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	42, 230
TXB2EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	42, 230
TXB2SIDL	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	42, 230
TXB2SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	42, 230
TXB2CON Mode 0	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	-000 0-00	42, 230
TXB2CON Mode 1, 2	TXBIF	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	0000 0-00	42, 230
RXM1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	42, 230
RXM1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXM1SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x 0-xx	43, 230
RXM1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXM0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXM0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXM0SIDL	SID2	SID1	SID0	—	EXIDM	—	EID17	EID16	xx-x 0-xx	43, 230
RXM0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF15EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
RXF15EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF15SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	47, 230
RXF15SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	47, 230
RXF14EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF14EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF14SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	47, 230
RXF14SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	47, 230
RXF13EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF13EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF13SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	47, 230
RXF13SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	47, 230
RXF12EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF12EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF12SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	47, 230
RXF12SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	47, 230
RXF11EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF11EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF11SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	47, 230
RXF11SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	47, 230
RXF10EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF10EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	47, 230
RXF10SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	48, 230
RXF10SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	48, 230
RXF9EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	47, 230
RXF9EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	48, 230
RXF9SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	48, 230
RXF9SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	48, 230
RXF8EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	48, 230
RXF8EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	48, 230
RXF8SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	48, 230
RXF8SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	48, 230
RXF7EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	48, 230
RXF7EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	48, 230
RXF7SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	48, 230
RXF7SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	48, 230
RXF6EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	48, 230
RXF6EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	48, 230
RXF6SIDL(7)	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	48, 230
RXF6SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	48, 230
RXF5EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF5EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
RXF5SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF5SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF4EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF4EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXF4SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF4SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF3EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF3EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXF3SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF3SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF2EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF2EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXF2SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF2SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXF1SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
RXF0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	43, 230
RXF0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	43, 230
RXF0SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xx-x x-xx	43, 230
RXF0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	43, 230
B5D7(7)	B5D77	B5D76	B5D75	B5D74	B5D73	B5D72	B5D71	B5D70	xxxx xxxx	44, 230
B5D6(7)	B5D67	B5D66	B5D65	B5D64	B5D63	B5D62	B5D61	B5D60	xxxx xxxx	44, 230
B5D5(7)	B5D57	B5D56	B5D55	B5D54	B5D53	B5D52	B5D51	B5D50	xxxx xxxx	44, 230
B5D4(7)	B5D47	B5D46	B5D45	B5D44	B5D43	B5D42	B5D41	B5D40	xxxx xxxx	44, 230
B5D3(7)	B5D37	B5D36	B5D35	B5D34	B5D33	B5D32	B5D31	B5D30	xxxx xxxx	44, 230
B5D2(7)	B5D27	B5D26	B5D25	B5D24	B5D23	B5D22	B5D21	B5D20	xxxx xxxx	44, 230
B5D1(7)	B5D17	B5D16	B5D15	B5D14	B5D13	B5D12	B5D11	B5D10	xxxx xxxx	44, 230
B5D0(7)	B5D07	B5D06	B5D05	B5D04	B5D03	B5D02	B5D01	B5D00	xxxx xxxx	44, 230
B5DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	44, 230
B5EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	44, 230
B5EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	44, 230
B5SIDL(7)	SID2	SID1	SID0	SRR	EXID/ EXIDE(5)	—	EID17	EID16	xxxx x-xx	44, 230
B5SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	44, 230
B5CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	44, 230
B4D7(7)	B4D77	B4D76	B4D75	B4D74	B4D73	B4D72	B4D71	B4D70	xxxx xxxx	44, 230
B4D6(7)	B4D67	B4D66	B4D65	B4D64	B4D63	B4D62	B4D61	B4D60	xxxx xxxx	44, 230
B4D5(7)	B4D57	B4D56	B4D55	B4D54	B4D53	B4D52	B4D51	B4D50	xxxx xxxx	44, 230
B4D4(7)	B4D47	B4D46	B4D45	B4D44	B4D43	B4D42	B4D41	B4D40	xxxx xxxx	44, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
B4D3(7)	B4D37	B4D36	B4D35	B4D34	B4D33	B4D32	B4D31	B4D30	xxxx xxxx	44, 230
B4D2(7)	B4D27	B4D26	B4D25	B4D24	B4D23	B4D22	B4D21	B4D20	xxxx xxxx	44, 230
B4D1(7)	B4D17	B4D16	B4D15	B4D14	B4D13	B4D12	B4D11	B4D10	xxxx xxxx	44, 230
B4D0(7)	B4D07	B4D06	B4D05	B4D04	B4D03	B4D02	B4D01	B4D00	xxxx xxxx	44, 230
B4DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	44, 230
B4EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	44, 230
B4EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	44, 230
B4SIDL(7)	SID2	SID1	SID0	SRR	EXID/ EXIDE(5)	—	EID17	EID16	xxxx x-xx	44, 230
B4SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	44, 230
B4CON(5, 7)	RXFUL/ TXB3IF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	44, 230
B3D7(7)	B3D77	B3D76	B3D75	B3D74	B3D73	B3D72	B3D71	B3D70	xxxx xxxx	44, 230
B3D6(7)	B3D67	B3D66	B3D65	B3D64	B3D63	B3D62	B3D61	B3D60	xxxx xxxx	44, 230
B3D5(7)	B3D57	B3D56	B3D55	B3D54	B3D53	B3D52	B3D51	B3D50	xxxx xxxx	44, 230
B3D4(7)	B3D47	B3D46	B3D45	B3D44	B3D43	B3D42	B3D41	B3D40	xxxx xxxx	45, 230
B3D3(7)	B3D37	B3D36	B3D35	B3D34	B3D33	B3D32	B3D31	B3D30	xxxx xxxx	45, 230
B3D2(7)	B3D27	B3D26	B3D25	B3D24	B3D23	B3D22	B3D21	B3D20	xxxx xxxx	45, 230
B3D1(7)	B3D17	B3D16	B3D15	B3D14	B3D13	B3D12	B3D11	B3D10	xxxx xxxx	45, 230
B3D0(7)	B3D07	B3D06	B3D05	B3D04	B3D03	B3D02	B3D01	B3D00	xxxx xxxx	45, 230
B3DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	45, 230
B3EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	45, 230
B3EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	45, 230
B3SIDL(7)	SID2	SID1	SID0	SRR	EXID/ EXIDE(5)	—	EID17	EID16	xxxx x-xx	45, 230
B3SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	45, 230
B3CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	45, 230
B2D7(7)	B2D77	B2D76	B2D75	B2D74	B2D73	B2D72	B2D71	B2D70	xxxx xxxx	45, 230
B2D6(7)	B2D67	B2D66	B2D65	B2D64	B2D63	B2D62	B2D61	B2D60	xxxx xxxx	45, 230
B2D5(7)	B2D57	B2D56	B2D55	B2D54	B2D53	B2D52	B2D51	B2D50	xxxx xxxx	45, 230
B2D4(7)	B2D47	B2D46	B2D45	B2D44	B2D43	B2D42	B2D41	B2D40	xxxx xxxx	45, 230
B2D3(7)	B2D37	B2D36	B2D35	B2D34	B2D33	B2D32	B2D31	B2D30	xxxx xxxx	45, 230
B2D2(7)	B2D27	B2D26	B2D25	B2D24	B2D23	B2D22	B2D21	B2D20	xxxx xxxx	45, 230
B2D1(7)	B2D17	B2D16	B2D15	B2D14	B2D13	B2D12	B2D11	B2D10	xxxx xxxx	45, 230
B2D0(7)	B2D07	B2D06	B2D05	B2D04	B2D03	B2D02	B2D01	B2D00	xxxx xxxx	45, 230
B2DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	45, 230
B2EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	45, 230
B2EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	45, 230
B2SIDL(7)	SID2	SID1	SID0	SRR	EXID/ EXIDE(5)	—	EID17	EID16	xxxx x-xx	45, 230
B2SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	45, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
B2CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	45, 230
B1D7(7)	B1D77	B1D76	B1D75	B1D74	B1D73	B1D72	B1D71	B1D70	xxxx xxxx	45, 230
B1D6(7)	B1D67	B1D66	B1D65	B1D64	B1D63	B1D62	B1D61	B1D60	xxxx xxxx	45, 230
B1D5(7)	B1D57	B1D56	B1D55	B1D54	B1D53	B1D52	B1D51	B1D50	xxxx xxxx	45, 230
B1D4(7)	B1D47	B1D46	B1D45	B1D44	B1D43	B1D42	B1D41	B1D40	xxxx xxxx	45, 230
B1D3(7)	B1D37	B1D36	B1D35	B1D34	B1D33	B1D32	B1D31	B1D30	xxxx xxxx	45, 230
B1D2(7)	B1D27	B1D26	B1D25	B1D24	B1D23	B1D22	B1D21	B1D20	xxxx xxxx	45, 230
B1D1(7)	B1D17	B1D16	B1D15	B1D14	B1D13	B1D12	B1D11	B1D10	xxxx xxxx	46, 230
B1D0(7)	B1D07	B1D06	B1D05	B1D04	B1D03	B1D02	B1D01	B1D00	xxxx xxxx	46, 230
B1DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	46, 230
B1EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	46, 230
B1EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	46, 230
B1SIDL(7)	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	46, 230
B1SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	46, 230
B1CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	46, 230
B0D7(7)	B0D77	B0D76	B0D75	B0D74	B0D73	B0D72	B0D71	B0D70	xxxx xxxx	46, 230
B0D6(7)	B0D67	B0D66	B0D65	B0D64	B0D63	B0D62	B0D61	B0D60	xxxx xxxx	46, 230
B0D5(7)	B0D57	B0D56	B0D55	B0D54	B0D53	B0D52	B0D51	B0D50	xxxx xxxx	46, 230
B0D4(7)	B0D47	B0D46	B0D45	B0D44	B0D43	B0D42	B0D41	B0D40	xxxx xxxx	46, 230
B0D3(7)	B0D37	B0D36	B0D35	B0D34	B0D33	B0D32	B0D31	B0D30	xxxx xxxx	46, 230
B0D2(7)	B0D27	B0D26	B0D25	B0D24	B0D23	B0D22	B0D21	B0D20	xxxx xxxx	46, 230
B0D1(7)	B0D17	B0D16	B0D15	B0D14	B0D13	B0D12	B0D11	B0D10	xxxx xxxx	46, 230
B0D0(7)	B0D07	B0D06	B0D05	B0D04	B0D03	B0D02	B0D01	B0D00	xxxx xxxx	46, 230
B0DLC(7)	—	RTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	46, 230
B0EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	46, 230
B0EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	46, 230
B0SIDL(7)	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	46, 230
B0SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	46, 230
B0CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	46, 230
TXBIE(7)	—	—	—	TXB2IE	TXB1IE	TXB0IE	—	—	---0 00--	46, 230
BIE0(7)	B5IE	B4IE	B3IE	B2IE	B1IE	B0IE	RXB1IE	RXB0IE	0000 0000	46, 230
BSEL0(7)	B5TXEN	B4TXEN	B3TXEN	B2TXEN	B1TXEN	B0TXEN	—	—	0000 00--	46, 230
MSEL3(7)	FIL15_1	FIL15_0	FIL14_1	FIL14_0	FIL13_1	FIL13_0	FIL12_1	FIL12_0	0000 0000	46, 230
MSEL2(7)	FIL11_1	FIL11_0	FIL10_1	FIL10_0	FIL9_1	FIL9_0	FIL8_1	FIL8_0	0000 0000	46, 230
MSEL1(7)	FIL7_1	FIL7_0	FIL6_1	FIL6_0	FIL5_1	FIL5_0	FIL4_1	FIL4_0	0000 0101	46, 230
MSEL0(7)	FIL3_1	FIL3_0	FIL2_1	FIL2_0	FIL1_1	FIL1_0	FIL0_1	FIL0_0	0101 0000	46, 230
SDFLC(7)	—	—	—	DFLC4	DFLC3	DFLC2	DFLC1	DFLC0	---0 0000	46, 230
RXFCON1(7)	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN	0000 0000	46, 230
RXFCON0(7)	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN	0011 1111	47, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
RXFBCON7(7)	F15BP_3	F15BP_2	F15BP_1	F15BP_0	F14BP_3	F14BP_2	F14BP_1	F14BP_01	0000 0000	47, 230
RXFBCON6(7)	F13BP_3	F13BP_2	F13BP_1	F13BP_0	F12BP_3	F12BP_2	F12BP_1	F12BP_01	0000 0000	47, 230
RXFBCON5(7)	F11BP_3	F11BP_2	F11BP_1	F11BP_0	F10BP_3	F10BP_2	F10BP_1	F10BP_01	0000 0000	47, 230
RXFBCON4(7)	F9BP_3	F9BP_2	F9BP_1	F9BP_0	F8BP_3	F8BP_2	F8BP_1	F8BP_01	0000 0000	47, 230
RXFBCON3(7)	F7BP_3	F7BP_2	F7BP_1	F7BP_0	F6BP_3	F6BP_2	F6BP_1	F6BP_01	0000 0000	47, 230
RXFBCON2(7)	F5BP_3	F5BP_2	F5BP_1	F5BP_0	F4BP_3	F4BP_2	F4BP_1	F4BP_01	0000 0000	47, 230
RXFBCON1(7)	F3BP_3	F3BP_2	F3BP_1	F3BP_0	F2BP_3	F2BP_2	F2BP_1	F2BP_01	0000 0000	47, 230
RXFBCON0(7)	F1BP_3	F1BP_2	F1BP_1	F1BP_0	F0BP_3	F0BP_2	F0BP_1	F0BP_01	0000 0000	47, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{\text{MCLR}}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

4.10 Access Bank

The Access Bank is an architectural enhancement which is very useful for C compiler code optimization. The techniques used by the C compiler may also be useful for programs written in assembly.

This data memory region can be used for:

- Intermediate computational values
- Local variables of subroutines
- Faster context saving/switching of variables
- Common variables
- Faster evaluation/control of SFRs (no banking)

The Access Bank is comprised of the upper 160 bytes in Bank 15 (SFRs) and the lower 96 bytes in Bank 0. These two sections will be referred to as Access RAM High and Access RAM Low, respectively. Figure 4-7 indicates the Access RAM areas.

A bit in the instruction word specifies if the operation is to occur in the bank specified by the BSR register or in the Access Bank. This bit is denoted by the 'a' bit (for access bit).

When forced in the Access Bank (a = 0), the last address in Access RAM Low is followed by the first address in Access RAM High. Access RAM High maps the Special Function Registers so that these registers can be accessed without any software overhead. This is useful for testing status flags and modifying control bits.

4.11 Bank Select Register (BSR)

The need for a large general purpose memory space dictates a RAM banking scheme. The data memory is partitioned into sixteen banks. When using direct addressing, the BSR should be configured for the desired bank.

BSR<3:0> holds the upper 4 bits of the 12-bit RAM address. The BSR<7:4> bits will always read '0's and writes will have no effect.

A MOVLB instruction has been provided in the instruction set to assist in selecting banks.

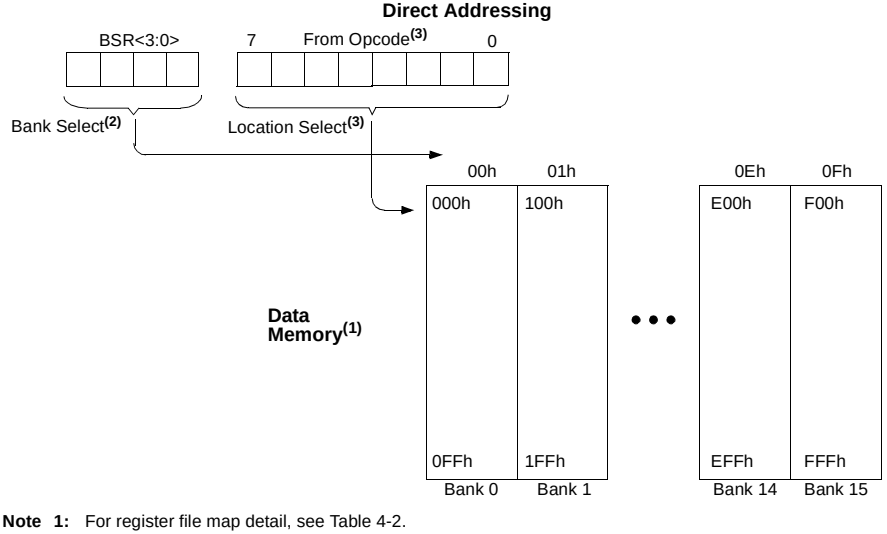
If the currently selected bank is not implemented, any read will return all '0's and all writes are ignored. The Status register bits will be set/cleared as appropriate for the instruction performed.

Each Bank extends up to 0FFh (256 bytes). All data memory is implemented as static RAM.

A MOVFF instruction ignores the BSR since the 12-bit addresses are embedded into the instruction word.

Section 4.12 "Indirect Addressing, INDF and FSR Registers" provides a description of indirect addressing which allows linear addressing of the entire RAM space.

FIGURE 4-8: DIRECT ADDRESSING



Note 1: For register file map detail, see Table 4-2.

2: The access bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

3: The MOVFF instruction embeds the entire 12-bit address in the instruction.

4.12 Indirect Addressing, INDF and FSR Registers

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. An FSR register is used as a pointer to the data memory location that is to be read or written. Since this pointer is in RAM, the contents can be modified by the program. This can be useful for data tables in the data memory and for software stacks. Figure 4-9 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Indirect addressing is possible by using one of the INDF registers. 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. The FSR register contains a 12-bit address which is shown in Figure 4-10.

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

Example 4-4 shows a simple use of indirect addressing to clear the RAM in Bank 1 (locations 100h-1FFh) in a minimum number of instructions.

EXAMPLE 4-4: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSR   FSR0, 100h   ;
NEXT    CLRF   POSTINC0     ; Clear INDF
                            ; register and
                            ; inc pointer
        BTFSS  FSR0H, 1     ; All done with
                            ; Bank1?
        BRA    NEXT         ; NO, clear next
CONTINUE                    ; YES, continue
```

There are three Indirect Addressing registers. To address the entire data memory space (4096 bytes), these registers are 12-bits wide. To store the 12 bits of addressing information, two 8-bit registers are required. These Indirect Addressing registers are:

1. FSR0: composed of FSR0H:FSR0L
2. FSR1: composed of FSR1H:FSR1L
3. FSR2: composed of FSR2H:FSR2L

In addition, there are registers INDF0, INDF1 and INDF2 which are not physically implemented. Reading or writing to these registers activates indirect addressing with the value in the corresponding FSR register being the address of the data. If an instruction writes a value to INDF0, the value will be written to the address pointed to by FSR0H:FSR0L. A read from INDF1 reads the data from the address pointed to by FSR1H:FSR1L. INDFn can be used in code anywhere an operand can be used.

If INDF0, INDF1, or INDF2 are read indirectly via an FSR, all '0's are read (zero bit is set). Similarly, if INDF0, INDF1, or INDF2 are written to indirectly, the operation will be equivalent to a NOP instruction and the Status bits are not affected.

4.12.1 INDIRECT ADDRESSING OPERATION

Each FSR register has an INDF register associated with it plus four additional register addresses. Performing an operation on one of these five registers determines how the FSR will be modified during indirect addressing.

When data access is done to one of the five INDFn locations, the address selected will configure the FSRn register to:

- Do nothing to FSRn after an indirect access (no change) – INDFn.
- Auto-decrement FSRn after an indirect access (post-decrement) – POSTDECn.
- Auto-increment FSRn after an indirect access (post-increment) – POSTINCn.
- Auto-increment FSRn before an indirect access (pre-increment) – PREINCn.
- Use the value in the WREG register as an offset to FSRn. Do not modify the value of the WREG or the FSRn register after an indirect access (no change) – PLUSWn.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the Status register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

Incrementing or decrementing an FSR affects all 12 bits. That is, when FSRnL overflows from an increment, FSRnH will be incremented automatically.

Adding these features allows the FSRn to be used as a stack pointer in addition to its uses for table operations in data memory.

Each FSR has an address associated with it that performs an indexed indirect access. When a data access to this INDFn location (PLUSWn) occurs, the FSRn is configured to add the signed value in the WREG register and the value in FSR to form the address before an indirect access. The FSR value is not changed.

If an FSR register contains a value that points to one of the INDFn, an indirect read will read 00h (zero bit is set), while an indirect write will be equivalent to a NOP (Status bits are not affected).

If an indirect addressing operation is done where the target address is an FSRnH or FSRnL register, the write operation will dominate over the pre- or post-increment/decrement functions.

FIGURE 4-9: INDIRECT ADDRESSING OPERATION

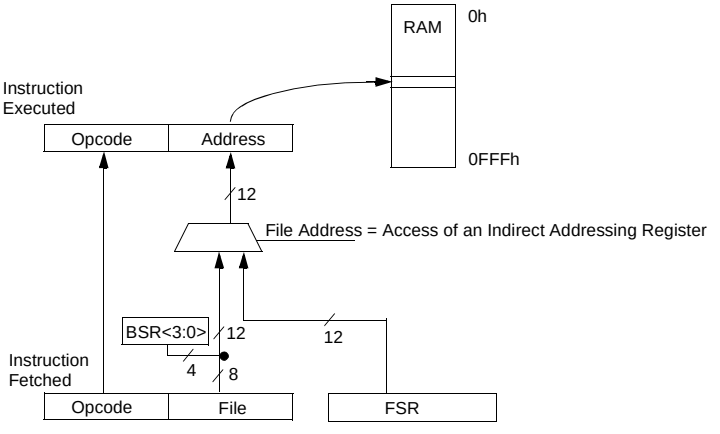
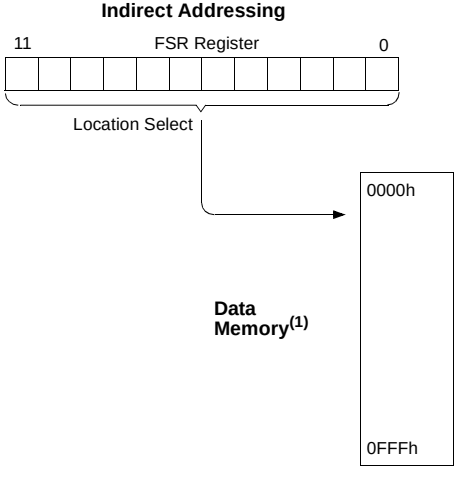


FIGURE 4-10: INDIRECT ADDRESSING



Note 1: For register file map detail, see Table 4-2.

4.13 Status Register

The Status register, shown in Register 4-3, contains the arithmetic status of the ALU. 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, C, OV or N bits, then the write to these five bits is disabled. These bits are set or cleared according to the device logic. 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`, `MOVFF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C, DC, OV or N bits from the Status register. For other instructions not affecting any status bits, see Table 25-2.

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

REGISTER 4-3: STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC	C
bit 7							bit 0

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

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative
0 = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

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

For `ADDWF`, `ADDLW`, `SUBLW`, and `SUBWF` instructions:

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

Note: For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the bit 4 or bit 3 of the source register.

bit 0 **C:** Carry/$\overline{\text{borrow}}$ bit

For `ADDWF`, `ADDLW`, `SUBLW`, and `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 $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2'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

4.14 RCON Register

The Reset Control (RCON) register contains flag bits that allow differentiation between the sources of a device Reset. These flags include the $\overline{TO}$, $\overline{PD}$, $\overline{POR}$, $\overline{BOR}$ and $\overline{RI}$ bits. This register is readable and writable.

Note 1: It is recommended that the $\overline{POR}$ bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

2: Brown-out Reset is said to have occurred when $\overline{BOR}$ is '0' and $\overline{POR}$ is '1' (assuming that $\overline{POR}$ was set to '1' by software immediately after POR).

REGISTER 4-4: RCON REGISTER

R/W-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0
IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
1 = The RESET instruction was not executed
0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 2 **$\overline{PD}$:** Power-down Detection Flag bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = A Power-on Reset has not occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = A Brown-out Reset has not 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

5.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation cannot be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

5.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

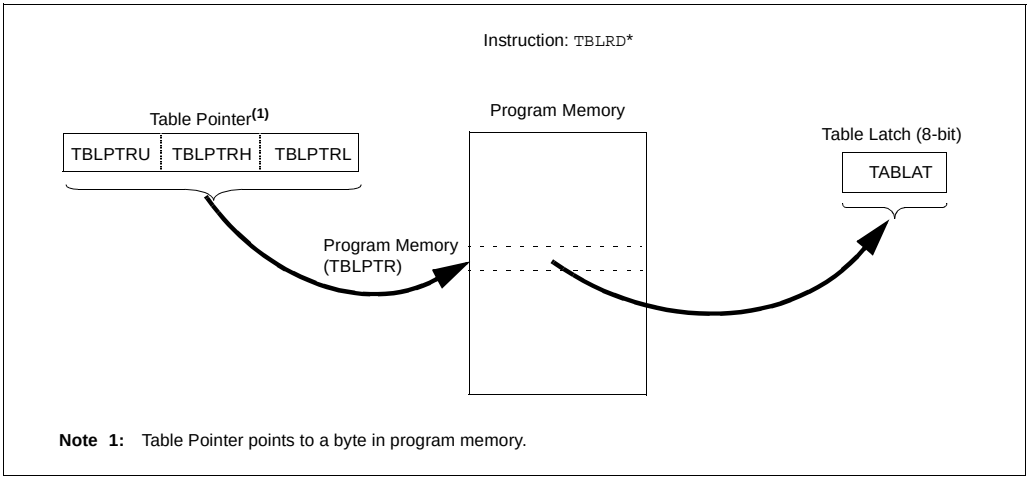
The program memory space is 16 bits wide, while the data RAM space is 8-bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and places it into the data RAM space. Figure 5-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 5.5 "Writing to Flash Program Memory"**. Figure 5-2 shows the operation of a table write with program memory and data RAM.

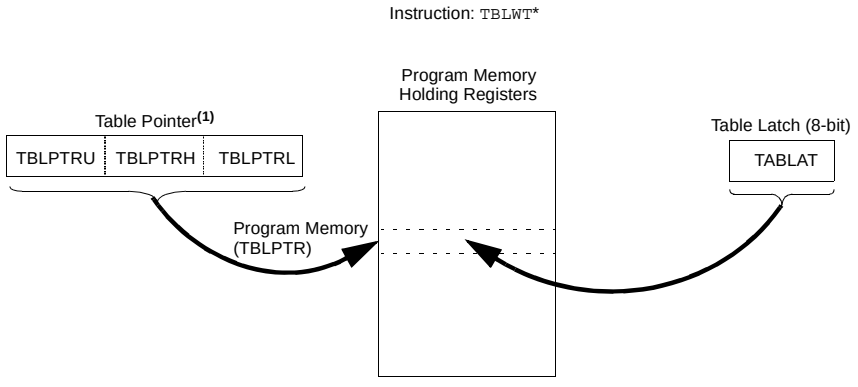
Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 5-1: TABLE READ OPERATION



Note 1: Table Pointer points to a byte in program memory.

FIGURE 5-2: TABLE WRITE OPERATION



Note 1: Table Pointer actually points to one of eight holding registers, the address of which is determined by TBLPTRL<2:0>. The process for physically writing data to the program memory array is discussed in **Section 5.5 "Writing to Flash Program Memory"**.

5.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

5.2.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the memory write and erase sequences.

Control bit EEPGD determines if the access will be a program or data EEPROM memory access. When clear, any subsequent operations will operate on the data EEPROM memory. When set, any subsequent operations will operate on the program memory.

Control bit CFGS determines if the access will be to the configuration/calibration registers or to program memory/data EEPROM memory. When set, subsequent operations will operate on configuration registers regardless of EEPGD (see **Section 24.0 "Special Features of the CPU"**). When clear, memory selection access is determined by EEPGD.

The FREE bit, when set, will allow a program memory erase operation. When the FREE bit is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to Reset values of zero.

The WR control bit initiates write operations. The bit cannot be cleared, only set in software; it is cleared in hardware at the completion of the write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

Note:	Interrupt flag bit, EEIF in the PIR2 register, is set when the write is complete. It must be cleared in software.

REGISTER 5-1: EECON1 REGISTER (ADDRESS FA6h)

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Access Flash program memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Access configuration registers
0 = Access Flash program or data EEPROM memory

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

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
0 = Perform write only

bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation)
0 = The write operation completed

Note: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read. (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Does not initiate an EEPROM read

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

5.2.2 TABLAT – TABLE LATCH REGISTER

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch is used to hold 8-bit data during data transfers between program memory and data RAM.

5.2.3 TBLPTR – TABLE POINTER REGISTER

The Table Pointer (TBLPTR) addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the device ID, the user ID and the configuration bits.

The Table Pointer, TBLPTR, is used by the `TBLRD` and `TBLWT` instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 5-1. These operations on the TBLPTR only affect the low-order 21 bits.

5.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a `TBLRD` is executed, all 22 bits of the table pointer determine which byte is read from program memory into TABLAT.

When a `TBLWT` is executed, the three LSbs of the Table Pointer (TBLPTR<2:0>) determine which of the eight program memory holding registers is written to. When the timed write to program memory (long write) begins, the 19 MSbs of the Table Pointer (TBLPTR<21:3>) will determine which program memory block of 8 bytes is written to. For more detail, see **Section 5.5 “Writing to Flash Program Memory”**.

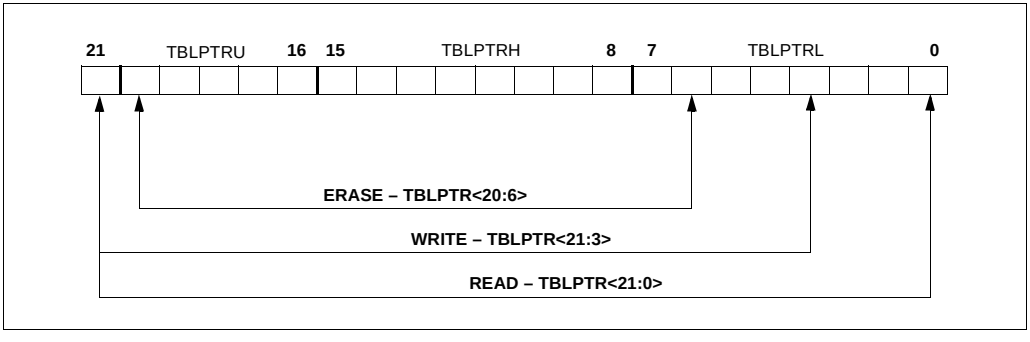
When an erase of program memory is executed, the 16 MSbs of the Table Pointer (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

Figure 5-3 describes the relevant boundaries of TBLPTR based on Flash program memory operations.

TABLE 5-1: TABLE POINTER OPERATIONS WITH `TBLRD` AND `TBLWT` INSTRUCTIONS

Example	Operation on Table Pointer
`TBLRD*` `TBLWT*`	TBLPTR is not modified
`TBLRD*+` `TBLWT*+`	TBLPTR is incremented after the read/write
`TBLRD*-` `TBLWT*-`	TBLPTR is decremented after the read/write
`TBLRD+*` `TBLWT+*`	TBLPTR is incremented before the read/write

FIGURE 5-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



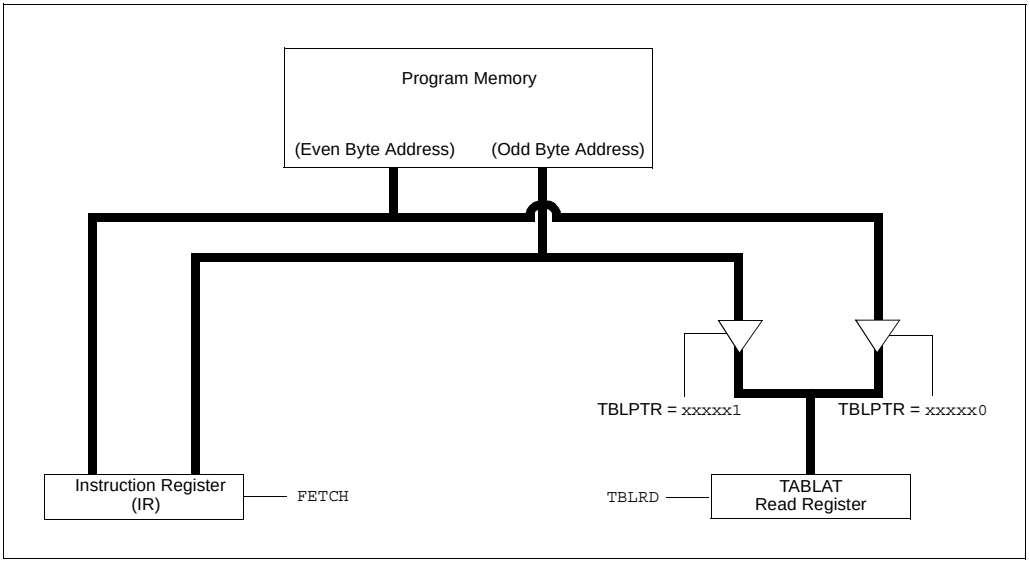
5.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 5-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 5-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 5-1: READING A FLASH PROGRAM MEMORY WORD

```
            MOVLW     upper(CODE_ADDR)          ; Load TBLPTR with the base
            MOVWF     TBLPTRU                   ; address of the word
            MOVLW     high(CODE_ADDR)
            MOVWF     TBLPTRH
            MOVLW     low(CODE_ADDR_LOW)
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                             ; read into TABLAT and increment
            MOVF      TABLAT, W                 ; get data
            MOVWF     LSB
            TBLRD*+                             ; read into TABLAT and increment
            MOVF      TABLAT, W                 ; get data
            MOVWF     MSB
```

5.4 Erasing Flash Program Memory

The minimum erase block is 32 words or 64 bytes. Only through the use of an external programmer or through ICSP control can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 64 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. TBLPTR<5:0> are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

For protection, the write initiate sequence for EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

5.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load table pointer with address of row being erased.
2. Set the EECON1 register for the erase operation:
   - set EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set WREN bit to enable writes;
   - set FREE bit to enable the erase.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the row erase cycle.
7. The CPU will stall for duration of the erase (about 2 ms using internal timer).
8. Execute a NOP.
9. Re-enable interrupts.

EXAMPLE 5-2: ERASING A FLASH PROGRAM MEMORY ROW

```
                MOVLW   upper(CODE_ADDR)    ; load TBLPTR with the base
                MOVWF   TBLPTRU             ; address of the memory block
                MOVLW   high(CODE_ADDR)
                MOVWF   TBLPTRH
                MOVLW   low(CODE_ADDR)
                MOVWF   TBLPTRL
    ERASE_ROW
                BSF     EECON1, EEPGD       ; point to Flash program memory
                BCF     EECON1, CFGS        ; access Flash program memory
                BSF     EECON1, WREN        ; enable write to memory
                BSF     EECON1, FREE        ; enable Row Erase operation
                BCF     INTCON, GIE         ; disable interrupts
                MOVLW   55h
                MOVWF   EECON2              ; write 55h
Required        MOVLW   0AAh
Sequence        MOVWF   EECON2              ; write 0AAh
                BSF     EECON1, WR          ; start erase (CPU stall)
                NOP
                BSF     INTCON, GIE         ; re-enable interrupts
```

5.5 Writing to Flash Program Memory

The minimum programming block is 4 words or 8 bytes. Word or byte programming is not supported.

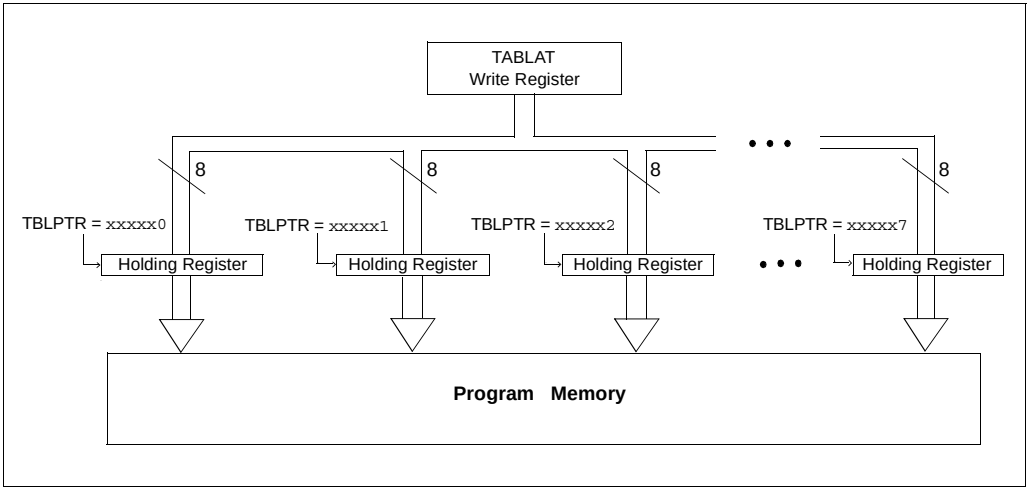
Table writes are used internally to load the holding registers needed to program the Flash memory. There are eight holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction has to be executed 8 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating eight registers, the EECON1 register must be written to, to start the programming operation with a long write.

The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

FIGURE 5-5: TABLE WRITES TO FLASH PROGRAM MEMORY



5.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load table pointer with address being erased.
4. Do the row erase procedure.
5. Load table pointer with address of first byte being written.
6. Write the first 8 bytes into the holding registers with auto-increment.
7. Set the EECON1 register for the write operation:
   - set EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 5 ms using internal timer).
13. Execute a `NOP`.
14. Re-enable interrupts.
15. Repeat steps 6-14 seven times to write 64 bytes.
16. Verify the memory (table read).

This procedure will require about 40 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 5-3.

Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the eight bytes in the holding register.

EXAMPLE 5-3: WRITING TO FLASH PROGRAM MEMORY

```
                MOVLW    D'64                    ; number of bytes in erase block
                MOVWF    COUNTER
                MOVLW    high(BUFFER_ADDR)       ; point to buffer
                MOVWF    FSR0H
                MOVLW    low(BUFFER_ADDR)
                MOVWF    FSR0L
                MOVLW    upper(CODE_ADDR)        ; Load TBLPTR with the base
                MOVWF    TBLPTRU                 ; address of the memory block
                MOVLW    high(CODE_ADDR)
                MOVWF    TBLPTRH
                MOVLW    low(CODE_ADDR)
                MOVWF    TBLPTRL
READ_BLOCK
                TBLRD*+                          ; read into TABLAT, and inc
                MOVF     TABLAT, W               ; get data
                MOVWF    POSTINC0                ; store data
                DECFSZ   COUNTER                 ; done?
                BRA      READ_BLOCK              ; repeat
MODIFY_WORD
                MOVLW    high(DATA_ADDR)         ; point to buffer
                MOVWF    FSR0H
                MOVLW    low(DATA_ADDR)
                MOVWF    FSR0L
                MOVLW    low(NEW_DATA)           ; update buffer word
                MOVWF    POSTINC0
                MOVLW    high(NEW_DATA)
                MOVWF    INDF0
```

EXAMPLE 5-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

```
ERASE_BLOCK
                MOVLW     upper(CODE_ADDR)        ; load TBLPTR with the base
                MOVWF     TBLPTRU                 ; address of the memory block
                MOVLW     high(CODE_ADDR)
                MOVWF     TBLPTRH
                MOVLW     low(CODE_ADDR)
                MOVWF     TBLPTRL
                BSF       EECON1, EEPGD           ; point to Flash program memory
                BCF       EECON1, CFGS            ; access Flash program memory
                BSF       EECON1, WREN            ; enable write to memory
                BSF       EECON1, FREE            ; enable Row Erase operation
                BCF       INTCON, GIE             ; disable interrupts
                MOVLW     55h
                MOVWF     EECON2                  ; write 55H
   Required     MOVLW     0AAh
   Sequence     MOVWF     EECON2                  ; write AAH
                BSF       EECON1, WR              ; start erase (CPU stall)
                NOP
                BSF       INTCON, GIE             ; re-enable interrupts
                TBLRD*-                           ; dummy read decrement
WRITE_BUFFER_BACK
                MOVLW     8                       ; number of write buffer groups of 8 bytes
                MOVWF     COUNTER_HI
                MOVLW     high(BUFFER_ADDR)       ; point to buffer
                MOVWF     FSR0H
                MOVLW     low(BUFFER_ADDR)
                MOVWF     FSR0L
PROGRAM_LOOP
                MOVLW     8                       ; number of bytes in holding register
                MOVWF     COUNTER
WRITE_WORD_TO_HREGS
                MOVFW     POSTINC0, W             ; get low byte of buffer data
                MOVWF     TABLAT                  ; present data to table latch
                TBLWT+*                           ; write data, perform a short write
                                                  ; to internal TBLWT holding register.
                DECFSZ    COUNTER                 ; loop until buffers are full
                BRA       WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
                BSF       EECON1, EEPGD           ; point to Flash program memory
                BCF       EECON1, CFGS            ; access Flash program memory
                BSF       EECON1, WREN            ; enable write to memory
                BCF       INTCON, GIE             ; disable interrupts
                MOVLW     55h
                MOVWF     EECON2                  ; write 55h
   Required     MOVLW     0AAh
   Sequence     MOVWF     EECON2                  ; write 0AAh
                BSF       EECON1, WR              ; start program (CPU stall)
                NOP
                BSF       INTCON, GIE             ; re-enable interrupts
                DECFSZ    COUNTER_HI              ; loop until done
                BRA       PROGRAM_LOOP
                BCF       EECON1, WREN            ; disable write to memory
```

5.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

5.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation. In these situations, users can check the WRERR bit and rewrite the location.

5.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 24.0 "Special Features of the CPU"** for more detail.

5.6 Flash Program Operation During Code Protection

See **Section 24.0 "Special Features of the CPU"** for details on code protection of Flash program memory.

TABLE 5-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

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
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000	--00 0000
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000	0000 0000
TBLPTRL	Program Memory Table Pointer High Byte (TBLPTR<7:0>)								0000 0000	0000 0000
TABLAT	Program Memory Table Latch								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 0000	0000 0000
EECON2	EEPROM Control Register 2 (not a physical register)								—	—
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-1 1111	-1-1 1111
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-0 0000	-0-0 0000
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-0 0000	-0-0 0000

Legend: x = unknown, u = unchanged, r = reserved, – = unimplemented, read as '0'.
Shaded cells are not used during Flash/EEPROM access.

6.0 EXTERNAL MEMORY INTERFACE

Note:	The external memory interface is not implemented on PIC18F6X8X (64/68-pin) devices.

The external memory interface is a feature of the PIC18F8X8X devices that allows the controller to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program memory.

The physical implementation of the interface uses 27 pins. These pins are reserved for external address/data bus functions; they are multiplexed with I/O port pins on four ports. Three I/O ports are multiplexed with the address/data bus, while the fourth port is multiplexed with the bus control signals. The I/O port functions are enabled when the EBDIS bit in the MEMCON register is set (see Register 6-1). A list of the multiplexed pins and their functions is provided in Table 6-1.

As implemented in the PIC18F8X8X devices, the interface operates in a similar manner to the external memory interface introduced on PIC18C601/801 microcontrollers. The most notable difference is that the interface on PIC18F8X8X devices only operates in 16-bit modes. The 8-bit mode is not supported.

For a more complete discussion of the operating modes that use the external memory interface, refer to **Section 4.1.1 "PIC18F8X8X Program Memory Modes"**.

6.1 Program Memory Modes and the External Memory Interface

As previously noted, PIC18F8X8X controllers are capable of operating in any one of four program memory modes using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the program memory mode selected as well as the setting of the EBDIS bit.

In **Microprocessor Mode**, the external bus is always active and the port pins have only the external bus function.

In **Microcontroller Mode,** the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted.

In **Microprocessor with Boot Block** or **Extended Microcontroller Mode,** the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function. If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port functions. When EBDIS = `0`, the pins function as the external bus. When EBDIS = `1`, the pins function as I/O ports.

REGISTER 6-1: MEMCON REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS(1)	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 7							bit 0

bit 7 **EBDIS**: External Bus Disable bit(1)
1 = External system bus disabled, all external bus drivers are mapped as I/O ports
0 = External system bus enabled and I/O ports are disabled

Note 1: This bit is ignored when device is accessing external memory either to fetch an instruction or perform TBLRD/TBLWT.

bit 6 **Unimplemented**: Read as '0'

bit 5-4 **WAIT<1:0>**: Table Reads and Writes Bus Cycle Wait Count bits
11 = Table reads and writes will wait 0 TCY
10 = Table reads and writes will wait 1 TCY
01 = Table reads and writes will wait 2 TCY
00 = Table reads and writes will wait 3 TCY

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **WM<1:0>**: TBLWT Operation with 16-bit Bus bits
1x = Word Write mode: LSB and MSB word output, $\overline{WRH}$ active when MSB written
01 = Byte Select mode: TABLAT data copied on both MS and LS Byte, $\overline{WRH}$ and ($\overline{UB}$ or $\overline{LB}$) will activate
00 = Byte Write mode: TABLAT data copied on both MS and LS Byte, $\overline{WRH}$ or $\overline{WRL}$ will activate

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

Note: The MEMCON register is held in Reset in Microcontroller mode.

If the device fetches or accesses external memory while EBDIS = 1, the pins will switch to external bus. If the EBDIS bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

When the device is executing out of internal memory (with EBDIS = 0) in Microprocessor with Boot Block mode or Extended Microcontroller mode, the control signals will be in inactive. They will go to a state where the AD<15:0>, A<19:16> are tri-state; the $\overline{OE}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{UB}$ and $\overline{LB}$ signals are '1'; and ALE and BA0 are '0'.

TABLE 6-1: PIC18F8X8X EXTERNAL BUS – I/O PORT FUNCTIONS

Name	Port	Bit	Function
RD0/AD0	PORTD	bit 0	Input/Output or System Bus Address bit 0 or Data bit 0
RD1/AD1	PORTD	bit 1	Input/Output or System Bus Address bit 1 or Data bit 1
RD2/AD2	PORTD	bit 2	Input/Output or System Bus Address bit 2 or Data bit 2
RD3/AD3	PORTD	bit 3	Input/Output or System Bus Address bit 3 or Data bit 3
RD4/AD4	PORTD	bit 4	Input/Output or System Bus Address bit 4 or Data bit 4
RD5/AD5	PORTD	bit 5	Input/Output or System Bus Address bit 5 or Data bit 5
RD6/AD6	PORTD	bit 6	Input/Output or System Bus Address bit 6 or Data bit 6
RD7/AD7	PORTD	bit 7	Input/Output or System Bus Address bit 7 or Data bit 7
RE0/AD8	PORTE	bit 0	Input/Output or System Bus Address bit 8 or Data bit 8
RE1/AD9	PORTE	bit 1	Input/Output or System Bus Address bit 9 or Data bit 9
RE2/AD10	PORTE	bit 2	Input/Output or System Bus Address bit 10 or Data bit 10
RE3/AD11	PORTE	bit 3	Input/Output or System Bus Address bit 11 or Data bit 11
RE4/AD12	PORTE	bit 4	Input/Output or System Bus Address bit 12 or Data bit 12
RE5/AD13	PORTE	bit 5	Input/Output or System Bus Address bit 13 or Data bit 13
RE6/AD14	PORTE	bit 6	Input/Output or System Bus Address bit 14 or Data bit 14
RE7/AD15	PORTE	bit 7	Input/Output or System Bus Address bit 15 or Data bit 15
RH0/A16	PORTH	bit 0	Input/Output or System Bus Address bit 16
RH1/A17	PORTH	bit 1	Input/Output or System Bus Address bit 17
RH2/A18	PORTH	bit 2	Input/Output or System Bus Address bit 18
RH3/A19	PORTH	bit 3	Input/Output or System Bus Address bit 19
RJ0/ALE	PORTJ	bit 0	Input/Output or System Bus Address Latch Enable (ALE) Control pin
RJ1/$\overline{OE}$	PORTJ	bit 1	Input/Output or System Bus Output Enable ($\overline{OE}$) Control pin
RJ2/$\overline{WRL}$	PORTJ	bit 2	Input/Output or System Bus Write Low ($\overline{WRL}$) Control pin
RJ3/$\overline{WRH}$	PORTJ	bit 3	Input/Output or System Bus Write High ($\overline{WRH}$) Control pin
RJ4/BA0	PORTJ	bit 4	Input/Output or System Bus Byte Address bit 0
RJ5/$\overline{CE}$	PORTJ	bit 5	Input/Output or Chip Enable
RJ6/$\overline{LB}$	PORTJ	bit 6	Input/Output or System Bus Lower Byte Enable ($\overline{LB}$) Control pin
RJ7/$\overline{UB}$	PORTJ	bit 7	Input/Output or System Bus Upper Byte Enable ($\overline{UB}$) Control pin

6.2 16-bit Mode

The external memory interface implemented in PIC18F8X8X devices operates only in 16-bit mode. The mode selection is not software configurable but is programmed via the configuration bits.

The WM<1:0> bits in the MEMCON register determine three types of connections in 16-bit mode. They are referred to as:

- 16-bit Byte Write
- 16-bit Word Write
- 16-bit Byte Select

These three different configurations allow the designer maximum flexibility in using 8-bit and 16-bit memory devices.

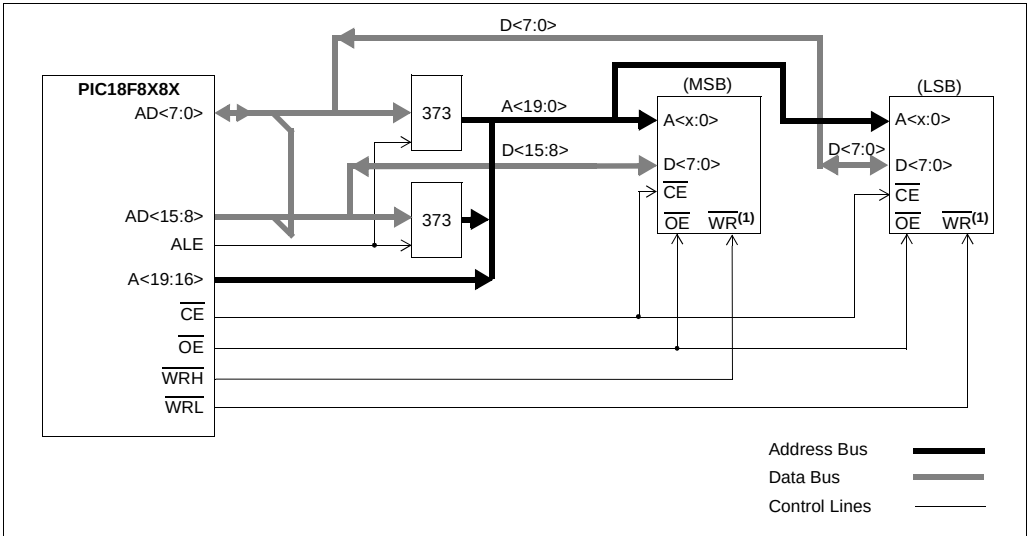
For all 16-bit modes, the Address Latch Enable (ALE) pin indicates that the Address bits (A<15:0>) are available on the external memory interface bus. Following the address latch, the Output Enable signal ($\overline{OE}$) will enable both bytes of program memory at once to form a 16-bit instruction word.

In Byte Select mode, JEDEC standard Flash memories will require BA0 for the byte address line, and one I/O line to select between Byte and Word mode. The other 16-bit modes do not need BA0. JEDEC standard static RAM memories will use the $\overline{UB}$ or $\overline{LB}$ signals for byte selection.

6.2.1 16-BIT BYTE WRITE MODE

Figure 6-1 shows an example of 16-bit Byte Write mode for PIC18F8X8X devices.

FIGURE 6-1: 16-BIT BYTE WRITE MODE EXAMPLE

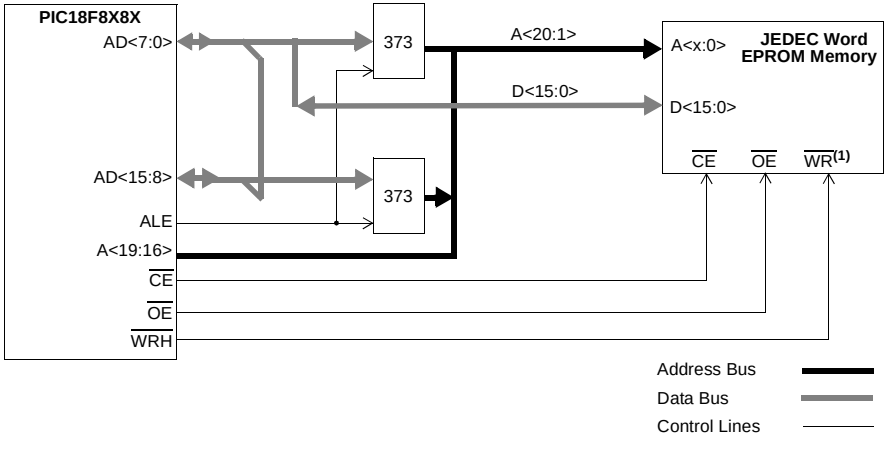


Note 1: This signal only applies to table writes. See **Section 5.1 "Table Reads and Table Writes"**.

6.2.2 16-BIT WORD WRITE MODE

Figure 6-2 shows an example of 16-bit Word Write mode for PIC18F8X8X devices.

FIGURE 6-2: 16-BIT WORD WRITE MODE EXAMPLE

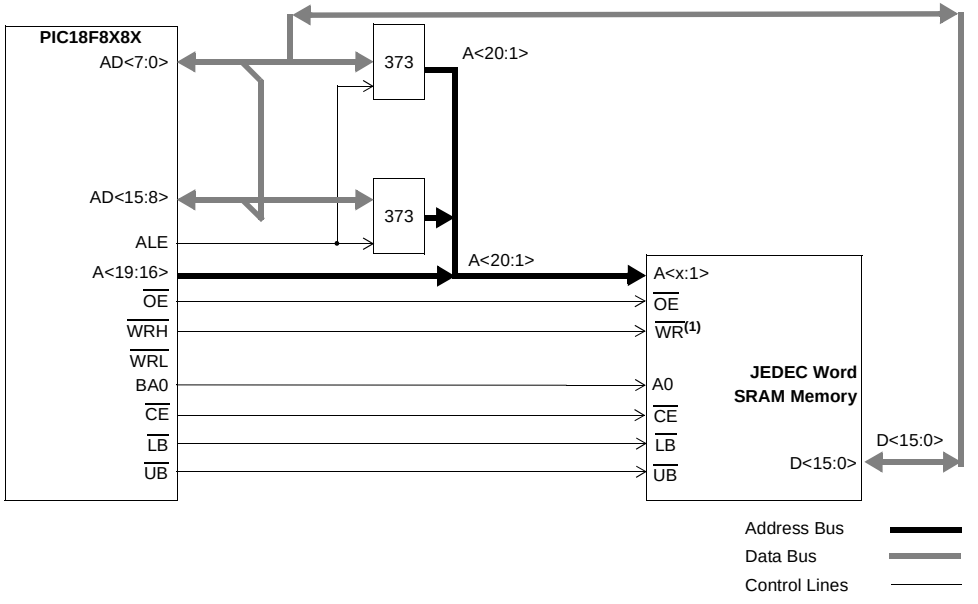


Note 1: This signal only applies to table writes. See **Section 5.1 "Table Reads and Table Writes"**.

6.2.3 16-BIT BYTE SELECT MODE

Figure 6-3 shows an example of 16-bit Byte Select mode for PIC18F8X8X devices.

FIGURE 6-3: 16-BIT BYTE SELECT MODE EXAMPLE

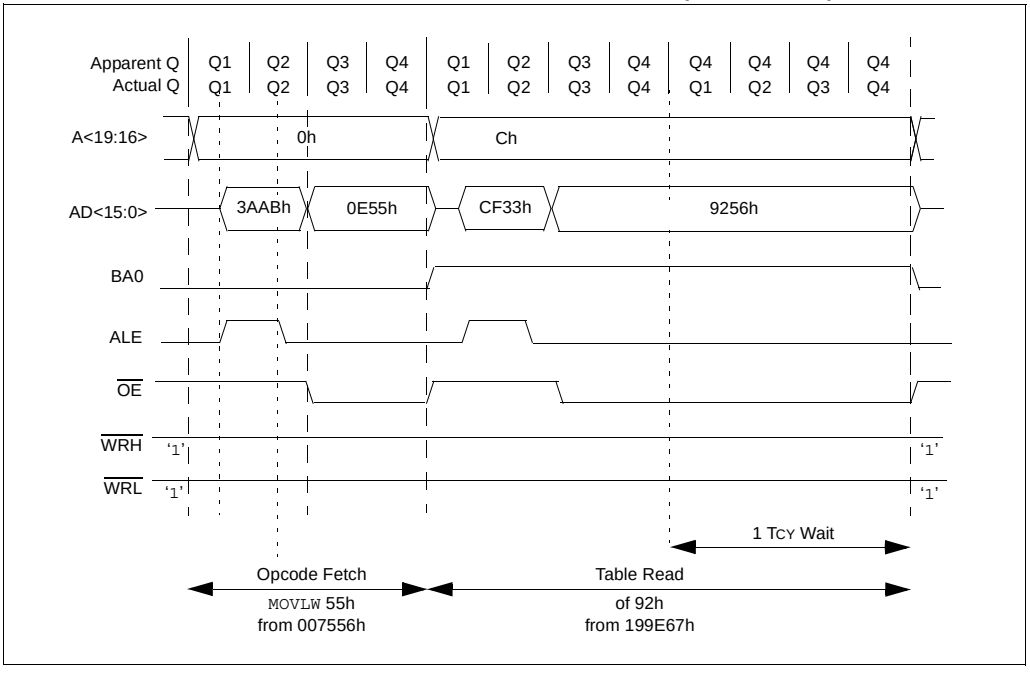


Note 1: This signal only applies to table writes. See **Section 5.1 "Table Reads and Table Writes"**.

6.2.4 16-BIT MODE TIMING

Figure 6-4 shows the 16-bit mode external bus timing for PIC18F8X8X devices.

FIGURE 6-4: EXTERNAL PROGRAM MEMORY BUS TIMING (16-BIT MODE)



NOTES:

7.0 DATA EEPROM MEMORY

The data EEPROM is readable and writable during normal operation over the entire VDD range. The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are five SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADR
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 1024 bytes of data EEPROM with an address range from 0h to 3FFh.

The EEPROM data memory is rated for high erase/write cycles. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Electrical Characteristics, **Section 27.0 "Electrical Characteristics"**) for exact limits.

7.1 EEADRH:EEADR

The address register pair, EEADRH:EEADR, can address up to a maximum of 1024 bytes of data EEPROM.

7.2 EECON1 and EECON2 Registers

EECON1 is the control register for EEPROM memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the EEPROM write sequence.

Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset or a WDT Time-out Reset during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to the Reset condition forcing the contents of the registers to zero.

Note:	Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared in software.

REGISTER 7-1: EECON1 REGISTER (ADDRESS FA6h)

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Access Flash program memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EE or Configuration Select bit
1 = Access configuration or calibration registers
0 = Access Flash program or data EEPROM memory

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

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
0 = Perform write only

bit 3 **WRERR:** Flash Program/Data EE Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or any WDT Reset during self-timed programming in normal operation)
0 = The write operation completed

Note: When a WRERR occurs, the EEPGD or FREE bits are not cleared. This allows tracing of the error condition.

bit 2 **WREN:** Flash Program/Data EE Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read. (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Does not initiate an EEPROM read

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

7.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>), clear the CFGS control bit (EECON1<6>) and then set control bit, RD (EECON1<0>). The data is available for the very next instruction cycle; therefore, the EEDATA register can be read by the next instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

EXAMPLE 7-1: DATA EEPROM READ

```
MOVLW   DATA_EE_ADR_HI      ;
MOVWF   EEADRH              ;
MOVLW   DATA_EE_ADDR_LOW    ;
MOVWF   EEADR               ; Data Memory Address to read
BCF     EECON1, EEPGD       ; Point to DATA memory
BCF     EECON1, CFGS        ; Access program Flash or Data EEPROM memory
BSF     EECON1, RD          ; EEPROM Read
MOVF    EEDATA, W           ; W = EEDATA
```

7.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADRH:EEADR register pair and the data written to the EEDATA register. Then the sequence in Example 7-2 must be followed to initiate the write cycle.

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADRH:EEADR and EDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt or poll this bit. EEIF must be cleared by software.

EXAMPLE 7-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDR_HI     ;
                MOVWF   EEADRH              ;
                MOVLW   DATA_EE_ADDR_LOW    ;
                MOVWF   EEADR               ; Data Memory Address to read
                MOVLW   DATA_EE_DATA        ;
                MOVWF   EEDATA              ; Data Memory Value to write
                BCF     EECON1, EEPGD       ; Point to DATA memory
                BCF     EECON1, CFGS        ; Access program Flash or Data EEPROM memory
                BSF     EECON1, WREN        ; Enable writes

                BCF     INTCON, GIE         ; Disable interrupts
Required        MOVLW   55h                 ;
Sequence        MOVWF   EECON2              ; Write 55h
                MOVLW   0AAh                ;
                MOVWF   EECON2              ; Write 0AAh
                BSF     EECON1, WR          ; Set WR bit to begin write
                BSF     INTCON, GIE         ; Enable interrupts

                .                           ; user code execution
                .
                .
                BCF     EECON1, WREN        ; Disable writes on write complete (EEIF set)
```

7.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

7.6 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

7.7 Operation During Code-Protect

Data EEPROM memory has its own code-protect mechanism. External read and write operations are disabled if either of these mechanisms are enabled.

The microcontroller itself can both read and write to the internal data EEPROM regardless of the state of the code-protect configuration bit. Refer to **Section 24.0 "Special Features of the CPU"** for additional information.

7.8 Using the Data EEPROM

The data EEPROM is a high endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than specification D124. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

Note:	If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification D124.

EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADRH              ;
        CLRF    EEADR               ; Start at address 0
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   0AAh                ;
        MOVWF   EECON2              ; Write 0AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     Loop                ; Not zero, do it again

        INCFS2  EEADRH, F           ;
        BRA     Loop                ;
        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

TABLE 7-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

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
INTCON	GIE/ GIEH	PEIE/ GIEL	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
EEADRH	—	—	—	—	—	—	EE Addr High		---- --00	---- --00
EEADR	EEPROM Address Register								0000 0000	0000 0000
EEDATA	EEPROM Data Register								0000 0000	0000 0000
EECON2	EEPROM Control Register 2 (not a physical register)								—	—
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-1 1111	---1 1111
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-0 0000	---0 0000
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-0 0000	---0 0000

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented, read as '0'.
Shaded cells are not used during Flash/EEPROM access.

NOTES:

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

An 8 x 8 hardware multiplier is included in the ALU of the PIC18F6585/8585/6680/8680 devices. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored in the 16-bit product register pair (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors.

Table 8-1 shows a performance comparison between enhanced devices using the single-cycle hardware multiply and performing the same function without the hardware multiply.

8.2 Operation

Example 8-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiply. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W         ;
MULWF   ARG2            ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
```

EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W         ;
MULWF   ARG2            ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
BTFSC   ARG2, SB        ; Test Sign Bit
SUBWF   PRODH           ; PRODH = PRODH
                        ;         - ARG1
MOVF    ARG2, W         ;
BTFSC   ARG1, SB        ; Test Sign Bit
SUBWF   PRODH           ; PRODH = PRODH
                        ;         - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 µs	27.6 µs	69 µs
	Hardware multiply	1	1	100 ns	400 ns	1 µs
8 x 8 signed	Without hardware multiply	33	91	9.1 µs	36.4 µs	91 µs
	Hardware multiply	6	6	600 ns	2.4 µs	6 µs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 µs	96.8 µs	242 µs
	Hardware multiply	24	24	2.4 µs	9.6 µs	24 µs
16 x 16 signed	Without hardware multiply	52	254	25.4 µs	102.6 µs	254 µs
	Hardware multiply	36	36	3.6 µs	14.4 µs	36 µs

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in four registers, RES3:RES0.

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

```
RES3:RES0  =  ARG1H:ARG1L • ARG2H:ARG2L
           =  (ARG1H • ARG2H • 216) +
              (ARG1H • ARG2L • 28) +
              (ARG1L • ARG2H • 28) +
              (ARG1L • ARG2L)
```

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L         ; ARG1L * ARG2L ->
                          ; PRODH:PRODL
    MOVFF   PRODH, RES1   ;
    MOVFF   PRODL, RES0   ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H         ; ARG1H * ARG2H ->
                          ; PRODH:PRODL
    MOVFF   PRODH, RES3   ;
    MOVFF   PRODL, RES2   ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H         ; ARG1L * ARG2H ->
                          ; PRODH:PRODL
    MOVF    PRODL, W      ;
    ADDWF   RES1          ; Add cross
    MOVF    PRODH, W      ; products
    ADDWFC  RES2          ;
    CLRF    WREG          ;
    ADDWFC  RES3          ;
;
    MOVF    ARG1H, W      ;
    MULWF   ARG2L         ; ARG1H * ARG2L ->
                          ; PRODH:PRODL
    MOVF    PRODL, W      ;
    ADDWF   RES1          ; Add cross
    MOVF    PRODH, W      ; products
    ADDWFC  RES2          ;
    CLRF    WREG          ;
    ADDWFC  RES3          ;
```

Example 8-4 shows the sequence to do a 16 x 16 signed multiply. Equation 8-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pairs' Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

```
RES3:RES0
    =  ARG1H:ARG1L • ARG2H:ARG2L
    =  (ARG1H • ARG2H • 216) +
       (ARG1H • ARG2L • 28) +
       (ARG1L • ARG2H • 28) +
       (ARG1L • ARG2L) +
       (-1 • ARG2H<7> • ARG1H:ARG1L • 216) +
       (-1 • ARG1H<7> • ARG2H:ARG2L • 216)
```

EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L         ; ARG1L * ARG2L ->
                          ; PRODH:PRODL
    MOVFF   PRODH, RES1   ;
    MOVFF   PRODL, RES0   ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H         ; ARG1H * ARG2H ->
                          ; PRODH:PRODL
    MOVFF   PRODH, RES3   ;
    MOVFF   PRODL, RES2   ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H         ; ARG1L * ARG2H ->
                          ; PRODH:PRODL
    MOVF    PRODL, W      ;
    ADDWF   RES1          ; Add cross
    MOVF    PRODH, W      ; products
    ADDWFC  RES2          ;
    CLRF    WREG          ;
    ADDWFC  RES3          ;
;
    MOVF    ARG1H, W      ;
    MULWF   ARG2L         ; ARG1H * ARG2L ->
                          ; PRODH:PRODL
    MOVF    PRODL, W      ;
    ADDWF   RES1          ; Add cross
    MOVF    PRODH, W      ; products
    ADDWFC  RES2          ;
    CLRF    WREG          ;
    ADDWFC  RES3          ;
;
    BTFSS   ARG2H, 7      ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1     ; no, check ARG1
    MOVF    ARG1L, W      ;
    SUBWF   RES2          ;
    MOVF    ARG1H, W      ;
    SUBWFB  RES3
;
SIGN_ARG1
    BTFSS   ARG1H, 7      ; ARG1H:ARG1L neg?
    BRA     CONT_CODE     ; no, done
    MOVF    ARG2L, W      ;
    SUBWF   RES2          ;
    MOVF    ARG2H, W      ;
    SUBWFB  RES3
;
CONT_CODE
      :
```

9.0 INTERRUPTS

The PIC18F6585/8585/6680/8680 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high or a low priority level. The high priority interrupt vector is at 000008h while the low priority interrupt vector is at 000018h. High priority interrupt events will override any low priority interrupts that may be in progress.

There are thirteen registers which are used to control interrupt operation. They are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

Each interrupt source (except INT0) has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set. Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared. When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h depending on the priority level. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

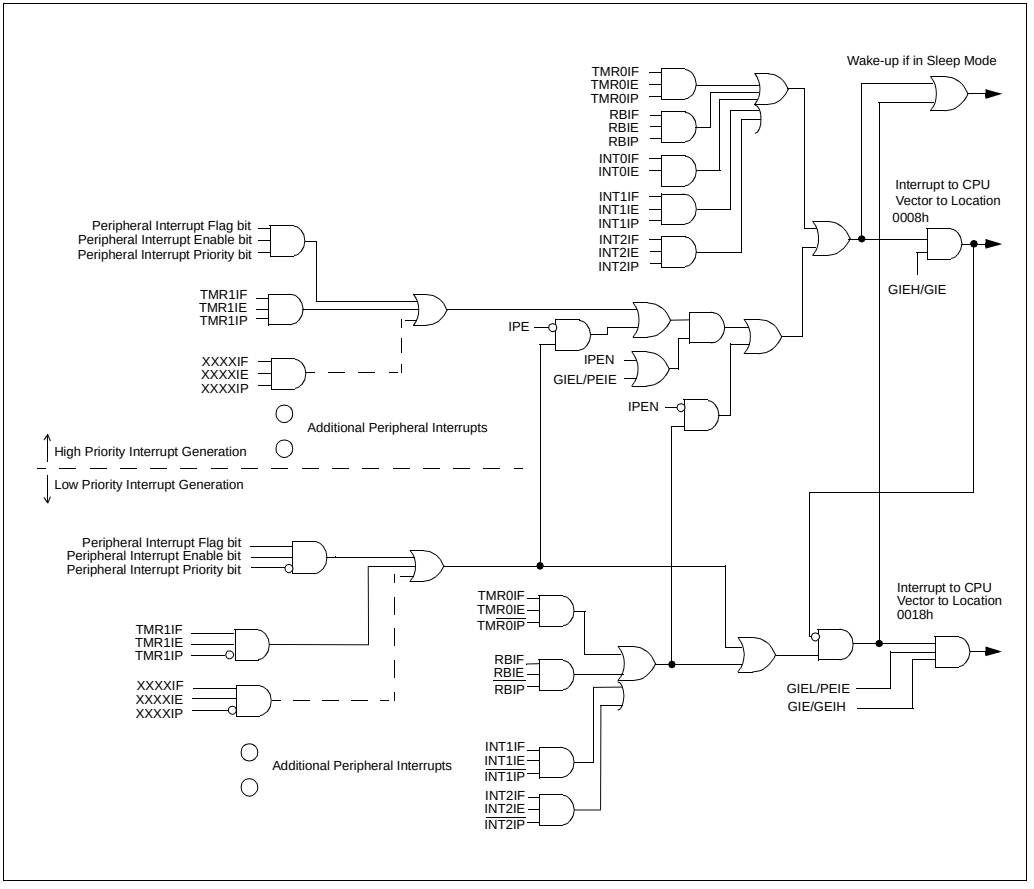
When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one- or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

FIGURE 9-1: INTERRUPT LOGIC



9.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits.

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON 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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit

When IPEN (RCON<7>) = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts

When IPEN (RCON<7>) = 1:
1 = Enables all high priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit

When IPEN (RCON<7>) = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

When IPEN (RCON<7>) = 1:
1 = Enables all low priority peripheral interrupts
0 = Disables all low priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 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 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 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

Note: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

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

REGISTER 9-2: INTCON2 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
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2**: External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3**: External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RBIP**: RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

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

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-3: INTCON3 REGISTER

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
bit 7							bit 0

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

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

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

9.2 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Flag registers (PIR1, PIR2 and PIR3).

Note 1:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).
2:	User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt, and after servicing that interrupt.

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

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

bit 7 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit(1)
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 (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer, RCREG, is full (cleared when RCREG is read)
0 = The USART receive buffer is empty

bit 4 **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer, TXREG, is empty (cleared when TXREG is written)
0 = The USART transmit buffer is full

bit 3 **SSPIF**: Master Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2 **CCP1IF**: Enhanced 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: Available in Microcontroller mode only.

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

REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF
bit 7							bit 0

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

bit 6 **CMIF**: Comparator Interrupt Flag bit
1 = The comparator input has changed (must be cleared in software)
0 = The comparator input has not changed

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

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

bit 3 **BCLIF**: Bus Collision Interrupt Flag bit
1 = A bus collision occurred while the SSP module (configured in I^2C Master mode) was transmitting (must be cleared in software)
0 = No bus collision occurred

bit 2 **LVDIF**: Low-Voltage Detect Interrupt Flag bit
1 = A low-voltage condition occurred (must be cleared in software)
0 = The device voltage is above the Low-Voltage Detect trip point

bit 1 **TMR3IF**: TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **CCP2IF**: CCP2 Interrupt Flag bit
Capture mode:
1 = A TMR1 or TMR3 register capture occurred (must be cleared in software)
0 = No TMR1 or TMR3 register capture occurred
Compare mode:
1 = A TMR1 or TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1 or TMR3 register compare match occurred
PWM mode:
Unused in this mode.

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

REGISTER 9-6: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRXIF	WAKIF	ERRIF	TXB2IF/ TXBnIF	TXB1IF(1)	TXB0IF(1)	RXB1IF/ RXBnIF	RXB0IF/ FIFOWMIF
bit 7							bit 0

bit 7 **IRXIF:** CAN Invalid Received Message Interrupt Flag bit
1 = An invalid message has occurred on the CAN bus
0 = No invalid message on CAN bus

bit 6 **WAKIF:** CAN bus Activity Wake-up Interrupt Flag bit
1 = Activity on CAN bus has occurred
0 = No activity on CAN bus

bit 5 **ERRIF:** CAN bus Error Interrupt Flag bit
1 = An error has occurred in the CAN module (multiple sources)
0 = No CAN module errors

bit 4 When CAN is in Mode 0:
TXB2IF: CAN Transmit Buffer 2 Interrupt Flag bit
1 = Transmit Buffer 2 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 2 has not completed transmission of a message
When CAN is in Mode 1 or 2:
TXBnIF: Any Transmit Buffer Interrupt Flag bit
1 = One or more transmit buffers has completed transmission of a message and may be reloaded (TXBIE or BIE0<7:2> must be non-zero)
0 = No message was transmitted

bit 3 **TXB1IF:** CAN Transmit Buffer 1 Interrupt Flag bit(1)
1 = Transmit Buffer 1 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 1 has not completed transmission of a message

bit 2 **TXB0IF:** CAN Transmit Buffer 0 Interrupt Flag bit(1)
1 = Transmit Buffer 0 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 0 has not completed transmission of a message

bit 1 When CAN is in Mode 0:
RXB1IF: CAN Receive Buffer 1 Interrupt Flag bit
1 = Receive Buffer 1 has received a new message
0 = Receive Buffer 1 has not received a new message
When CAN is in Mode 1 or 2:
RXBnIF: CAN Receive Buffer Interrupt Flag bit
1 = One or more receive buffers has received a new message
0 = No receive buffer has received a new message

bit 0 When CAN is in Mode 0:
RXB0IF: CAN Receive Buffer 0 Interrupt Flag bit(1)
1 = Receive Buffer 0 has received a new message
0 = Receive Buffer 0 has not received a new message
When CAN is in Mode 1:
Unimplemented: Read as '0'
When CAN is in Mode 2:
FIFOWMIF: FIFO Watermark Interrupt Flag bit
1 = FIFO high watermark is reached
0 = FIFO high watermark is not reached

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

9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2 and PIE3). When the IPEN bit (RCON<7>) is '0', the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 9-7: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

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

bit 7 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit(1)
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

Note 1: Available in Microcontroller mode only.

bit 6 **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D 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 **SSPIE**: Master Synchronous Serial Port Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE**: Enhanced 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

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

REGISTER 9-8: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE
bit 7							bit 0

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

bit 6 **CMIE**: Comparator Interrupt Enable bit
1 = Enables the comparator interrupt
0 = Disables the comparator interrupt

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

bit 4 **EEIE**: Data EEPROM/Flash Write Operation Interrupt Enable bit
1 = Enables the write operation interrupt
0 = Disables the write operation interrupt

bit 3 **BCLIE**: Bus Collision Interrupt Enable bit
1 = Enables the bus collision interrupt
0 = Disables the bus collision interrupt

bit 2 **LVDIE**: Low-Voltage Detect Interrupt Enable bit
1 = Enables the Low-Voltage Detect interrupt
0 = Disables the Low-Voltage Detect interrupt

bit 1 **TMR3IE**: TMR3 Overflow Interrupt Enable bit
1 = Enables the TMR3 overflow interrupt
0 = Disables the TMR3 overflow interrupt

bit 0 **CCP2IE**: CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

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

REGISTER 9-9: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRXIE	WAKIE	ERRIE	TXB2IE/ TXBnIE	TXB1IE(1)	TXB0IE(1)	RXB1IE/ RXBnIE	RXB0IE/ FIFOWMIE
bit 7							bit 0

bit 7 **IRXIE:** CAN Invalid Received Message Interrupt Enable bit
1 = Enable invalid message received interrupt
0 = Disable invalid message received interrupt

bit 6 **WAKIE:** CAN bus Activity Wake-up Interrupt Enable bit
1 = Enable bus activity wake-up interrupt
0 = Disable bus activity wake-up interrupt

bit 5 **ERRIE:** CAN bus Error Interrupt Enable bit
1 = Enable CAN bus error interrupt
0 = Disable CAN bus error interrupt

bit 4 When CAN is in Mode 0:
TXB2IE: CAN Transmit Buffer 2 Interrupt Enable bit
1 = Enable Transmit Buffer 2 interrupt
0 = Disable Transmit Buffer 2 interrupt
When CAN is in Mode 1 or 2:
TXBnIE: CAN Transmit Buffer Interrupts Enable bit
1 = Enable transmit buffer interrupt; individual interrupt is enabled by TXBIE and BIE0
0 = Disable all transmit buffer interrupts

bit 3 **TXB1IE:** CAN Transmit Buffer 1 Interrupt Enable bit(1)
1 = Enable Transmit Buffer 1 interrupt
0 = Disable Transmit Buffer 1 interrupt

bit 2 **TXB0IE:** CAN Transmit Buffer 0 Interrupt Enable bit(1)
1 = Enable Transmit Buffer 0 interrupt
0 = Disable Transmit Buffer 0 interrupt

bit 1 When CAN is in Mode 0:
RXB1IE: CAN Receive Buffer 1 Interrupt Enable bit
1 = Enable Receive Buffer 1 interrupt
0 = Disable Receive Buffer 1 interrupt
When CAN is in Mode 1 or 2:
RXBnIE: CAN Receive Buffer Interrupts Enable bit
1 = Enable receive buffer interrupt; individual interrupt is enabled by BIE0
0 = Disable all receive buffer interrupts

bit 0 When CAN is in Mode 0:
RXB0IE: CAN Receive Buffer 0 Interrupt Enable bit
1 = Enable Receive Buffer 0 interrupt
0 = Disable Receive Buffer 0 interrupt
When CAN is in Mode 1:
Unimplemented: Read as '0'
When CAN is in Mode 2:
FIFOWMIE: FIFO Watermark Interrupt Enable bit
1 = Enable FIFO watermark interrupt
0 = Disable FIFO watermark interrupt

Note 1: In CAN Mode 1 and 2, this bit is forced to '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

9.4 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Priority registers (IPR1, IPR2 and IPR3). The operation of the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

REGISTER 9-10: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

bit 7 **PSPIP:** Parallel Slave Port Read/Write Interrupt Priority bit(1)
1 = High priority
0 = Low priority

Note 1: Available in Microcontroller mode only.

bit 6 **ADIP**: A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RCIP**: USART Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TXIP**: USART Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **SSPIP**: Master Synchronous Serial Port Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP1IP**: CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP**: TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP**: TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

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

REGISTER 9-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

U-0	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP
bit 7							bit 0

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

bit 6 **CMIP**: Comparator Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 4 **EEIP**: Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **BCLIP**: Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **LVDIP**: Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP**: TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **CCP2IP**: CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

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

REGISTER 9-12: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
IRXIP	WAKIP	ERRIP	TXB2IP/ TXBnIP	TXB1IP(1)	TXB0IP(1)	RXB1IP/ RXBnIP	RXB0IP/ FIFOWMIP
bit 7							bit 0

bit 7 **IRXIP:** CAN Invalid Received Message Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **WAKIP:** CAN bus Activity Wake-up Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **ERRIP:** CAN bus Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 When CAN is in Mode 0:
TXB2IP: CAN Transmit Buffer 2 Interrupt Priority bit
1 = High priority
0 = Low priority
When CAN is in Mode 1 or 2:
TXBnIP: CAN Transmit Buffer Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **TXB1IP:** CAN Transmit Buffer 1 Interrupt Priority bit(1)
1 = High priority
0 = Low priority

bit 2 **TXB0IP:** CAN Transmit Buffer 0 Interrupt Priority bit(1)
1 = High priority
0 = Low priority

bit 1 When CAN is in Mode 0:
RXB1IP: CAN Receive Buffer 1 Interrupt Priority bit
1 = High priority
0 = Low priority
When CAN is in Mode 1 or 2:
RXBnIP: CAN Receive Buffer Interrupts Priority bit
1 = High priority
0 = Low priority

bit 0 When CAN is in Mode 0:
RXB0IP: CAN Receive Buffer 0 Interrupt Priority bit
1 = High priority
0 = Low priority
When CAN is in Mode 1:
Unimplemented: Read as '0'
When CAN is in Mode 2:
FIFOWMIP: FIFO Watermark Interrupt Priority bit
1 = High priority
0 = Low priority

Note 1: In CAN Mode 1 and 2, this bit is forced to '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

9.5 RCON Register

The RCON register contains the IPEN bit which is used to enable prioritized interrupts. The functions of the other bits in this register are discussed in more detail in **Section 4.14 “RCON Register”**.

REGISTER 9-13: RCON REGISTER

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16 Compatibility mode)

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
For details of bit operation, see Register 4-4.

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
For details of bit operation, see Register 4-4.

bit 2 **$\overline{PD}$:** Power-down Detection Flag bit
For details of bit operation, see Register 4-4.

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
For details of bit operation, see Register 4-4.

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
For details of bit operation, see Register 4-4.

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

9.6 INT0 Interrupt

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/INT3 pins are edge-triggered: either rising if the corresponding INTEDGx bit is set in the INTCON2 register, or falling if the INTEDGx bit is clear. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit, INTxF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxE. Flag bit, INTxF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt. All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from Sleep if bit INTxIE was set prior to going into Sleep. If the global interrupt enable bit GIE is set, the processor will branch to the interrupt vector following wake-up.

The interrupt priority for INT, INT2 and INT3 is determined by the value contained in the interrupt priority bits: INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0; it is always a high priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (0FFh → 00h) will set flag bit TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L registers (0FFFFh → 0000h) will set flag bit, TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 11.0 "Timer0 Module"** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During an interrupt, the return PC value is saved on the stack. Additionally, the WREG, Status and BSR registers are saved on the fast return stack. If a fast return from interrupt is not used (See **Section 4.3 "Fast Register Stack"**), the user may need to save the WREG, Status and BSR registers in software. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, Status and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF     W_TEMP                    ; W_TEMP is in virtual bank
MOVFF     STATUS, STATUS_TEMP       ; STATUS_TEMP located anywhere
MOVFF     BSR, BSR_TEMP             ; BSR located anywhere
;
; USER ISR CODE
;
MOVFF     BSR_TEMP, BSR             ; Restore BSR
MOVF      W_TEMP, W                 ; Restore WREG
MOVFF     STATUS_TEMP, STATUS       ; Restore STATUS
```

10.0 I/O PORTS

Depending on the device selected, there are either seven or nine I/O ports available on PIC18F6X8X/8X8X devices. Some of their pins are multiplexed with one or more alternate functions from the other 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.

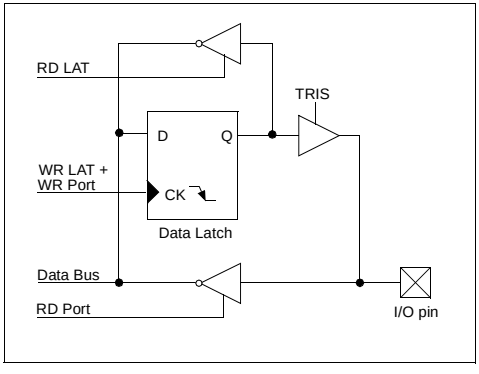
Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (output latch)

The Data Latch register (LAT) is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified version of a generic I/O port and its operation is shown in Figure 10-1.

FIGURE 10-1: SIMPLIFIED BLOCK DIAGRAM OF PORT/LAT/TRIS OPERATION



10.1 PORTA, TRISA and LATA Registers

PORTA is a 7-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.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA4 pin 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.

The RA6 pin is only enabled as a general I/O pin in ECIO and RCIO Oscillator modes.

The other PORTA pins are multiplexed with analog inputs and the analog VREF+ and VREF- inputs. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

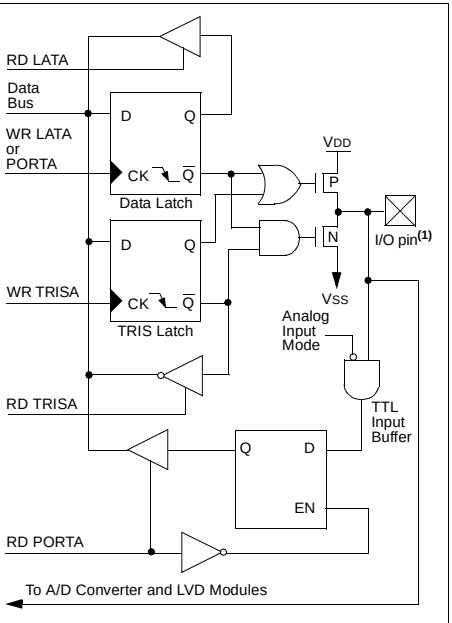
Note:	On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as '0'. RA6 and RA4 are configured as digital inputs.

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.

EXAMPLE 10-1: INITIALIZING PORTA

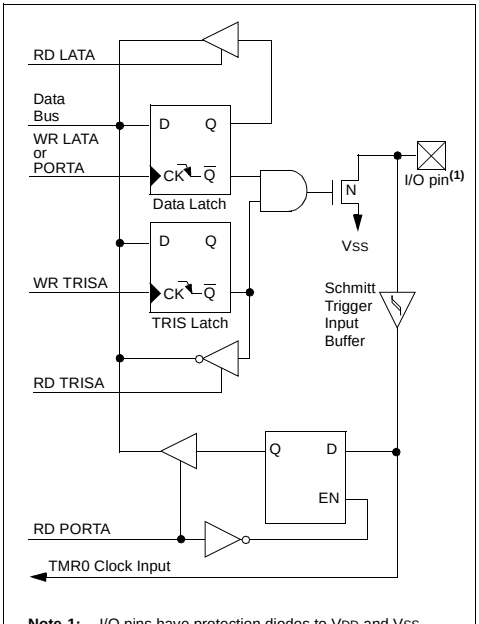
```
CLRF   PORTA    ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF   LATA     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0Fh      ; Configure A/D
MOVWF  ADCON1   ; for digital inputs
MOVLW  0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISA    ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```

FIGURE 10-2: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS



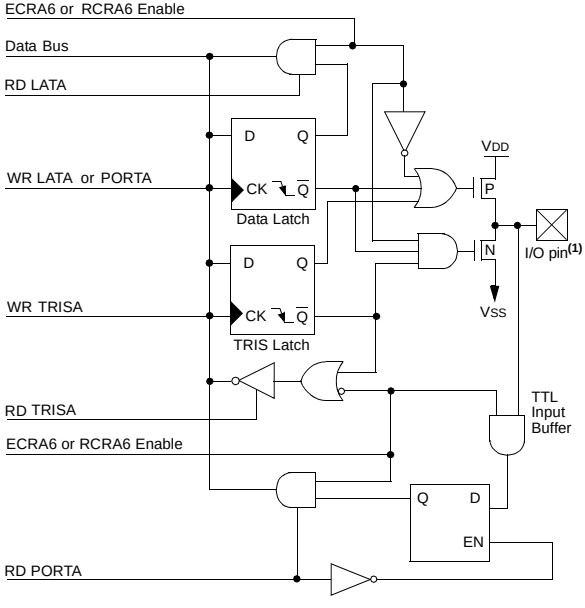
Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-3: BLOCK DIAGRAM OF RA4/T0CKI PIN



Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-4: BLOCK DIAGRAM OF RA6 PIN (WHEN ENABLED AS I/O)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 10-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	TTL	Input/output or analog input.
RA1/AN1	bit 1	TTL	Input/output or analog input.
RA2/AN2/VREF-	bit 2	TTL	Input/output or analog input or VREF-.
RA3/AN3/VREF+	bit 3	TTL	Input/output or analog input or VREF+.
RA4/T0CKI	bit 4	ST/OD	Input/output or external clock input for Timer0. Output is open-drain type.
RA5/AN4/LVDIN	bit 5	TTL	Input/output or slave select input for synchronous serial port or analog input, or Low-Voltage Detect input.
OSC2/CLKO/RA6	bit 6	TTL	OSC2 or clock output, or I/O pin.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 10-2: 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	—	RA6	RA5	RA4	RA3	RA2	RA1	RA0	-x0x 0000	-u0u 0000
LATA	—	LATA Data Output Register							-xxx xxxx	-uuu uuuu
TRISA	—	PORTA Data Direction Register							-111 1111	-111 1111
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	--00 0000

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

10.2 PORTB, TRISB and LATB Registers

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).

The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

EXAMPLE 10-2: INITIALIZING PORTB

```
CLRF    PORTB   ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; 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 $\overline{\text{RBPU}}$ (INTCON2<7>). 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.

Note:	On a Power-on Reset, these pins are configured as digital inputs.

Four of the PORTB pins (RB3:RB0) are the external interrupt pins, INT3 through INT0. In order to use these pins as external interrupts, the corresponding TRISB bit must be set to '1'.

The other four PORTB pins (RB7:RB4) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB port change interrupt with flag bit, RBIF (INTCON<0>).

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

a) Any read or write of PORTB (except with the MOVFF instruction). This will end the mismatch condition.

b) 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.

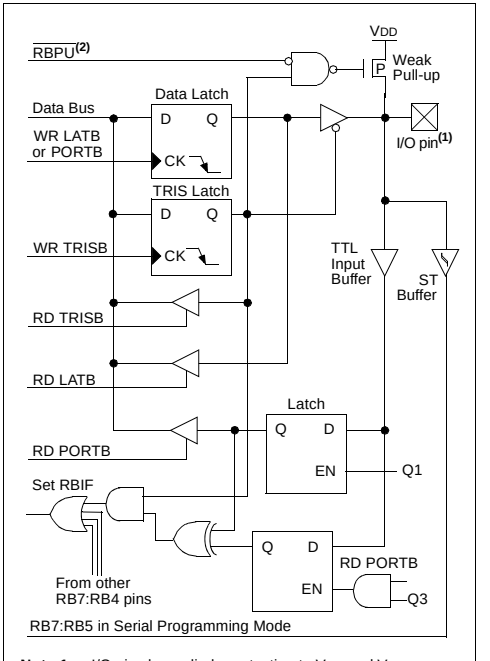
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

For PIC18FXX85 devices, RB3 can be configured by the configuration bit, CCP2MX, as the alternate peripheral pin for the CCP2 module. This is only available when the device is configured in Microprocessor, Microprocessor with Boot Block, or Extended Microcontroller Operating modes.

The RB5 pin is used as the LVP programming pin. When the LVP configuration bit is programmed, this pin loses the I/O function and becomes a programming test function.

Note:	When LVP is enabled, the weak pull-up on RB5 is disabled.

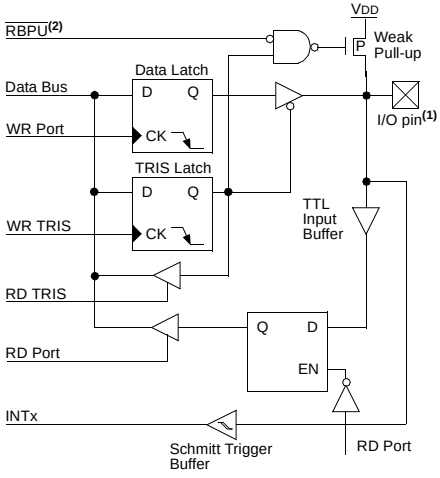
FIGURE 10-5: BLOCK DIAGRAM OF RB7:RB4 PINS



Note 1: I/O pins have diode protection to VDD and VSS.

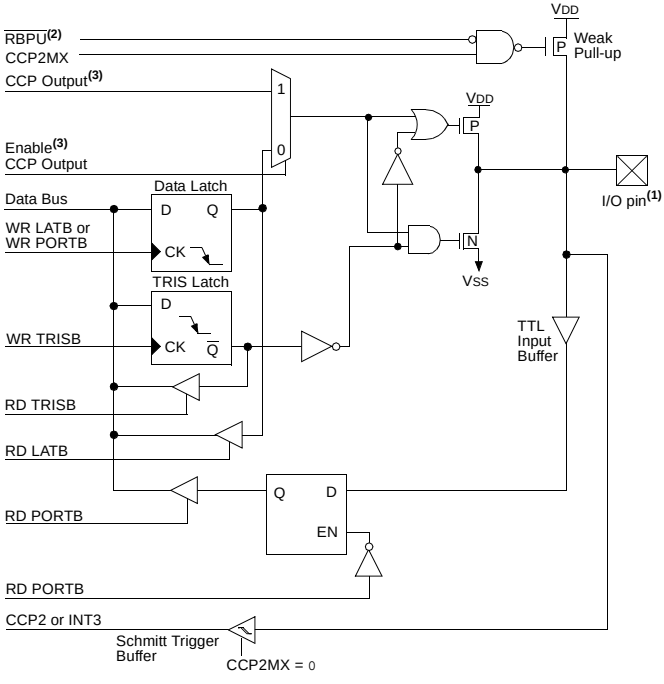
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (INTCON2<7>).

FIGURE 10-6: BLOCK DIAGRAM OF RB2:RB0 PINS



Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

FIGURE 10-7: BLOCK DIAGRAM OF RB3 PIN



Note 1: I/O pin has diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (INTCON2<7>).
3: For PIC18FXX85 parts, the CCP2 input/output is multiplexed with RB3 if the CCP2MX bit is enabled (= 0) in the Configuration register and the device is operating in Microprocessor, Microprocessor with Boot Block or Extended Microcontroller mode.

TABLE 10-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT0	bit 0	TTL/ST(1)	Input/output pin or external interrupt input 0. Internal software programmable weak pull-up.
RB1/INT1	bit 1	TTL/ST(1)	Input/output pin or external interrupt input 1. Internal software programmable weak pull-up.
RB2/INT2	bit 2	TTL/ST(1)	Input/output pin or external interrupt input 2. Internal software programmable weak pull-up.
RB3/INT3/CCP2(3)	bit 3	TTL/ST(4)	Input/output pin or external interrupt input 3. Capture 2 input/ Compare 2 output/PWM output (when CCP2MX configuration bit is enabled, all PIC18FXX85 operating modes except Microcontroller mode). Internal software programmable weak pull-up.
RB4/KBI0	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5/KBI1/PGM	bit 5	TTL/ST(2)	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Low-voltage ICSP enable pin.
RB6/KBI2/PGC	bit 6	TTL/ST(2)	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7/KBI3/PGD	bit 7	TTL/ST(2)	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

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

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

3: RC1 is the alternate assignment for CCP2 when CCP2MX is not set (all operating modes except Microcontroller mode).

4: This buffer is a Schmitt Trigger input when configured as the CCP2 input.

TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
LATB	LATB Data Output Register								xxxx xxxx	uuuu uuuu
TRISB	PORTB Data Direction Register								1111 1111	1111 1111
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	1111 1111
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	1100 0000

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

10.3 PORTC, TRISC and LATC Registers

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

The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is multiplexed with several peripheral functions (Table 10-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC 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. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

Note:	On a Power-on Reset, these pins are configured as digital inputs.

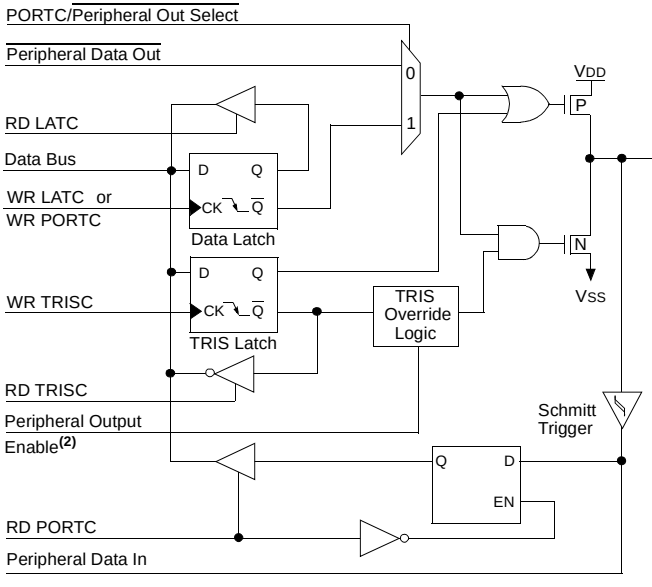
The pin override value is not loaded into the TRIS register. This allows read-modify-write of the TRIS register without concern due to peripheral overrides.

RC1 is normally configured by configuration bit, CCP2MX, as the default peripheral pin of the CCP2 module (default/erased state, CCP2MX = 1).

EXAMPLE 10-3: INITIALIZING PORTC

```
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

FIGURE 10-8: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)



Note 1: I/O pins have diode protection to VDD and VSS.

2: Peripheral output enable is only active if peripheral select is active.

TRIS OVERRIDE

Pin	Override	Peripheral
RC0	Yes	Timer1 Osc for Timer1/Timer3
RC1	Yes	Timer1 Osc for Timer1/Timer3, CCP2 I/O
RC2	Yes	CCP1 I/O
RC3	Yes	SPI/I^2C Master Clock
RC4	Yes	I^2C Data Out
RC5	Yes	SPI Data Out
RC6	Yes	USART Async Xmit, Sync Clock
RC7	Yes	USART Sync Data Out

TABLE 10-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T13CKI	bit 0	ST	Input/output port pin, Timer1 oscillator output or Timer1/Timer3 clock input.
RC1/T1OSI/CCP2(1)	bit 1	ST	Input/output port pin, Timer1 oscillator input or Capture 2 input/ Compare 2 output/PWM output (when CCP2MX configuration bit is disabled).
RC2/CCP1/P1A	bit 2	ST	Input/output port pin or Capture 1 input/Compare 1 output/ PWM1 output.
RC3/SCK/SCL	bit 3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit 4	ST	RC4 can also be the SPI data in (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit 5	ST	Input/output port pin or synchronous serial port data output.
RC6/TX/CK	bit 6	ST	Input/output port pin, addressable USART asynchronous transmit or addressable USART synchronous clock.
RC7/RX/DT	bit 7	ST	Input/output port pin, addressable USART asynchronous receive or addressable USART synchronous data.

Legend: ST = Schmitt Trigger input

Note 1: RB3 is the alternate assignment for CCP2 when CCP2MX is set.

TABLE 10-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

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
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
LATC	LATC Data Output Register								xxxx xxxx	uuuu uuuu
TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

10.4 PORTD, TRISD and LATD Registers

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

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note:	On a Power-on Reset, these pins are configured as digital inputs.

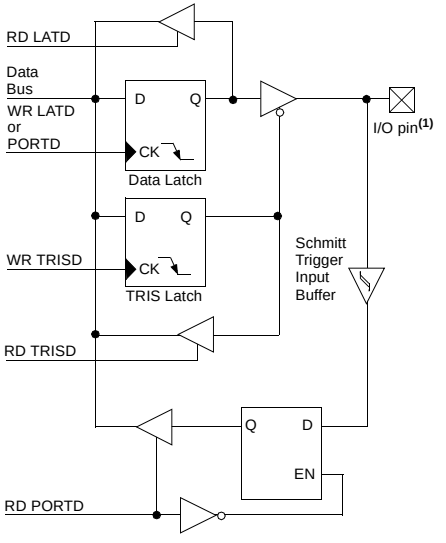
On PIC18F8X8X devices, PORTD is multiplexed with the system bus as the external memory interface; I/O port functions are only available when the system bus is disabled by setting the EBDIS bit in the MEMCON register (MEMCON<7>). When operating as the external memory interface, PORTD is the low-order byte of the multiplexed address/data bus (AD7:AD0).

PORTD can also be configured as an 8-bit wide microprocessor port (Parallel Slave Port) by setting control bit, PSPMODE (TRISE<4>). In this mode, the input buffers are TTL. See **Section 10.10 "Parallel Slave Port (PSP)"** for additional information.

EXAMPLE 10-4: INITIALIZING PORTD

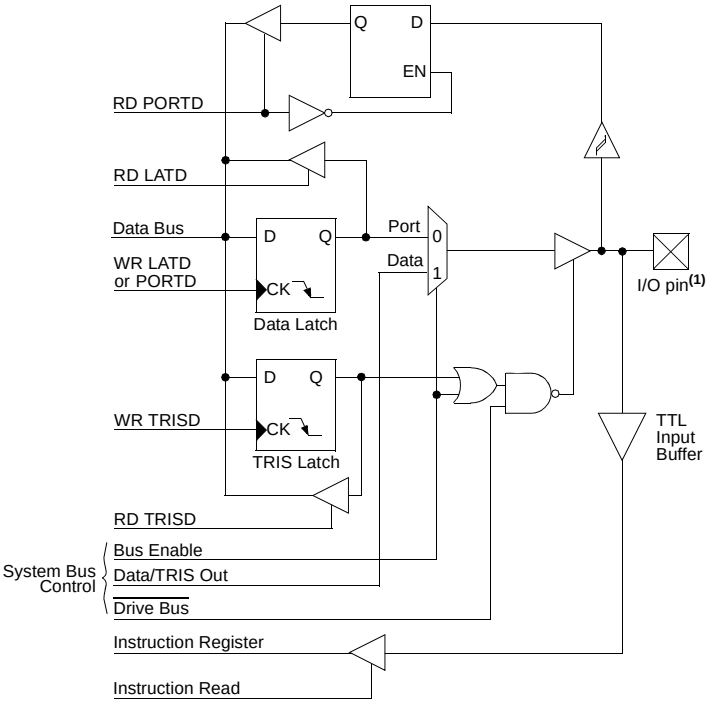
```
CLRF   PORTD   ; Initialize PORTD by
               ; clearing output
               ; data latches
CLRF   LATD    ; Alternate method
               ; to clear output
               ; data latches
MOVLW  0CFh    ; Value used to
               ; initialize data
               ; direction
MOVWF  TRISD   ; Set RD<3:0> as inputs
               ; RD<5:4> as outputs
               ; RD<7:6> as inputs
```

FIGURE 10-9: PORTD BLOCK DIAGRAM IN I/O PORT MODE



Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-10: PORTD BLOCK DIAGRAM IN SYSTEM BUS MODE (PIC18F8X8X ONLY)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 10-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0/AD0(2)	bit 0	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 0 or address/data bus bit 0.
RD1/PSP1/AD1(2)	bit 1	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 1 or address/data bus bit 1.
RD2/PSP2/AD2(2)	bit 2	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 2 or address/data bus bit 2.
RD3/PSP3/AD3(2)	bit 3	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 3 or address/data bus bit 3.
RD4/PSP4/AD4(2)	bit 4	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 4 or address/data bus bit 4.
RD5/PSP5/AD5(2)	bit 5	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 5 or address/data bus bit 5.
RD6/PSP6/AD6(2)	bit 6	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 6 or address/data bus bit 6.
RD7/PSP7/AD7(2)	bit 7	ST/TTL(1)	Input/output port pin, Parallel Slave Port bit 7 or address/data bus bit 7.

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in System Bus or Parallel Slave Port mode.

2: Available in PIC18F8X8X devices only.

TABLE 10-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

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
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
LATD	LATD Data Output Register								xxxx xxxx	uuuu uuuu
TRISD	PORTD Data Direction Register								1111 1111	1111 1111
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	0000 ----
MEMCON	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	0-00 --00

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by PORTD.

10.5 PORTE, TRISE and LATE Registers

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

Read-modify-write operations on the LATE register read and write the latched output value for PORTE.

PORTE is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output. PORTE is multiplexed with the Enhanced CCP module (Table 10-9).

On PIC18F8X8X devices, PORTE is also multiplexed with the system bus as the external memory interface; the I/O bus is available only when the system bus is disabled by setting the EBDIS bit in the MEMCON register (MEMCON<7>). If the device is configured in Microprocessor or Extended Microcontroller mode, then the PORTE<7:0> becomes the high byte of the address/data bus for the external program memory interface. In Microcontroller mode, the PORTE<2:0> pins become the control inputs for the Parallel Slave Port when bit PSPMODE (PSPCON<4>) is set. (Refer to **Section 4.1.1 "PIC18F8X8X Program Memory Modes"** for more information on program memory modes.)

When the Parallel Slave Port is active, three PORTE pins (RE0/$\overline{RD}$/AD8, RE1/$\overline{WR}$/AD9 and RE2/$\overline{CS}$/AD10) function as its control inputs. This automatically occurs when the PSPMODE bit (PSPCON<4>) is set. Users must also make certain that bits TRISE<2:0> are set to configure the pins as digital inputs and the ADCON1 register is configured for digital I/O. The PORTE PSP control functions are summarized in Table 10-9.

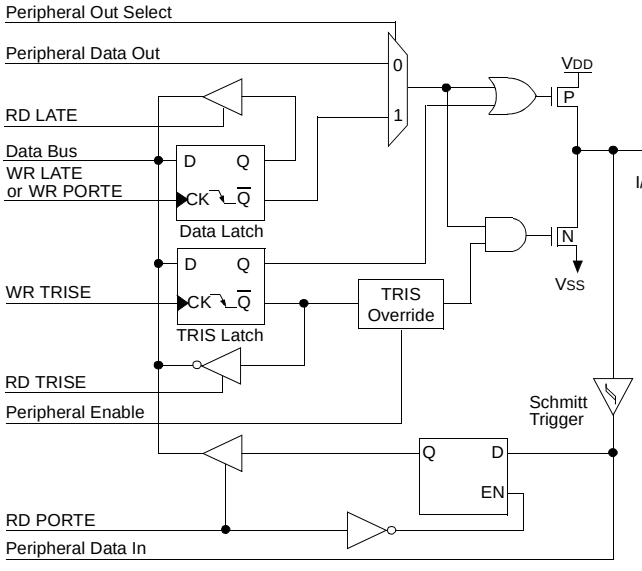
Pin RE7 can be configured as the alternate peripheral pin for the CCP2 module when the device is operating in Microcontroller mode. This is done by clearing the configuration bit, CCP2MX, in configuration register, CONFIG3H (CONFIG3H<0>).

Note:	For PIC18F8X8X (80-pin) devices operating in other than Microcontroller mode, PORTE defaults to the system bus on Power-on Reset.

EXAMPLE 10-5: INITIALIZING PORTE

```
CLRF    PORTE   ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   03h     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISE   ; Set RE1:RE0 as inputs
                ; RE7:RE2 as outputs
```

FIGURE 10-11: PORTE BLOCK DIAGRAM IN I/O MODE

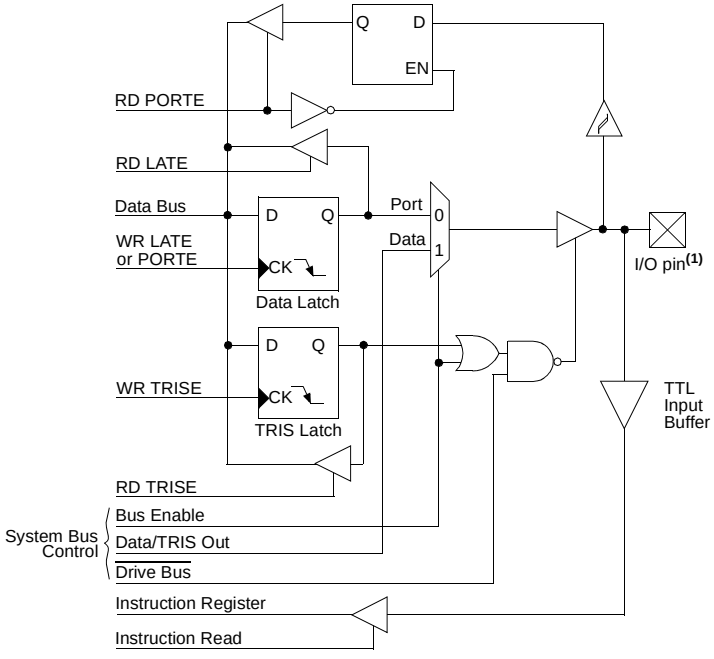


TRIS OVERRIDE

Pin	Override	Peripheral
RE0	Yes	External Bus
RE1	Yes	External Bus
RE2	Yes	External Bus
RE3	Yes	External Bus
RE4	Yes	External Bus
RE5	Yes	External Bus
RE6	Yes	External Bus
RE7	Yes	External Bus

Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-12: PORTE BLOCK DIAGRAM IN SYSTEM BUS MODE (PIC18F8X8X ONLY)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 10-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/$\overline{RD}$/AD8(2)	bit 0	ST/TTL(1)	Input/output port pin, read control for Parallel Slave Port or address/data bit 8. For $\overline{RD}$ (PSP Control mode): 1 = Not a read operation 0 = Read operation, reads PORTD register (if chip selected)
RE1/$\overline{WR}$/AD9(2)	bit 1	ST/TTL(1)	Input/output port pin, write control for Parallel Slave Port or address/data bit 9. For $\overline{WR}$ (PSP Control mode): 1 = Not a write operation 0 = Write operation, writes PORTD register (if chip selected)
RE2/$\overline{CS}$/AD10(2)	bit 2	ST/TTL(1)	Input/output port pin, chip select control for Parallel Slave Port or address/data bit 10. For $\overline{CS}$ (PSP Control mode): 1 = Device is not selected 0 = Device is selected
RE3/AD11(2)	bit 3	ST/TTL(1)	Input/output port pin or address/data bit 11.
RE4/AD12(2)	bit 4	ST/TTL(1)	Input/output port pin or address/data bit 12.
RE5/AD13/(2)P1C(3)	bit 5	ST/TTL(1)	Input/output port pin, address/data bit 13 or ECCP1 PWM output C.
RE6/AD14/(2)P1B(3)	bit 6	ST/TTL(1)	Input/output port pin, address/data bit 13 or ECCP1 PWM output B.
RE7/CCP2/AD15(2)	bit 7	ST/TTL(1)	Input/output port pin, Capture 2 input/Compare 2 output/PWM output (PIC18F8X20 devices in Microcontroller mode only) or address/data bit 15.

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O or CCP mode, and TTL buffers when in System Bus or PSP Control mode.

2: Available in PIC18F8X8X devices only.

3: On PIC18F8X8X devices, these pins may be moved to RHY or RH6 by changing the ECCPMX configuration bit.

TABLE 10-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

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
TRISE	PORTE Data Direction Control Register								1111 1111	1111 1111
PORTE	Read PORTE pin/Write PORTE Data Latch								xxxx xxxx	uuuu uuuu
LATE	Read PORTE Data Latch/Write PORTE Data Latch								xxxx xxxx	uuuu uuuu
MEMCON	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	0000 --00
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	0000 ----

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTE.

10.6 PORTF, LATF and TRISF Registers

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

Read-modify-write operations on the LATF register read and write the latched output value for PORTF.

PORTF is multiplexed with several analog peripheral functions, including the A/D converter inputs and comparator inputs, outputs, and voltage reference.

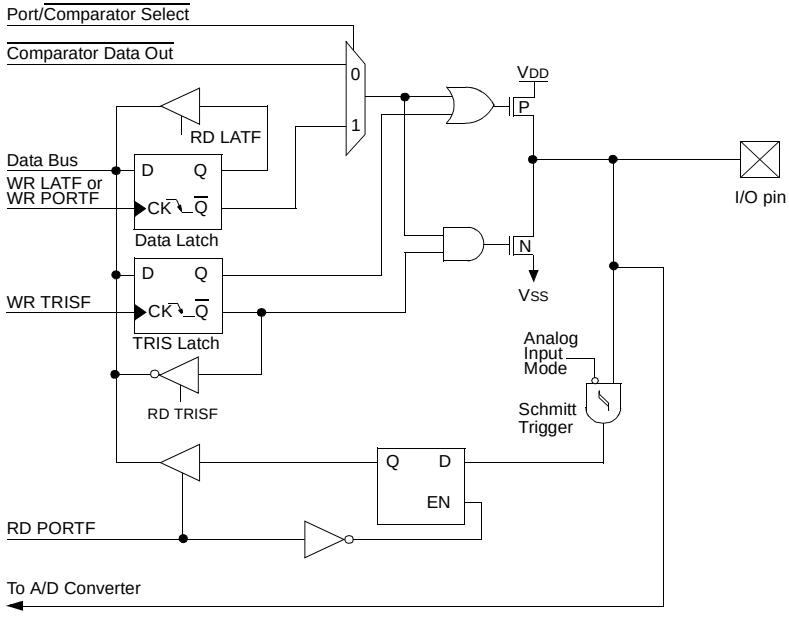
Note 1: On a Power-on Reset, the RF6:RF0 pins are configured as inputs and read as '0'.

2: To configure PORTF as digital I/O, turn off comparators and set ADCON1 value.

EXAMPLE 10-6: INITIALIZING PORTF

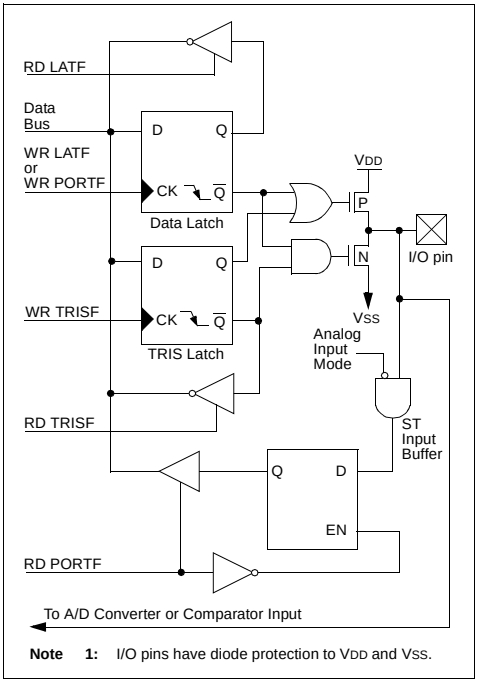
```
CLRF    PORTF       ; Initialize PORTF by
                    ; clearing output
                    ; data latches
CLRF    LATF        ; Alternate method
                    ; to clear output
                    ; data latches
MOVLW   07h         ;
MOVWF   CMCON       ; Turn off comparators
MOVLW   0Fh         ;
MOVWF   ADCON1      ; Set PORTF as digital I/O
MOVLW   0CFh        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISF       ; Set RF3:RF0 as inputs
                    ; RF5:RF4 as outputs
                    ; RF7:RF6 as inputs
```

FIGURE 10-13: PORTF RF1/AN6/C2OUT AND RF2/AN7/C1OUT PINS BLOCK DIAGRAM



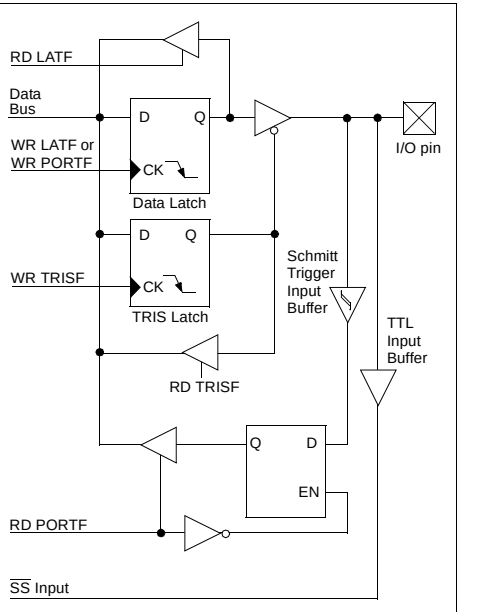
Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-14: RF6:RF3 AND RF0 PINS BLOCK DIAGRAM



Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-15: RF7 PIN BLOCK DIAGRAM



Note: I/O pins have diode protection to VDD and VSS.

TABLE 10-11: PORTF FUNCTIONS

Name	Bit#	Buffer Type	Function
RF0/AN5	bit 0	ST	Input/output port pin or analog input.
RF1/AN6/C2OUT	bit 1	ST	Input/output port pin, analog input or comparator 2 output.
RF2/AN7/C1OUT	bit 2	ST	Input/output port pin, analog input or comparator 1 output.
RF3/AN8/C2IN+	bit 3	ST	Input/output port pin, analog input or comparator 2 input (+).
RF4/AN9/C2IN-	bit 4	ST	Input/output port pin, analog input or comparator 2 input (-).
RF5/AN10/ C1IN+/CVREF	bit 5	ST	Input/output port pin, analog input, comparator 1 input (+) or comparator reference output.
RF6/AN11/C1IN-	bit 6	ST	Input/output port pin, analog input or comparator 1 input (-).
RF7/$\overline{SS}$	bit 7	ST/TTL	Input/output port pin or slave select pin for synchronous serial port.

Legend: ST = Schmitt Trigger input, TTL = TTL input

TABLE 10-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

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
TRISF	PORTF Data Direction Control Register								1111 1111	1111 1111
PORTF	Read PORTF pin/Write PORTF Data Latch								xxxx xxxx	uuuu uuuu
LATF	Read PORTF Data Latch/Write PORTF Data Latch								0000 0000	uuuu uuuu
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	--00 0000
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	0000 0000

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTF.

10.7 PORTG, TRISG and LATG Registers

PORTG is a 6-bit wide port with 5 bidirectional pins and 1 unidirectional pin. The corresponding data direction register is TRISG. Setting a TRISG bit (= 1) will make the corresponding PORTG pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISG bit (= 0) will make the corresponding PORTG pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATG) is also memory mapped. Read-modify-write operations on the LATG register read and write the latched output value for PORTG.

Pins RG0-RG2 on PORTG are multiplexed with the CAN peripheral. Refer to **Section 23.0 "ECAN Module"** for proper settings of TRISG when CAN is enabled. RG5 is multiplexed with MCLR/VPP. Refer to Register 24-5 for more information.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTG 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. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

Note:	On a Power-on Reset, these pins are configured as digital inputs.

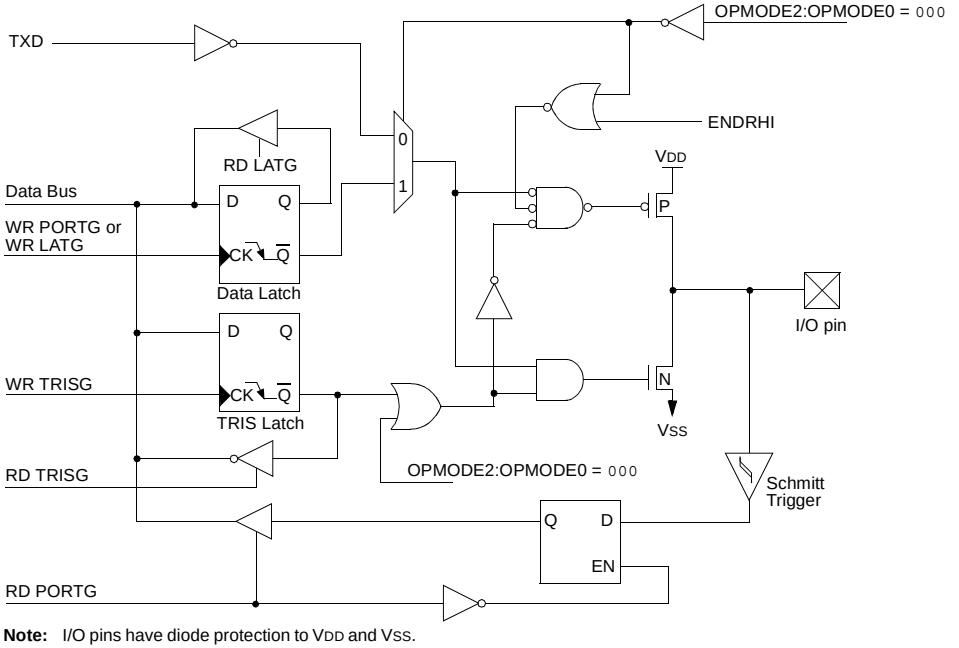
The pin override value is not loaded into the TRIS register. This allows read-modify-write of the TRIS register without concern due to peripheral overrides.

EXAMPLE 10-7: INITIALIZING PORT

```
CLRF   PORTG    ; Initialize PORTG by
                ; clearing output
                ; data latches
CLRF   LATG     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  04h      ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISG    ; Set RG1:RG0 as outputs
                ; RG2 as input
                ; RG4:RG3 as inputs
```

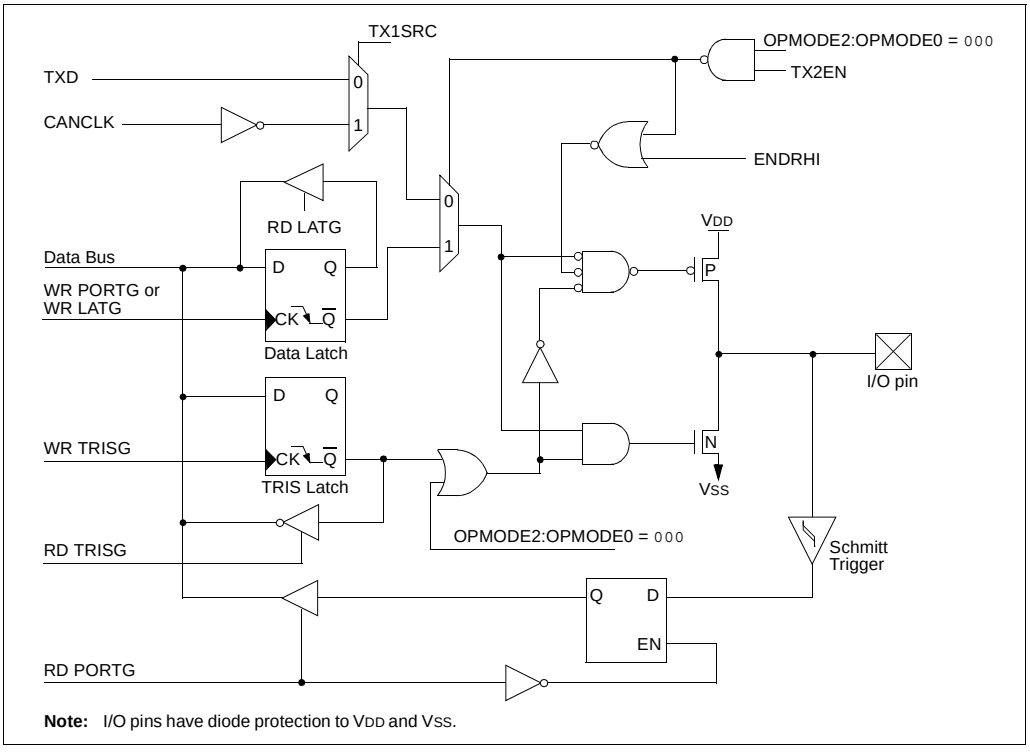
Note 1:	On a Power-on Reset, RG5 is enabled as a digital input only if Master Clear functionality is disabled (MCLRE = 0).
2:	If the device Master Clear is disabled, verify that either of the following is done to ensure proper entry into ICSP mode: a) disable Low-Voltage Programming (CONFIG4L<2> = 0); or b) make certain that RB5/KBI1/PGM is held low during entry into ICSP.

FIGURE 10-16: RG0/CANTX1 PIN BLOCK DIAGRAM



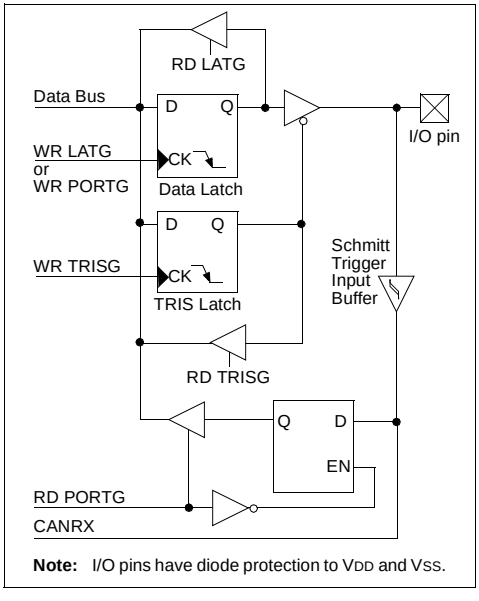
Note: I/O pins have diode protection to VDD and VSS.

FIGURE 10-17: RG1/CANTX2 PIN BLOCK DIAGRAM



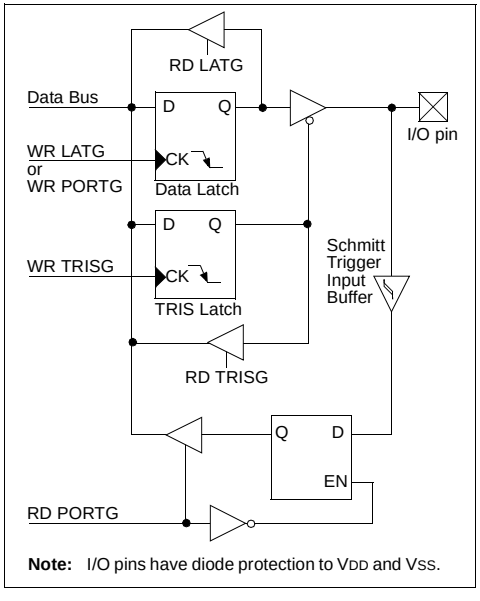
Note: I/O pins have diode protection to VDD and VSS.

FIGURE 10-18: RG2/CANRX PIN BLOCK DIAGRAM



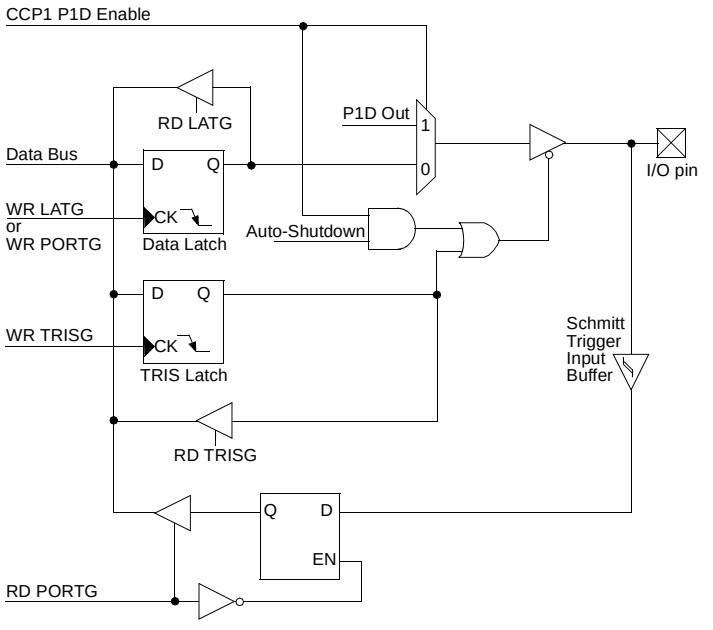
Note: I/O pins have diode protection to VDD and VSS.

FIGURE 10-19: RG3 PIN BLOCK DIAGRAM



Note: I/O pins have diode protection to VDD and VSS.

FIGURE 10-20: RG4/P1D PIN BLOCK DIAGRAM



Note: I/O pins have diode protection to VDD and VSS.

FIGURE 10-21: RG5/$\overline{\text{MCLR}}$/VPP PIN BLOCK DIAGRAM

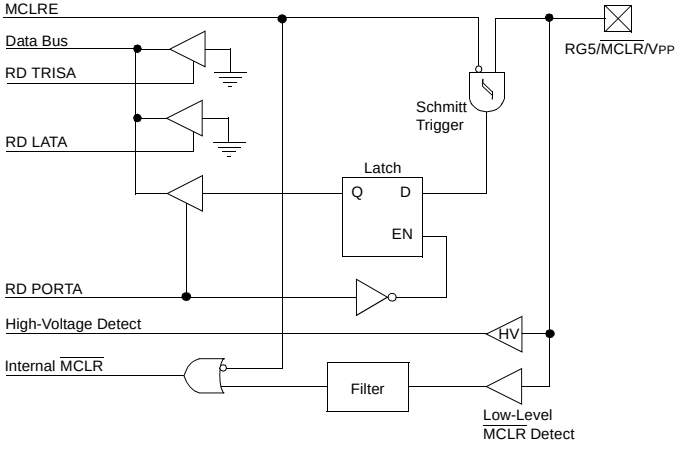


TABLE 10-13: PORTG FUNCTIONS

Name	Bit#	Buffer Type	Function
RG0/CANTX1	bit 0	ST	Input/output port pin or CAN bus transmit output.
RG1/CANTX2	bit 1	ST	Input/output port pin, CAN bus complimentary transmit output or CAN bus bit time clock.
RG2/CANRX	bit 2	ST	Input/output port pin or CAN bus receive.
RG3	bit 3	ST	Input/output port pin.
RG4/P1D	bit 4	ST	Input/output port pin or ECCP1 PWM output D.
RG5/$\overline{MCLR}$/V_{PP}	bit 5	ST	Master Clear input or programming voltage input (if $\overline{MCLR}$ is enabled). Input only port pin or programming voltage input (if $\overline{MCLR}$ is disabled).

Legend: ST = Schmitt Trigger input

TABLE 10-14: SUMMARY OF REGISTERS ASSOCIATED WITH PORTG

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
PORTG	—	—	RG5(1)	Read PORTF pin/Write PORTF Data Latch					--0x xxxx	--0u uuuu
LATG	—	—	—	LATG Data Output Register					---x xxxx	---u uuuu
TRISG	—	—	—	Data Direction Control Register for PORTG					---1 1111	---1 1111

Legend: x = unknown, u = unchanged

Note 1: RG5 is available as an input only when $\overline{MCLR}$ is disabled.

10.8 PORTH, LATH and TRISH Registers

Note:	PORTH is available only on PIC18F8X8X devices.

PORTH is an 8-bit wide, bidirectional I/O port. The corresponding data direction register is TRISH. Setting a TRISH bit (= 1) will make the corresponding PORTH pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISH bit (= 0) will make the corresponding PORTH pin an output (i.e., put the contents of the output latch on the selected pin).

Read-modify-write operations on the LATH register read and write the latched output value for PORTH.

Pins RH7:RH4 are multiplexed with analog inputs AN15:AN12. Pins RH3:RH0 are multiplexed with the system bus as the external memory interface; they are the high-order address bits, A19:A16. By default, pins RH7:RH4 are enabled as A/D inputs and pins RH3:RH0 are enabled as the system address bus. Register ADCON1 configures RH7:RH4 as I/O or A/D inputs. Register MEMCON configures RH3:RH0 as I/O or system bus pins.

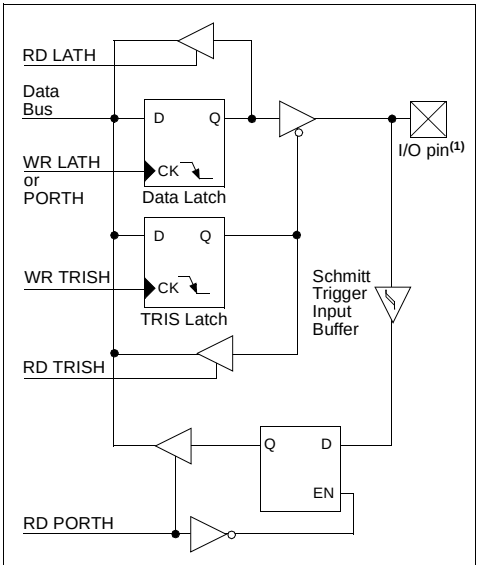
Pins RH7 and RH6 can be configured as the alternate peripheral pins for CCP1 PWM output P1B and P1C, respectively. This is done by clearing the configuration bit ECCPMX, in configuration register CONFIG3H (CONFIG3H<1>).

Note 1:	On Power-on Reset, PORTH pins RH7:RH4 default to A/D inputs and read as '0'.
2:	On Power-on Reset, PORTH pins RH3:RH0 default to system bus signals.

EXAMPLE 10-8: INITIALIZING PORTH

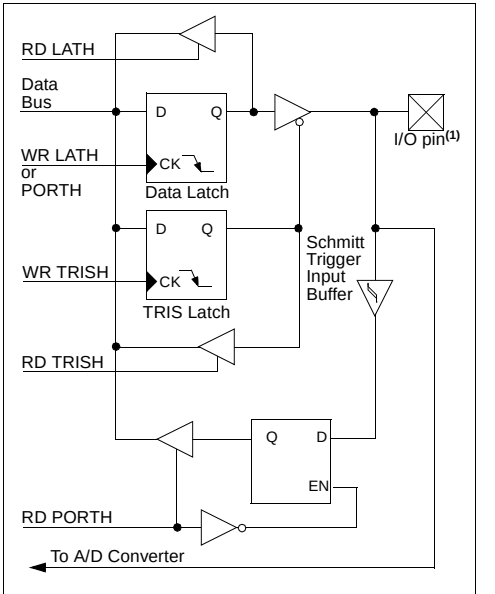
```
CLRF    PORTH       ; Initialize PORTH by
                    ; clearing output
                    ; data latches
CLRF    LATH        ; Alternate method
                    ; to clear output
                    ; data latches
MOVLW   0Fh         ;
MOVWF   ADCON1      ;
MOVLW   0CFh        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISH       ; Set RH3:RH0 as inputs
                    ; RH5:RH4 as outputs
                    ; RH7:RH6 as inputs
```

FIGURE 10-22: RH3:RH0 PINS BLOCK DIAGRAM IN I/O MODE



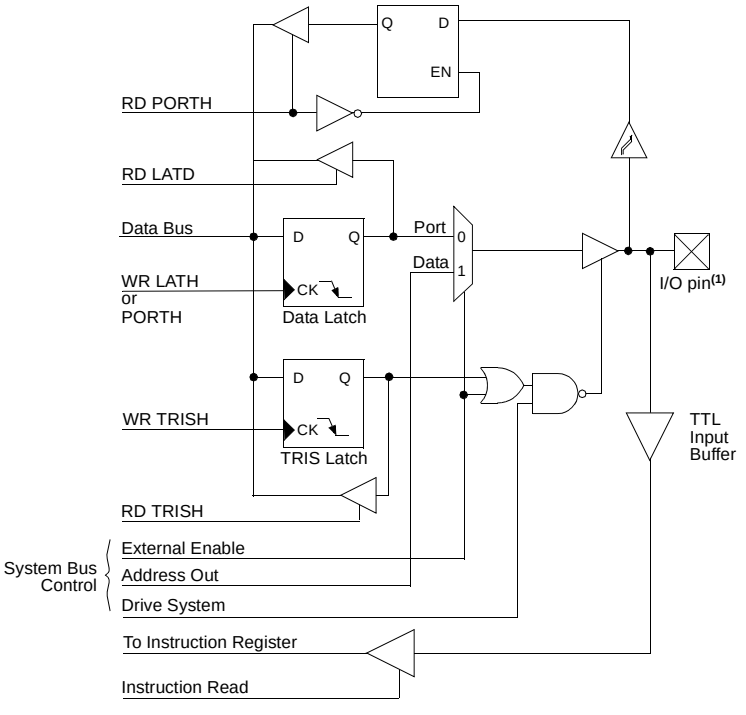
Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-23: RH7:RH4 PINS BLOCK DIAGRAM IN I/O MODE



Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-24: RH3:RH0 PINS BLOCK DIAGRAM IN SYSTEM BUS MODE



Note 1: I/O pins have diode protection to VDD and VSS.

TABLE 10-15: PORTH FUNCTIONS

Name	Bit#	Buffer Type	Function
RH0/A16	bit 0	ST/TTL(1)	Input/output port pin or address bit 16 for external memory interface.
RH1/A17	bit 1	ST/TTL(1)	Input/output port pin or address bit 17 for external memory interface.
RH2/A18	bit 2	ST/TTL(1)	Input/output port pin or address bit 18 for external memory interface.
RH3/A19	bit 3	ST/TTL(1)	Input/output port pin or address bit 19 for external memory interface.
RH4/AN12	bit 4	ST	Input/output port pin or analog input channel 12.
RH5/AN13	bit 5	ST	Input/output port pin or analog input channel 13.
RH6/AN14/P1C(2)	bit 6	ST	Input/output port pin or analog input channel 14.
RH7/AN15/P1B(2)	bit 7	ST	Input/output port pin or analog input channel 15.

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in System Bus or Parallel Slave Port mode.

2: Alternate pin assignment when ECCPMX configuration bit is cleared.

TABLE 10-16: SUMMARY OF REGISTERS ASSOCIATED WITH PORTH

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
TRISH	PORTH Data Direction Control Register								1111 1111	1111 1111
PORTH	Read PORTH pin/Write PORTH Data Latch								xxxx xxxx	uuuu uuuu
LATH	Read PORTH Data Latch/Write PORTH Data Latch								xxxx xxxx	uuuu uuuu
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	--00 0000
MEMCON(1)	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	0-00 --00

Legend: x = unknown, u = unchanged, – = unimplemented. Shaded cells are not used by PORTH.

Note 1: This register is held in Reset in Microcontroller mode.

10.9 PORTJ, TRISJ and LATJ Registers

Note:	PORTJ is available only on PIC18F8X8X devices.

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

The Data Latch register (LATJ) is also memory mapped. Read-modify-write operations on the LATJ register read and write the latched output value for PORTJ.

PORTJ is multiplexed with the system bus as the external memory interface; I/O port functions are only available when the system bus is disabled. When operating as the external memory interface, PORTJ provides the control signal to external memory devices. The RJ5 pin is not multiplexed with any system bus functions.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTJ 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. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

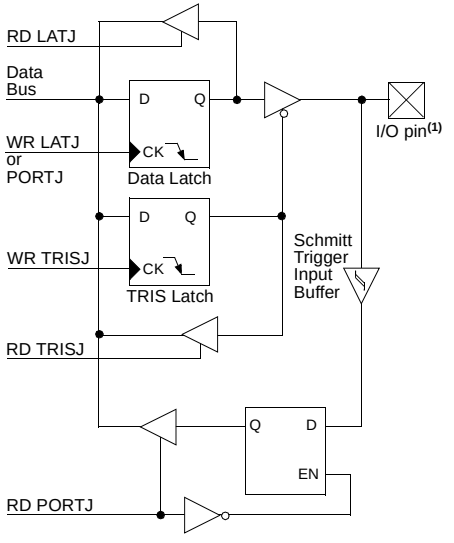
Note:	On a Power-on Reset, these pins are configured as digital inputs.

The pin override value is not loaded into the TRIS register. This allows read-modify-write of the TRIS register without concern due to peripheral overrides.

EXAMPLE 10-9: INITIALIZING PORTJ

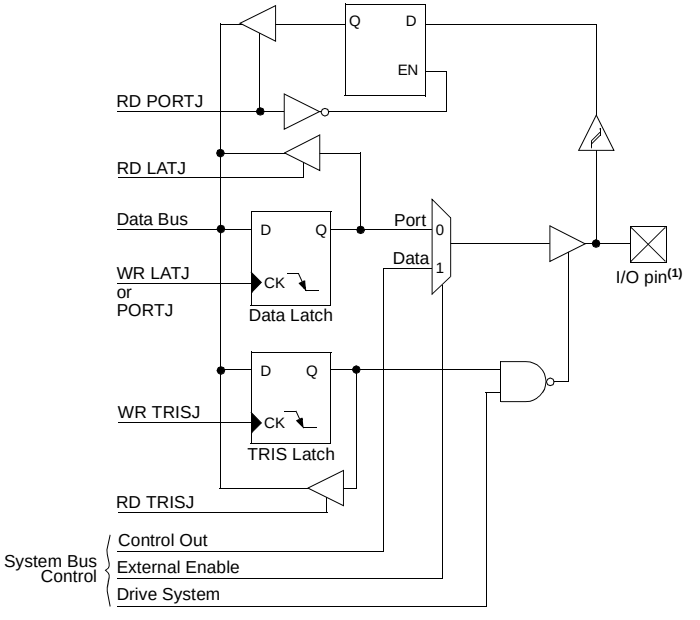
```
CLRF   PORTJ    ; Initialize PORTG by
                ; clearing output
                ; data latches
CLRF   LATJ     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISJ    ; Set RJ3:RJ0 as inputs
                ; RJ5:RJ4 as output
                ; RJ7:RJ6 as inputs
```

FIGURE 10-25: PORTJ BLOCK DIAGRAM IN I/O MODE



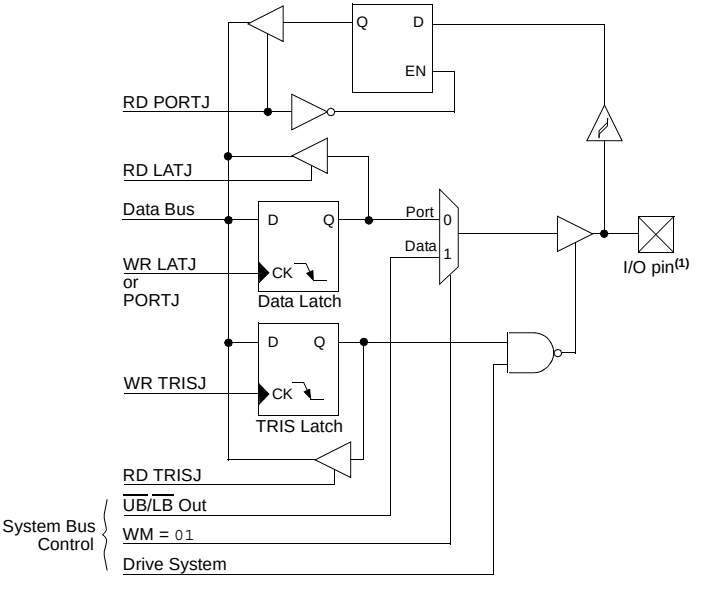
Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-26: RJ5:RJ0 PINS BLOCK DIAGRAM IN SYSTEM BUS MODE



Note 1: I/O pins have diode protection to VDD and VSS.

FIGURE 10-27: RJ7:RJ6 PINS BLOCK DIAGRAM IN SYSTEM BUS MODE



Note 1: I/O pins have diode protection to VDD and VSS.

TABLE 10-17: PORTJ FUNCTIONS

Name	Bit#	Buffer Type	Function
RJ0/ALE	bit 0	ST	Input/output port pin or address latch enable control for external memory interface.
RJ1/$\overline{OE}$	bit 1	ST	Input/output port pin or output enable control for external memory interface.
RJ2/$\overline{WRL}$	bit 2	ST	Input/output port pin or write low byte control for external memory interface.
RJ3/$\overline{WRH}$	bit 3	ST	Input/output port pin or write high byte control for external memory interface.
RJ4/BA0	bit 4	ST	Input/output port pin or byte address 0 control for external memory interface.
RJ5/$\overline{CE}$	bit 5	ST	Input/output port pin or external memory chip enable.
RJ6/$\overline{LB}$	bit 6	ST	Input/output port pin or lower byte select control for external memory interface.
RJ7/$\overline{UB}$	bit 7	ST	Input/output port pin or upper byte select control for external memory interface.

Legend: ST = Schmitt Trigger input

TABLE 10-18: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ

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
PORTJ	Read PORTJ pin/Write PORTJ Data Latch								xxxx xxxx	uuuu uuuu
LATJ	LATJ Data Output Register								xxxx xxxx	uuuu uuuu
TRISJ	Data Direction Control Register for PORTJ								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

10.10 Parallel Slave Port (PSP)

PORTD also operates as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (TRISE<4>) is set. It is asynchronously readable and writable by the external world through $\overline{RD}$ control input pin, RE0/$\overline{RD}$/AD8 and $\overline{WR}$ control input pin, RE1/$\overline{WR}$/AD9.

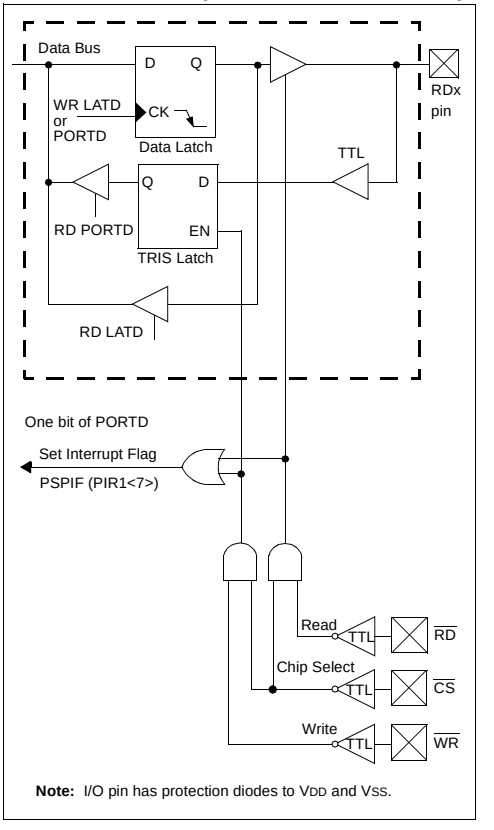
Note:	For PIC18F8X8X devices, the Parallel Slave Port is available only in Microcontroller mode.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$/AD8 to be the $\overline{RD}$ input, RE1/$\overline{WR}$/AD9 to be the $\overline{WR}$ input and RE2/$\overline{CS}$/AD10 to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port configuration bits PCFG2:PCFG0 (ADCON1<2:0>) must be set, which will configure pins RE2:RE0 as digital I/O.

A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low.

The PORTE I/O pins become control inputs for the microprocessor port when bit PSPMODE (PSPCON<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs) and the ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

FIGURE 10-28: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



REGISTER 10-1: PSPCON REGISTER

R-0	R-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IBF	OBF	IBOV	PSPMODE	—	—	—	—
bit 7							bit 0

bit 7 **IBF:** Input Buffer Full Status bit
1 = A data byte has been received and is waiting to be read by the CPU
0 = No data byte has been received

bit 6 **OBF:** Output Buffer Full Status bit
1 = The output buffer still holds a previously written data byte
0 = The output buffer has been read

bit 5 **IBOV:** Input Buffer Overflow Detect bit
1 = A write occurred when a previously input data byte has not been read (must be cleared in software)
0 = No overflow occurred

bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit
1 = Parallel Slave Port mode
0 = General Purpose I/O mode

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

FIGURE 10-29: PARALLEL SLAVE PORT WRITE WAVEFORMS

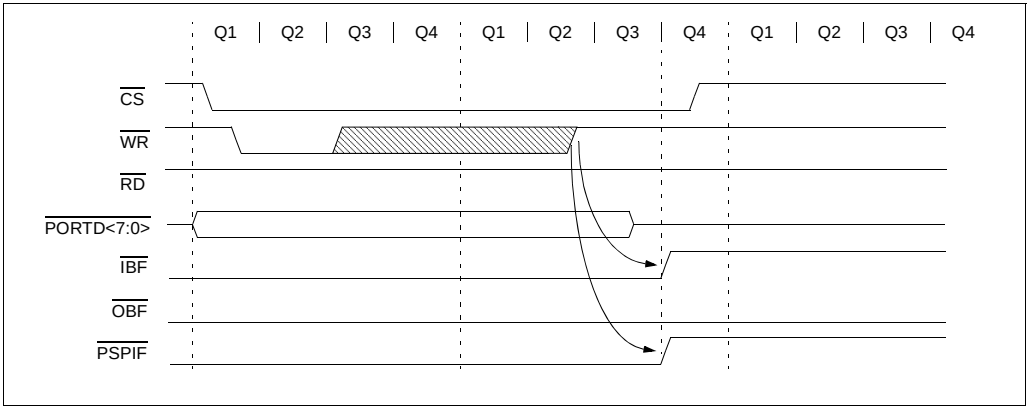


FIGURE 10-30: PARALLEL SLAVE PORT READ WAVEFORMS

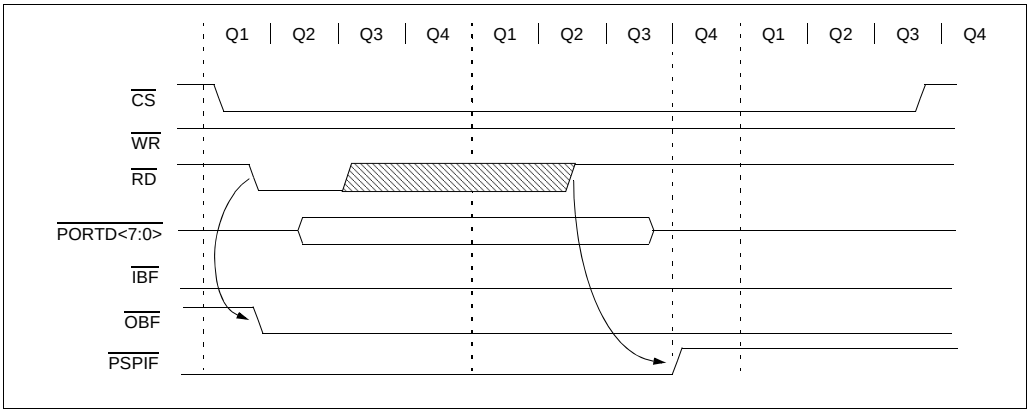


TABLE 10-19: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

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
PORTD	Port Data Latch when Written; Port pins when Read								xxxx xxxx	uuuu uuuu
LATD	LATD Data Output bits								xxxx xxxx	uuuu uuuu
TRISD	PORTD Data Direction bits								1111 1111	1111 1111
PORTE	RE7/CCP2/ AD15	RE6/AD14/ P1B	RE5/AD13/ P1C	RE4/ AD12	RE3/ AD11	RE2/$\overline{CS}$(1)/ AD10	RE1/$\overline{WR}$(1)/ AD9	RE0/$\overline{RD}$(1)/ AD8	xxxx xxxx	uuuu uuuu
LATE	LATE Data Output bits								xxxx xxxx	uuuu uuuu
TRISE	PORTE Data Direction bits								1111 1111	1111 1111
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	0000 ----
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IF	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Parallel Slave Port.

Note 1: Enabled only in Microcontroller mode.

11.0 TIMER0 MODULE

The Timer0 module has the following features:

- Software selectable as an 8-bit or 16-bit timer/counter
- Readable and writable
- Dedicated 8-bit software programmable prescaler
- Clock source selectable to be external or internal
- Interrupt-on-overflow from 0FFh to 00h in 8-bit mode and 0FFFFh to 0000h in 16-bit mode
- Edge select for external clock

Figure 11-1 shows a simplified block diagram of the Timer0 module in 8-bit mode and Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

The T0CON register (Register 11-1) is a readable and writable register that controls all the aspects of Timer0, including the prescale selection.

Note:	Timer0 is enabled on POR.

REGISTER 11-1: T0CON: TIMER0 CONTROL 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
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-bit/16-bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

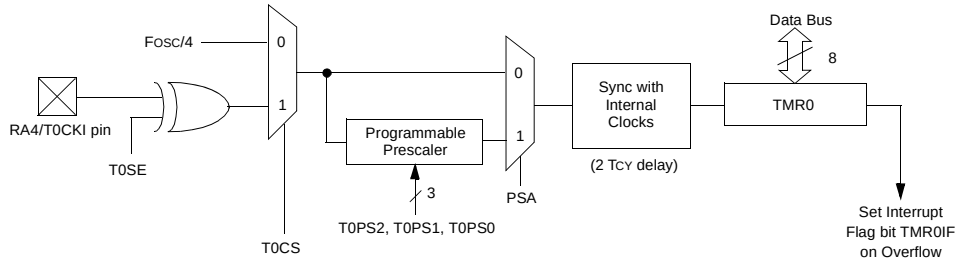
bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits
111 = 1:256 prescale value
110 = 1:128 prescale value
101 = 1:64 prescale value
100 = 1:32 prescale value
011 = 1:16 prescale value
010 = 1:8 prescale value
001 = 1:4 prescale value
000 = 1:2 prescale value

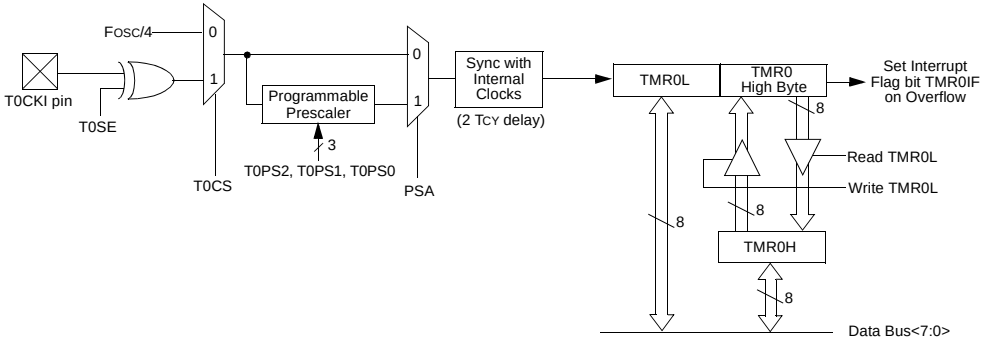
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

FIGURE 11-1: TIMER0 BLOCK DIAGRAM IN 8-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

FIGURE 11-2: TIMER0 BLOCK DIAGRAM IN 16-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

11.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

Timer mode is selected by clearing the T0CS bit. In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit. In Counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit (T0SE). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed below.

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

11.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not readable or writable.

The PSA and T0PS2:T0PS0 bits determine the prescaler assignment and prescale ratio.

Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0, x`, ..., etc.) will clear the prescaler count.

Note:	Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

11.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution).

11.3 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from 0FFh to 00h in 8-bit mode, or 0FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF bit. The interrupt can be masked by clearing the TMR0IE bit. The TMR0IE bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from Sleep since the timer is shut-off during Sleep.

11.4 16-Bit Mode Timer Reads and Writes

TMR0H is not the high byte of the timer/counter in 16-bit mode, but is actually a buffered version of the high byte of Timer0 (refer to Figure 11-2). The high byte of the Timer0 counter/timer is not directly readable nor writable. TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid due to a rollover between successive reads of the high and low byte.

A write to the high byte of Timer0 must also take place through the TMR0H buffer register. Timer0 high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0

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
TMR0L	Timer0 Module Low Byte Register								xxxx xxxx	uuuu uuuu
TMR0H	Timer0 Module High Byte Register								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111
TRISA	—	PORTA Data Direction Register							-111 1111	-111 1111

Legend: x = unknown, u = unchanged, – = unimplemented locations, read as ‘0’. Shaded cells are not used by Timer0.

NOTES:

12.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter
  (two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt on overflow from 0FFFFh to 0000h
- Reset from CCP module special event trigger

Figure 12-1 is a simplified block diagram of the Timer1 module.

Register 12-1 details the Timer1 Control register. This register controls the operating mode of the Timer1 module and contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7 **RD16:** 16-bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **Unimplemented:** Read as '0'

bit 5-4 **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits
11 = 1:8 prescale value
10 = 1:4 prescale value
01 = 1:2 prescale value
00 = 1:1 prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut-off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

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

12.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

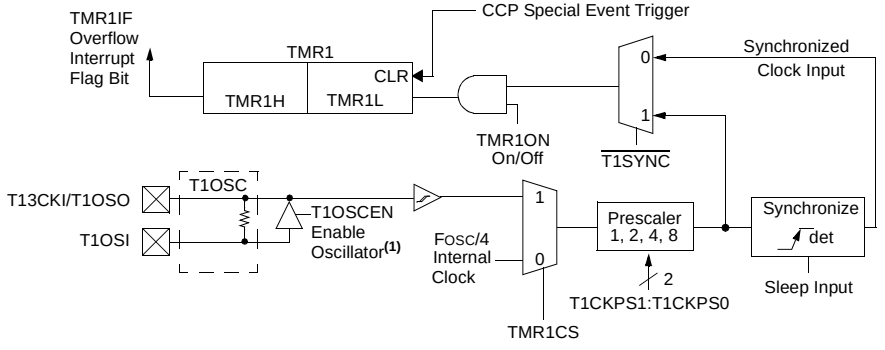
The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

When TMR1CS = 0, Timer1 increments every instruction cycle. When TMR1CS = 1, Timer1 increments on every rising edge of the external clock input or the Timer1 oscillator if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. That is, the TRISC<1:0> value is ignored and the pins are read as '0'.

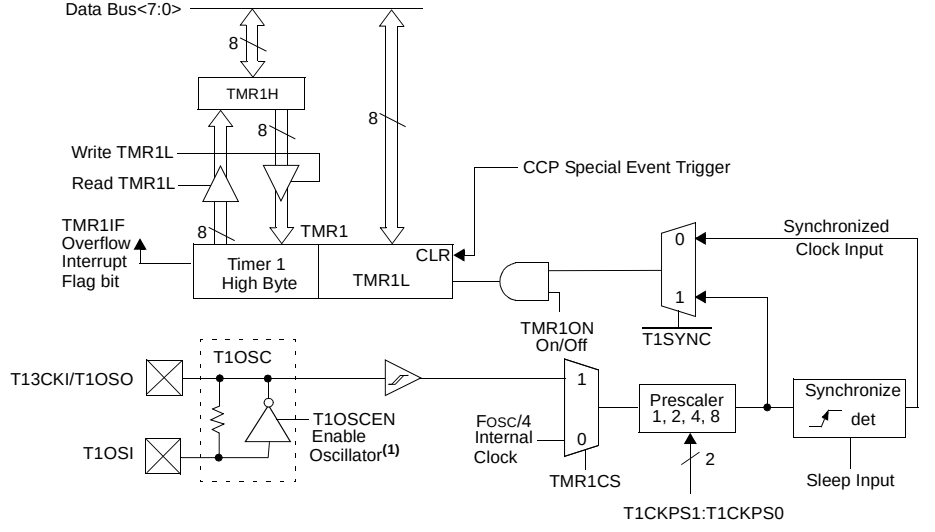
Timer1 also has an internal "Reset input". This Reset can be generated by the CCP module (**Section 15.0 "Capture/Compare/PWM (CCP) Modules"**).

FIGURE 12-1: TIMER1 BLOCK DIAGRAM



Note 1: When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

FIGURE 12-2: TIMER1 BLOCK DIAGRAM: 16-BIT READ/WRITE MODE



Note 1: When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

12.2 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit, T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator rated up to 200 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 12-1 shows the capacitor selection for the Timer1 oscillator.

The user must provide a software time delay to ensure proper start-up of the Timer1 oscillator.

TABLE 12-1: CAPACITOR SELECTION FOR THE ALTERNATE OSCILLATOR

Osc Type	Freq	C1	C2
LP	32 kHz	TBD(1)	TBD(1)
Crystal to be Tested:			
32.768 kHz	Epson C-001R32.768K-A		± 20 PPM

Note 1: Microchip suggests 33 pF as a starting point in validating the oscillator circuit.

2: Higher capacitance increases the stability of the oscillator but also increases the start-up time.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

4: Capacitor values are for design guidance only.

12.3 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to 0FFFFh and rolls over to 0000h. The TMR1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit, TMR1IE (PIE1<0>).

12.4 Resetting Timer1 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a "special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

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

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

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

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

12.5 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 high byte buffer. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, is valid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0111 1111	0111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu

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

13.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Timer2 has a control register shown in Register 13-1. Timer2 can be shut-off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption. Figure 13-1 is a simplified block diagram of the Timer2 module. Register 13-1 shows the Timer2 Control register. The prescaler and postscaler selection of Timer2 are controlled by this register.

13.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 register is readable and writable and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt latched in flag bit, TMR2IF (PIR1<1>).

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
•
•
•
1111 = 1:16 postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

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

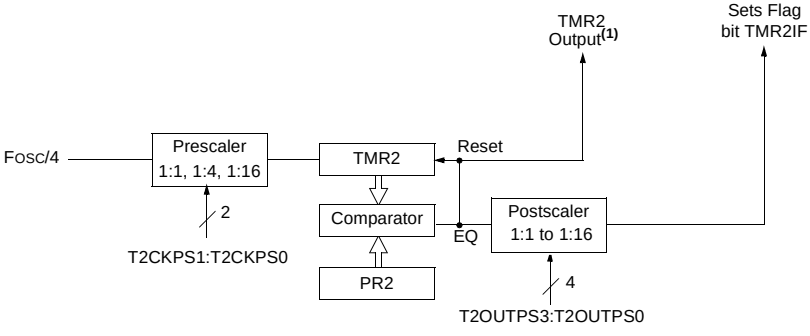
13.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to 0FFh upon Reset.

13.3 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the synchronous serial port module which optionally uses it to generate the shift clock.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



Note 1: TMR2 register output can be software selected by the SSP module as a baud clock.

TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

14.0 TIMER3 MODULE

The Timer3 module timer/counter has the following features:

- 16-bit timer/counter
  (two 8-bit registers; TMR3H and TMR3L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Reset from CCP module trigger

Figure 14-1 is a simplified block diagram of the Timer3 module.

Register 14-1 shows the Timer3 Control register. This register controls the operating mode of the Timer3 module and sets the Enhanced CCP1 and CCP2 clock source.

Register 12-1 shows the Timer1 Control register. This register controls the operating mode of the Timer1 module, as well as containing the Timer1 oscillator enable bit (T1OSCEN) which can be a clock source for Timer3.

REGISTER 14-1: T3CON: TIMER3 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-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

bit 7 **RD16:** 16-bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6, 3 **T3CCP2:T3CCP1:** Timer3 and Timer1 to CCPx Enable bits
1x = Timer3 is the clock source for compare/capture of CCP1 and CCP2 modules
01 = Timer3 is the clock source for compare/capture of CCP2 module,
Timer1 is the clock source for compare/capture of CCP1 module
00 = Timer1 is the clock source for compare/capture of CCP1 and CCP2 modules

bit 5-4 **T3CKPS1:T3CKPS0**: Timer3 Input Clock Prescale Select bits
11 = 1:8 prescale value
10 = 1:4 prescale value
01 = 1:2 prescale value
00 = 1:1 prescale value

bit 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the system clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI
(on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

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

14.1 Timer3 Operation

Timer3 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

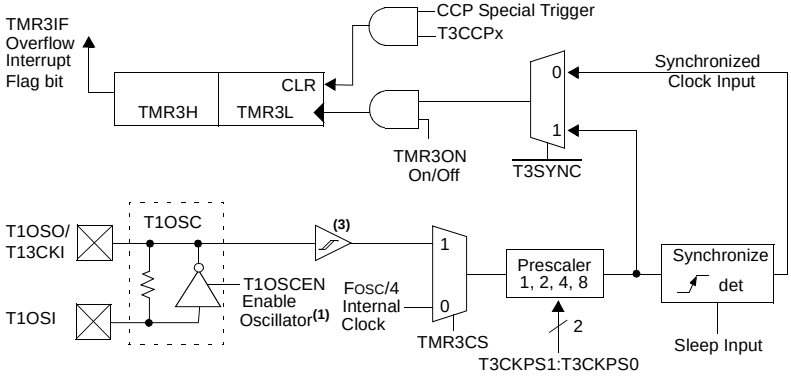
The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>).

When TMR3CS = 0, Timer3 increments every instruction cycle. When TMR3CS = 1, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. That is, the TRISC<1:0> value is ignored and the pins are read as '0'.

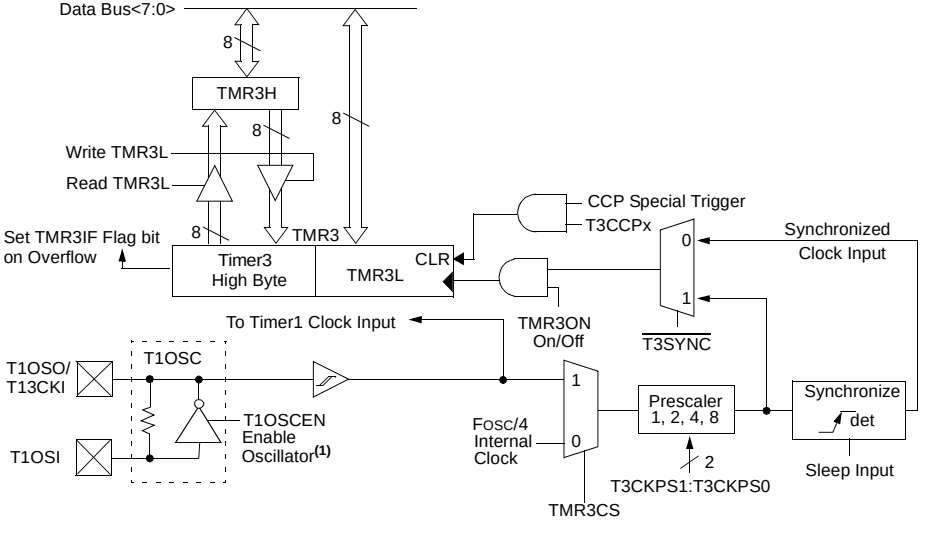
Timer3 also has an internal "Reset input". This Reset can be generated by the CCP module (**Section 14.0 "Timer3 Module"**).

FIGURE 14-1: TIMER3 BLOCK DIAGRAM



Note 1: When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

FIGURE 14-2: TIMER3 BLOCK DIAGRAM CONFIGURED IN 16-BIT READ/WRITE MODE



Note 1: When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

14.2 Timer1 Oscillator

The Timer1 oscillator may be used as the clock source for Timer3. The Timer1 oscillator is enabled by setting the T1OSCEN (T1CON<3>) bit. The oscillator is a low-power oscillator rated up to 200 kHz. See **Section 12.0 "Timer1 Module"** for further details.

14.3 Timer3 Interrupt

The TMR3 register pair (TMR3H:TMR3L) increments from 0000h to 0FFFFh and rolls over to 0000h. The TMR3 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR3IF (PIR2<1>). This interrupt can be enabled/disabled by setting/clearing TMR3 interrupt enable bit, TMR3IE (PIE2<1>).

14.4 Resetting Timer3 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a "special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer3.

Note:	The special event triggers from the CCP module will not set interrupt flag bit, TMR3IF (PIR1<0>).

Timer3 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer3 is running in Asynchronous Counter mode, this Reset operation may not work. In the event that a write to Timer3 coincides with a special event trigger from CCP1, the write will take precedence. In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer3.

TABLE 14-1: REGISTERS ASSOCIATED WITH TIMER3 AS A TIMER/COUNTER

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
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-0 0000	-0-0 0000
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-0 0000	-0-0 0000
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-1 1111	-1-1 1111
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0000 0000	uuuu uuuu

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Timer3 module.

15.0 CAPTURE/COMPARE/PWM (CCP) MODULES

PIC18FXX80/XX85 devices contain a total of two CCP modules: CCP1 and CCP2. CCP1 is an enhanced version of the CCP2 module. CCP1 is fully backward compatible with the CCP2 module.

The CCP1 module differs from CCP2 in the following respect:

- CCP1 contains a special trigger event that may reset Timer1 or the Timer3 register pair
- CCP1 contains “CAN Message Time-Stamp Trigger”
- CCP1 contains enhanced PWM output with programmable dead band and auto-shutdown functionality

Additionally, the CCP2 special event trigger may be used to start an A/D conversion if the A/D module is enabled.

To avoid duplicate information, this section describes basic CCP module operation that applies to both CCP1 and CCP2. Enhanced CCP functionality of the CCP1 module is described in **Section 16.0 “Enhanced Capture/Compare/PWM (ECCP) Module”**.

The control registers for the CCP1 and CCP2 modules are shown in Register 15-1 and Register 15-2, respectively. Table 15-2 details the interactions of the CCP and ECCP modules.

REGISTER 15-1: CCP1CON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

bit 7-6 **P1M1:P1M0:** Enhanced PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx =P1A assigned as capture/compare input; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 11:

00 = Single output; P1A modulated; P1B, P1C, P1D assigned as port pins

01 = Full-bridge output forward; P1D modulated; P1A active; P1B, P1C inactive

10 = Half-bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins

11 = Full-bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B1:DC1B0**: PWM Duty Cycle bit 1 and bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M3:CCP1M0**: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 module)

0001 = Reserved

0010 = Compare mode, toggle output on match

0011 = Reserved

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, initialize CCP pin low, on compare match force CCP pin high

1001 = Compare mode, initialize CCP pin high, on compare match force CCP pin low

1010 = Compare mode, generate software interrupt only, CCP pin is unaffected

1011 = Compare mode, trigger special event, resets TMR1 or TMR3

1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high

1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low

1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high

1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low

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

REGISTER 15-2: CCP2CON REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DC2B1:DC2B0**: PWM Duty Cycle bit 1 and bit 0

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR2L.

bit 3-0 **CCP2M3:CCP2M0**: CCP2 Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP2 module)
0001 = Reserved
0010 = Compare mode, toggle output on match
0011 = Reserved
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, initialize CCP pin low, on compare match force CCP pin high
1001 = Compare mode, initialize CCP pin high, on compare match force CCP pin low
1010 = Compare mode, generate software interrupt only, CCP pin is unaffected
1011 = Compare mode, trigger special event, resets TMR1 or TMR3 and starts A/D conversion if A/D module is enabled
11xx = PWM mode

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

15.1 CCP Module

Both CCP1 and CCP2 are comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte), $1 \leq x \leq 2$. The CCPxCON register controls the operation of CCPx. All are readable and writable.

Table 15-1 shows the timer resources of the CCP module modes.

TABLE 15-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

15.2 Capture Mode

In Capture mode, CCPRxH:CCPRxL captures the 16-bit value of the TMR1 or TMR3 register when an event occurs on pin CCPn. An event is defined as:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

An event is selected by control bits CCPxM3:CCPxM0 (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit, CCPxIF (PIR registers), is set. It must be cleared in software. If another capture occurs before the value in register CCPRx is read, the old captured value will be lost.

15.2.1 CCP PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the appropriate TRIS bit.

Note:	If the CCPx is configured as an output, a write to the port can cause a capture condition.

15.2.2 TIMER1/TIMER3 MODE SELECTION

The timer used with each CCP module is selected in the T3CCP2:T3CCP1 bits of the T3CON register. The timers used with the capture feature (either Timer1 or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work.

TABLE 15-2: INTERACTION OF CCP MODULES

CCP1 Mode	CCP2 Mode	Interaction
Capture	Capture	TMR1 or TMR3 time base. Time base can be different for each CCP.
Capture	Compare	The compare could be configured for the special event trigger which clears either TMR1 or TMR3 depending upon which time base is used.
Compare	Compare	The compare(s) could be configured for the special event trigger which clears TMR1 or TMR3 depending upon which time base is used.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt).
PWM	Capture	None.
PWM	Compare	None.

15.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCPxIE (PIE registers) clear to avoid false interrupts and should clear the flag bit, CCPxIF, following any such change in operating mode.

15.2.4 CCP PRESCALER

There are four prescaler settings specified by bits CCPxM3:CCPxM0. Whenever the CCPx module is turned off, or the CCPx module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. The prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

15.2.5 CAN MESSAGE TIME-STAMP

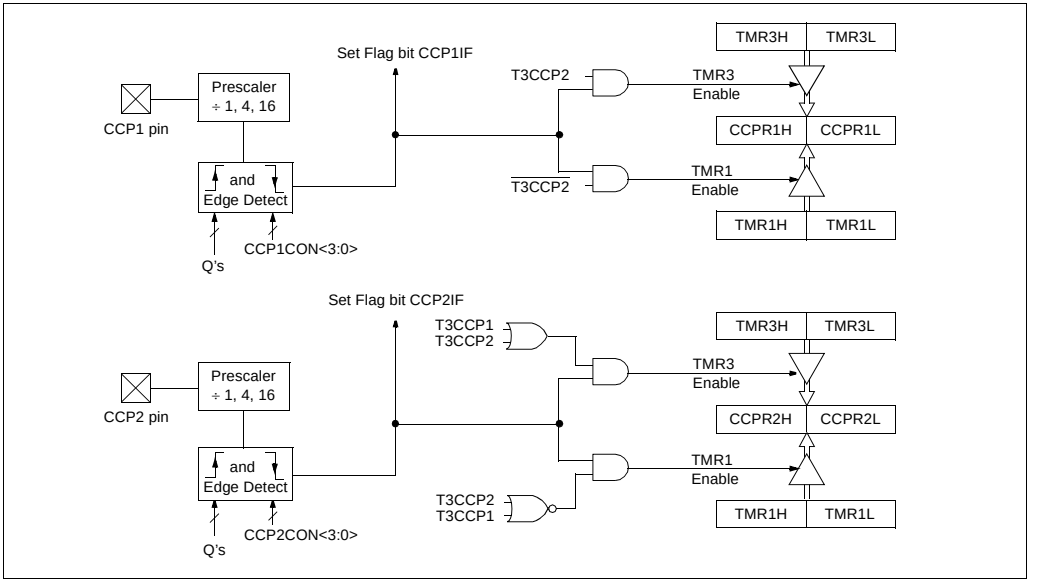
The CAN capture event occurs when a message is received in any of the receive buffers. When configured, the CAN module provides the trigger to the CCP1 module to cause a capture event. This feature is provided to time-stamp the received CAN messages.

This feature is enabled by setting the CANCAP bit of the CAN I/O Control register (CIOCON<4>). The message receive signal from the CAN module then takes the place of the events on RC2/CCP1.

EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON         ; Turn CCP module off
MOVLW   NEW_CAPT_PS     ; Load WREG with the
                        ; new prescaler mode
                        ; value and CCP ON
MOVWF   CCP1CON         ; Load CCP1CON with
                        ; this value
```

FIGURE 15-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



15.3 Compare Mode

In Compare mode, the 16-bit CCPRx register value is constantly compared against either the TMR1 register pair value or the TMR3 register pair value. When a match occurs, the CCPx pin can have one of the following actions:

- Driven high
- Driven low
- Toggle output (high-to-low or low-to-high)
- Remains unchanged

The action on the pin is based on the value of control bits, CCPxM3:CCPxM0. At the same time, interrupt flag bit, CCPxIF, is set.

When configured to drive the CCP pin, the CCP1 pin cannot be changed; CCP1 module controls the pin.

15.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

By default, the CCP2 pin is multiplexed with RC1. Alternately, it can also be multiplexed with either RB3 or RE7. This is done by changing the CCP2MX configuration bit.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the data latch.

15.3.2 TIMER1/TIMER3 MODE SELECTION

The timer used with each CCP module is selected in the T3CCP2:T3CCP1 bits of the T3CON register. Timer1 and/or Timer3 must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

15.3.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCPx pin is not affected. Only a CCP interrupt is generated (if enabled).

15.3.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The special event trigger output of CCP1 resets either the TMR1 or TMR3 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for TMR1 or TMR3.

Additionally, the CCP2 special event trigger will start an A/D conversion if the A/D module is enabled.

Note:	The special event trigger from the CCPx module will not set the Timer1 or Timer3 interrupt flag bits.

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM

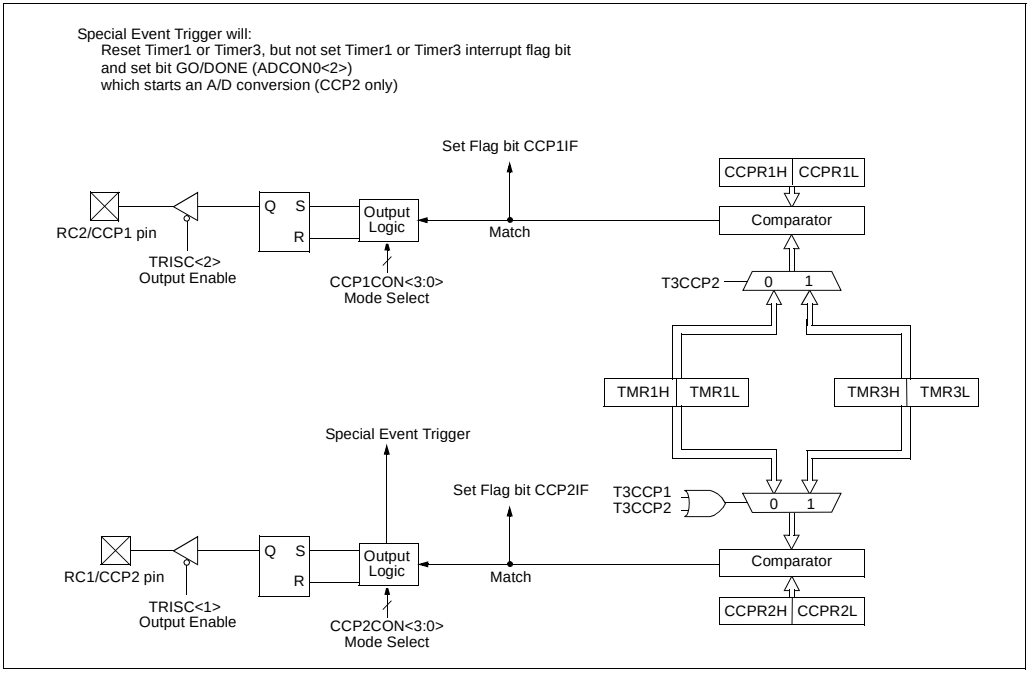


TABLE 15-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3

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
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISD	PORTD Data Direction Register								1111 1111	1111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu
CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-0 0000	-0-0 0000
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-0 0000	-0-0 0000
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-1 1111	-1-1 1111
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0000 0000	uuuu uuuu

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

15.4 PWM Mode

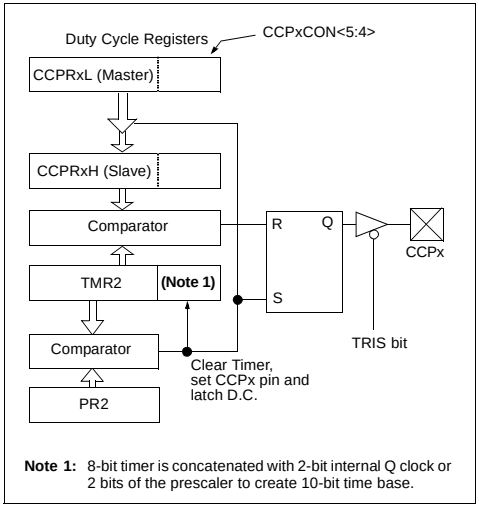
In Pulse Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. For PWM mode to function properly, the TRIS bit for the CCPx pin must be cleared to make it an output.

Note:	Clearing the CCPxCON register will force the CCPx PWM output latch to the default low level. This is not the port data latch.

Figure 15-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.3 "Setup for PWM Operation"**.

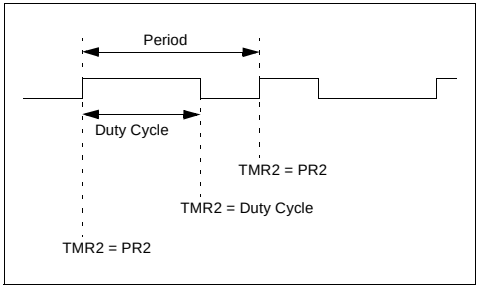
FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

A PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 15-4: PWM OUTPUT



15.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula.

EQUATION 15-1:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

Note:	The Timer2 postscaler (see **Section 13.0 "Timer2 Module"**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

15.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSbs and the CCPxCON<5:4> contain the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. The following equation is used to calculate the PWM duty cycle in time.

EQUATION 15-2:

$$\text{PWM Duty Cycle} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time but the duty cycle value is not latched into CCPRxH until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPRxH is a read-only register.

The CCPRxH register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation.

EQUATION 15-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)}\text{bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

15.4.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing corresponding TRIS bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCPx module for PWM operation.

TABLE 15-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.76 kHz	39.06 kHz	156.3 kHz	312.5 kHz	416.6 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0FFh	0FFh	0FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 15-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

16.0 ENHANCED CAPTURE/ COMPARE/PWM (ECCP) MODULE

The CCP1 module is implemented as a standard CCP module with enhanced PWM capabilities. These capabilities allow for 2 or 4 output channels, user selectable polarity, dead-band control, and automatic shutdown and restart and are discussed in detail in **Section 16.2 "Enhanced PWM Mode"**.

The control register for CCP1 is shown in Register 16-1.

In addition to the expanded functions of the CCP1CON register, the CCP1 module has two additional registers associated with enhanced PWM operation and auto-shutdown features:

- ECCP1DEL
- ECCP1AS

REGISTER 16-1: CCP1CON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

bit 7-6 **P1M1:P1M0:** Enhanced PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A assigned as capture/compare input; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 11:

00 = Single output; P1A modulated, P1B, P1C, P1D assigned as port pins
01 = Full-bridge output forward; P1D modulated; P1A active; P1B, P1C inactive
10 = Half-bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
11 = Full-bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B1:DC1B0**: PWM Duty Cycle bit 1 and bit 0

Capture mode:
Unused.
Compare mode:
Unused.
PWM mode:
These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M3:CCP1M0**: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 module)
0001 = Reserved
0010 = Compare mode, toggle output on match
0011 = Capture mode, CAN message time-stamp
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, initialize CCP pin low, on compare match, force CCP pin high
1001 = Compare mode, initialize CCP pin high, on compare match, force CCP pin low
1010 = Compare mode, generate software interrupt only, CCP pin is unaffected
1011 = Compare mode, trigger special event, resets TMR1 or TMR3
1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high
1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low
1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high
1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low

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

16.1 ECCP Outputs

The enhanced CCP module may have up to four outputs depending on the selected operating mode. These outputs, designated P1A through P1D, are multiplexed with I/O pins RC2, RE6, RE5 and RG4. The pin assignments are summarized in Table 16-1.

To configure I/O pins as PWM outputs, the proper PWM mode must be selected by setting the P1Mx and CCP1Mx bits (CCP1CON<7:6> and <3:0>, respectively). The appropriate TRIS direction bits for the port pins must also be set as outputs.

TABLE 16-1: PIN ASSIGNMENTS FOR VARIOUS ECCP MODES

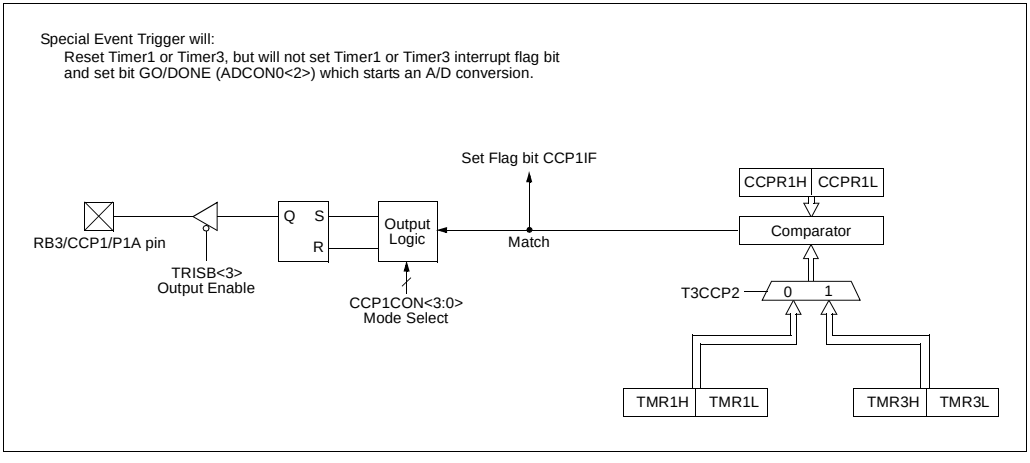
ECCP Mode	CCP1CON Configuration	RC2	RE6	RE5	RG4
Compatible CCP	00xx11xx	CCP1	RE6	RE5	RG4
Dual PWM	10xx11xx	P1A	P1B(2)	RE5	RG4
Quad PWM	x1xx11xx	P1A	P1B(2)	P1C(2)	P1D

Legend: x = Don't care. Shaded cells indicate pin assignments not used by ECCP in a given mode.

Note 1: TRIS register values must be configured appropriately.

2: On PIC18F8X8X devices, these pins can be alternately multiplexed with RH7 or RH6 by changing the ECCPMX configuration bit.

FIGURE 16-1: COMPARE MODE OPERATION BLOCK DIAGRAM



16.2 Enhanced PWM Mode

The Enhanced PWM mode provides additional PWM output options for a broader range of control applications. The module is a backward compatible version of the standard CCP module and offers up to four outputs, designated P1A through P1D. Users are also able to select the polarity of the signal (either active-high or active-low). The module's output mode and polarity are configured by setting the P1M1:P1M0 and CCP1M3:CCP1M0 bits of the CCP1CON register (CCP1CON<7:6> and CCP1CON<3:0>, respectively).

Figure 16-2 shows a simplified block diagram of PWM operation. All control registers are double-buffered and are loaded at the beginning of a new PWM cycle (the period boundary when Timer2 resets) in order to prevent glitches on any of the outputs. The exception is the PWM Delay register, ECCP1DEL, which is loaded at either the duty cycle boundary or the boundary period (whichever comes first). Because of the buffering, the module waits until the assigned timer resets instead of starting immediately. This means that enhanced PWM waveforms do not exactly match the standard PWM waveforms, but are instead offset by one full instruction cycle (4 TOSC).

As before, the user must manually configure the appropriate TRIS bits for output.

16.2.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following equation.

EQUATION 16-1:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period]. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is copied from CCPR1L into CCPR1H

Note:	The Timer2 postscaler (see **Section 13.0 "Timer2 Module"**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

16.2.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The PWM duty cycle is calculated by the following equation.

EQUATION 16-2:

$$\text{PWM Duty Cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not copied into CCPR1H until a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the CCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation:

EQUATION 16-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

16.2.3 PWM OUTPUT CONFIGURATIONS

The P1M1:P1M0 bits in the CCP1CON register allow one of four configurations:

- Single Output
- Half-Bridge Output
- Full-Bridge Output, Forward mode
- Full-Bridge Output, Reverse mode

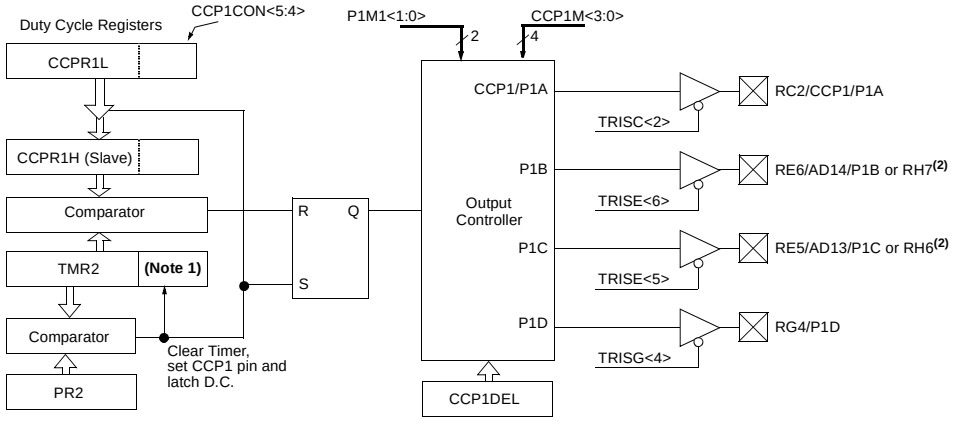
The Single Output mode is the standard PWM mode discussed in **Section 16.2 "Enhanced PWM Mode"**. The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 16-3.

TABLE 16-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

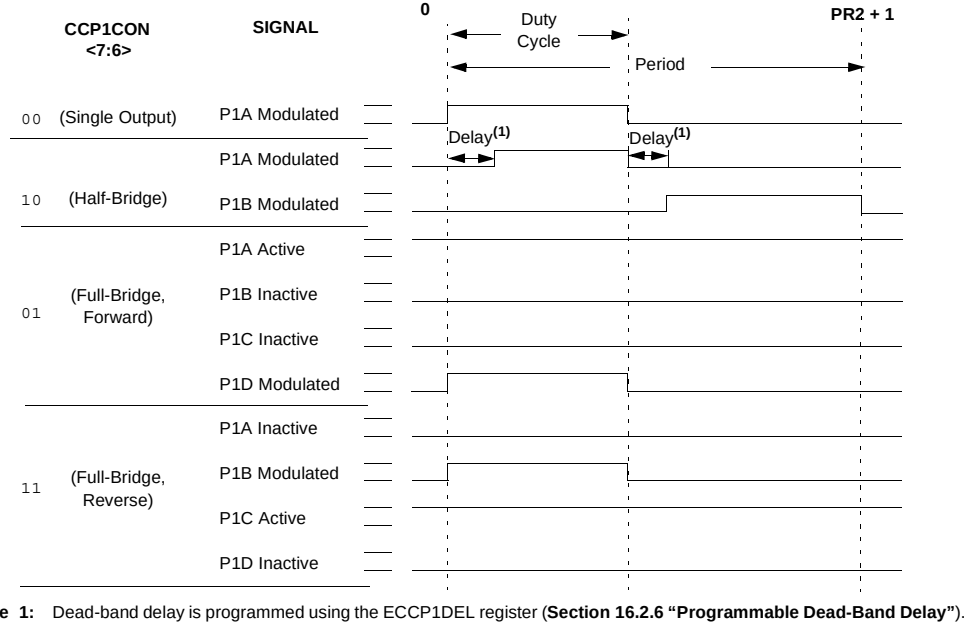
PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

FIGURE 16-2: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODULE



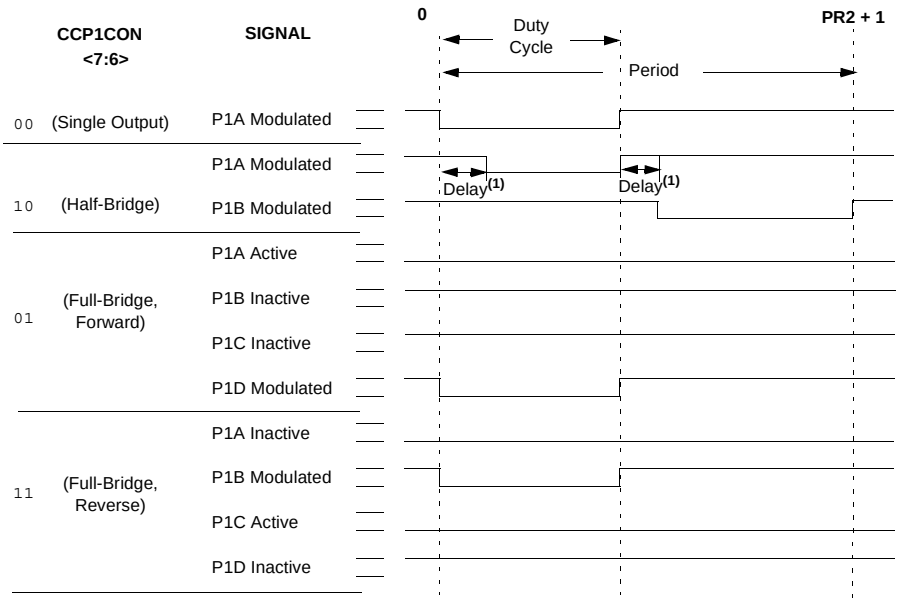
Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal Q clock or 2 bits of the prescaler to create the 10-bit time base.
2: Alternate setting controlled by the ECCPMX bit (PIC18F8X8X devices only).

FIGURE 16-3: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)



Note 1: Dead-band delay is programmed using the ECCP1DEL register (**Section 16.2.6 "Programmable Dead-Band Delay"**).

FIGURE 16-4: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)



Note 1: Dead-band delay is programmed using the ECCP1DEL register (**Section 16.2.6 "Programmable Dead-Band Delay"**).

Relationships:

- Period = 4 * TOSC * (PR2 + 1) * (TMR2 prescale value)
- Duty Cycle = TOSC * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 prescale value)
- Delay = 4 * TOSC * (PWM1CON<6:0>)

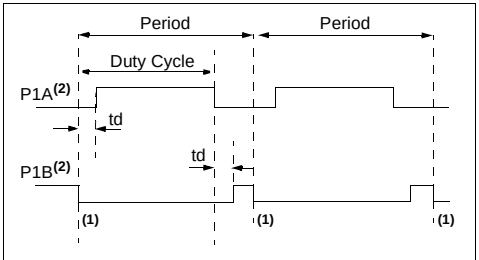
16.2.4 HALF-BRIDGE MODE

In the Half-Bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the P1A pin while the complementary PWM output signal is output on the P1B pin (Figure 16-5). This mode can be used for half-bridge applications, as shown in Figure 16-6, or for full-bridge applications where four power switches are being modulated with two PWM signals.

In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits PDC6:PDC0 sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 16.2.6 "Programmable Dead-Band Delay"** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORTC<2> and PORTE<6> data latches, the TRISC<2> and TRISE<6> bits must be cleared to configure P1A and P1B as outputs.

FIGURE 16-5: HALF-BRIDGE PWM OUTPUT

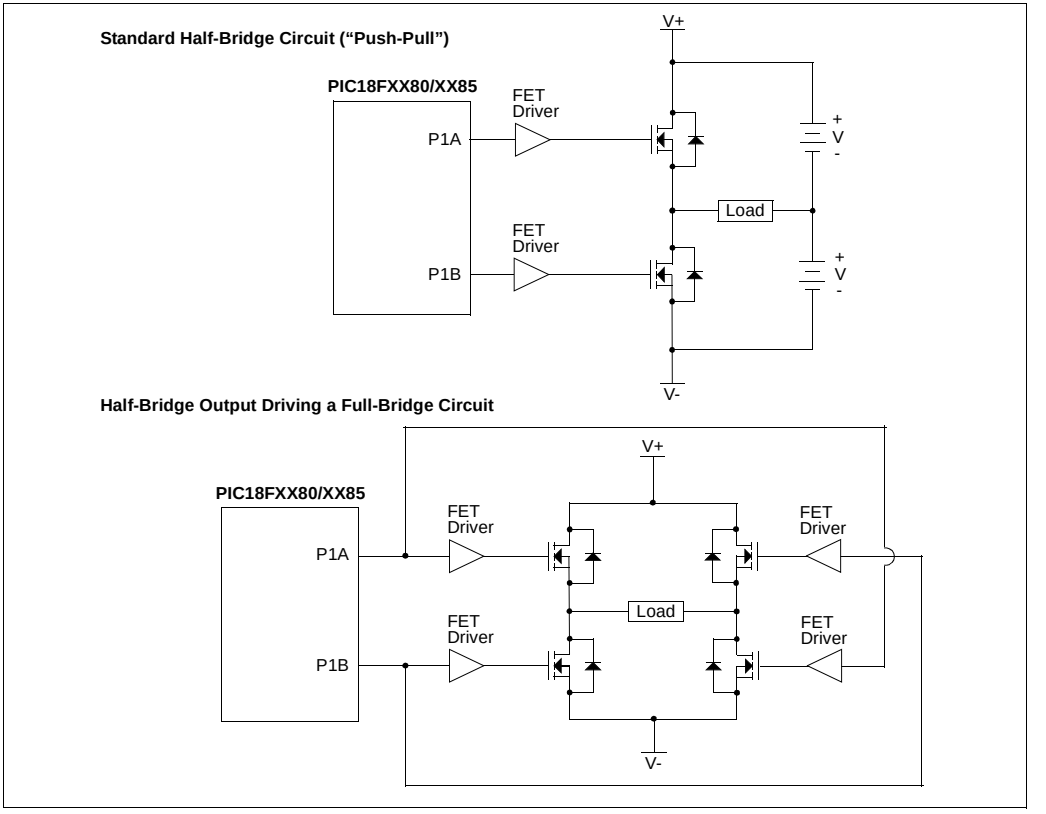


td = Dead-band Delay

Note 1: At this time, the TMR2 register is equal to the PR2 register.

2: Output signals are shown as active-high.

FIGURE 16-6: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS

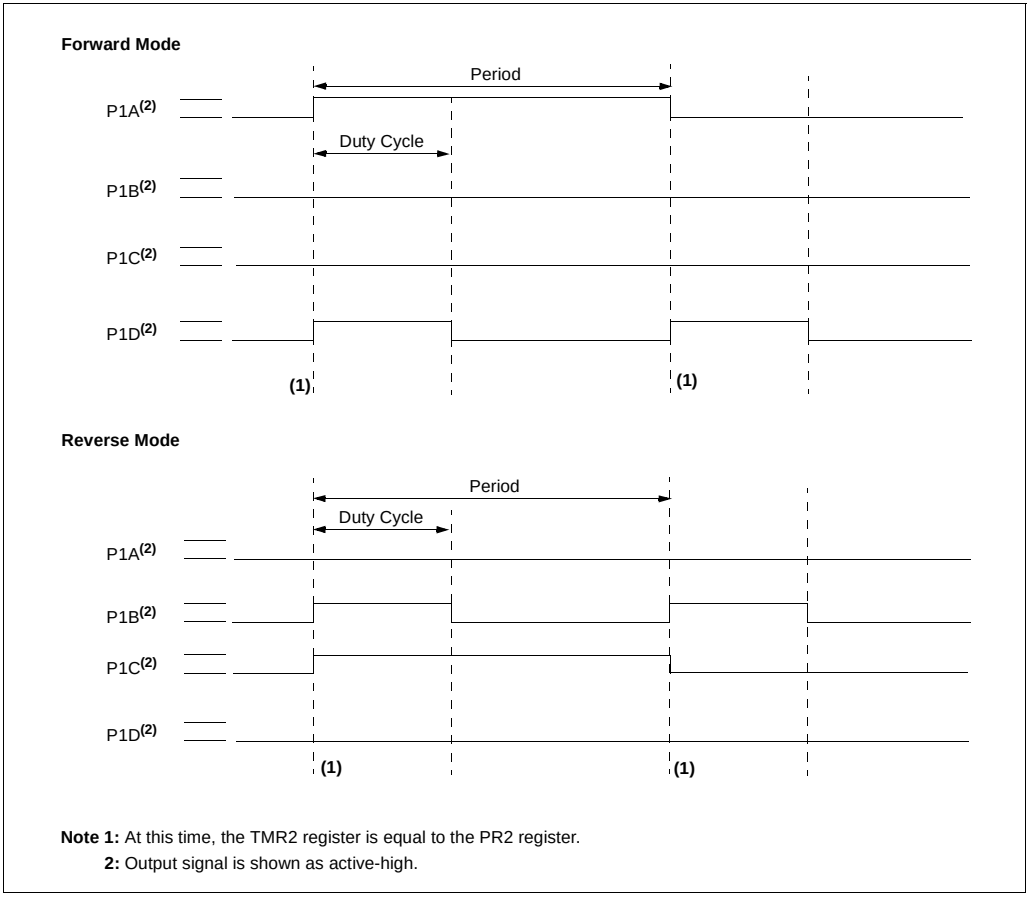


16.2.5 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin P1A is continuously active and pin P1D is modulated. In the Reverse mode, pin PGC is continuously active and pin P1B is modulated. These are illustrated in Figure 16-7.

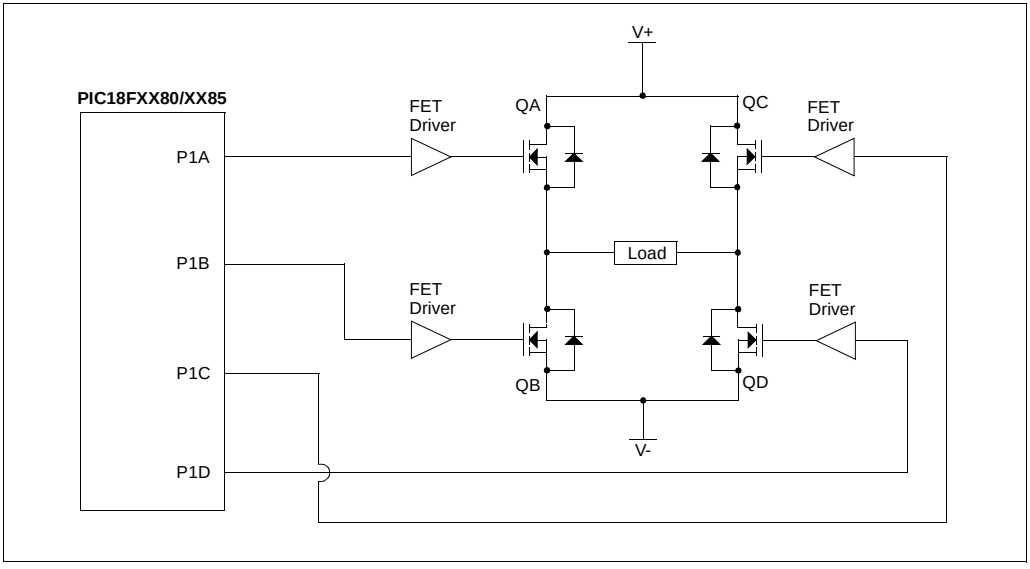
P1A, P1B, P1C and P1D outputs are multiplexed with the PORTC<2>, PORTE<6:5> and PORTG<4> data latches. The TRISC<2>, TRISC<6:5> and TRISG<4> bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 16-7: FULL-BRIDGE PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signal is shown as active-high.

FIGURE 16-8: EXAMPLE OF FULL-BRIDGE APPLICATION



16.2.5.1 Direction Change in Full-Bridge Mode

In the Full-Bridge Output mode, the P1M1 bit in the CCP1CON register allows the user to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will assume the new direction on the next PWM cycle.

Just before the end of the current PWM period, the modulated outputs (P1B and P1D) are placed in their inactive state while the unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction. This occurs in a time interval of (4 TOSC * (Timer2 Prescale value)) before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPS bit (T2CON<1:0>). During the interval from the switch of the unmodulated outputs to the beginning of the next period, the modulated outputs (P1B and P1D) remain inactive. This relationship is shown in Figure 16-9.

Note that in the Full-Bridge Output mode, the CCP1 module does not provide any dead-band delay. In general, since only one output is modulated at all times, dead-band delay is not required. However, there is a situation where a dead-band delay might be required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn off time of the power switch, including the power device and driver circuit, is greater than the turn on time.

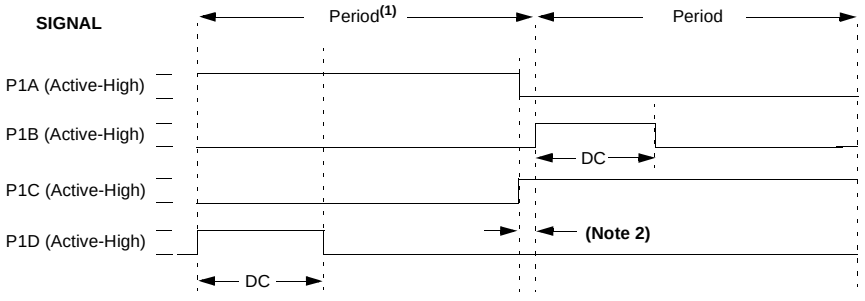
Figure 16-10 shows an example where the PWM direction changes from forward to reverse at a near 100% duty cycle. At time t1, the output P1A and P1D become inactive while output P1C becomes active. In this example, since the turn off time of the power devices is longer than the turn on time, a shoot-through current may flow through power devices QC and QD (see Figure 16-8) for the duration of ‘t’. The same phenomenon will occur to power devices QA and QB for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, one of the following requirements must be met:

1. Reduce PWM for a PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

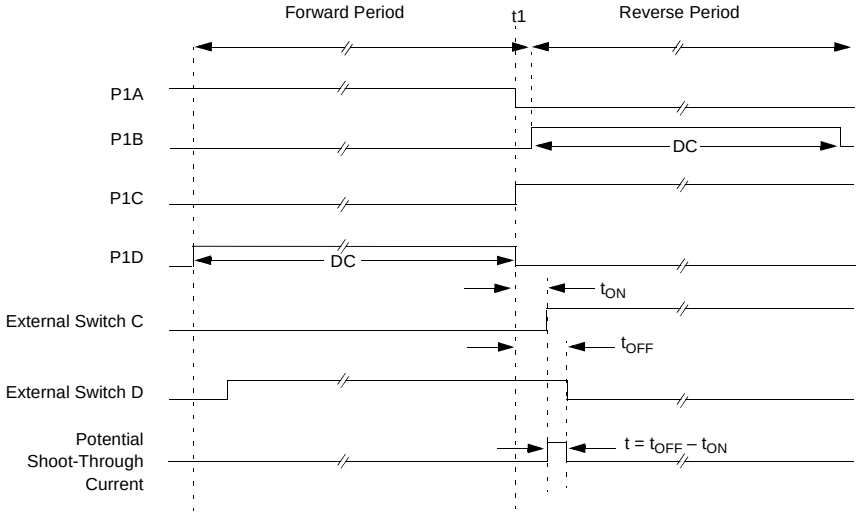
FIGURE 16-9: PWM DIRECTION CHANGE



Note 1: The direction bit in the CCP1 Control register (CCP1CON<7>) is written any time during the PWM cycle.

2: When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle at intervals of 4 TOSC, 16 TOSC or 64 TOSC, depending on the Timer2 prescaler value. The modulated P1B and P1D signals are inactive at this time.

FIGURE 16-10: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



Note 1: All signals are shown as active-high.

2: t_{ON} is the turn on delay of power switch QC and its driver.

3: t_{OFF} is the turn off delay of power switch QD and its driver.

16.2.6 PROGRAMMABLE DEAD-BAND DELAY

In half-bridge applications where all power switches are modulated at the PWM frequency at all times, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (shoot-through current) may flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In the Half-Bridge Output mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 16-5 for an illustration. The lower seven bits of the ECCP1DEL register (Register 16-2) set the delay period in terms of microcontroller instruction cycles (TCY or 4 TOSC).

16.2.7 ENHANCED PWM AUTO-SHUTDOWN

When the CCP1 is programmed for any of the enhanced PWM modes, the active output pins may be configured for auto-shutdown. Auto-shutdown immediately places the enhanced PWM output pins into a defined shutdown state when a shutdown event occurs.

A shutdown event can be caused by either of the two comparator modules or a low level on the RB0 pin (or any combination of these three sources). The comparators may be used to monitor a voltage input proportional to a current being monitored in the bridge circuit. If the voltage exceeds a threshold, the comparator switches state and triggers a shutdown. Alternatively, a low digital signal on the RB0 pin can also trigger a shutdown. The auto-shutdown feature can be disabled by not selecting any auto-shutdown sources. The auto-shutdown sources to be used are selected using the ECCPAS2:ECCPAS0 bits (bits <6:4> of the ECCP1AS register).

When a shutdown occurs, the output pins are asynchronously placed in their shutdown states, specified by the PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits (ECCP1AS<3:0>). Each pin pair (P1A/P1C and P1B/P1D) may be set to drive high, drive low, or be tri-stated (not driving). The ECCPASE bit (ECCP1AS<7>) is also set to hold the enhanced PWM outputs in their shutdown states.

The ECCPASE bit is set by hardware when a shutdown event occurs. If automatic restarts are not enabled, the ECCPASE bit is cleared by firmware when the cause of the shutdown clears. If automatic restarts are enabled, the ECCPASE bit is automatically cleared when the cause of the auto-shutdown has cleared.

If the ECCPASE bit is set when a PWM period begins, the PWM outputs remain in their shutdown state for that entire PWM period. When the ECCPASE bit is cleared, the PWM outputs will return to normal operation at the beginning of the next PWM period.

Note:	Writing to the ECCPASE bit is disabled while a shutdown condition is active.

REGISTER 16-2: ECCP1DEL: ECCP1 DELAY REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
bit 7							bit 0

bit 7 **PRSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCPASE must be cleared in software to restart the PWM

bit 6-0 **PDC<6:0>:** PWM Delay Count bits

Number of FOSC/4 (4 * TOSC) cycles between the scheduled time when a PWM signal should transition active and the actual time it transitions active.

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

REGISTER 16-3: ECCP1AS: ENHANCED CAPTURE/COMPARE/PWM AUTO-SHUTDOWN 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-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
bit 7							bit 0

bit 7 **ECCPASE:** ECCP Auto-Shutdown Event Status bit
0 = ECCP outputs are operating
1 = A shutdown event has occurred; ECCP outputs are in shutdown state

bit 6-4 **ECCPAS<2:0>:** ECCP Auto-Shutdown Source Select bits
000 = Auto-shutdown is disabled
001 = Comparator 1 output
010 = Comparator 2 output
011 = Either Comparator 1 or 2
100 = RB0
101 = RB0 or Comparator 1
110 = RB0 or Comparator 2
111 = RB0 or Comparator 1 or Comparator 2

bit 3-2 **PSSACn:** Pins A and C Shutdown State Control bits
00 = Drive pins A and C to '0'
01 = Drive pins A and C to '1'
1x = Pins A and C tri-state

bit 1-0 **PSSBDn:** Pins B and D Shutdown State Control bits
00 = Drive pins B and D to '0'
01 = Drive pins B and D to '1'
1x = Pins B and D tri-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

16.2.7.1 Auto-Shutdown and Automatic Restart

The auto-shutdown feature can be configured to allow automatic restarts of the module following a shutdown event. This is enabled by setting the PRSEN bit of the ECCP1DEL register (ECCP1DEL<7>).

In Shutdown mode with PRSEN = 1 (Figure 16-11), the ECCPASE bit will remain set for as long as the cause of the shutdown continues. When the shutdown condition clears, the ECCPASE bit is cleared. If PRSEN = 0 (Figure 16-12), once a shutdown condition occurs, the ECCPASE bit will remain set until it is cleared by firmware. Once ECCPASE is cleared, the enhanced PWM will resume at the beginning of the next PWM period.

Note:	Writing to the ECCPASE bit is disabled while a shutdown condition is active.

Independent of the PRSEN bit setting, if the auto-shutdown source is one of the comparators, the shutdown condition is a level. The ECCPASE bit cannot be cleared as long as the cause of the shutdown persists.

The Auto-Shutdown mode can be forced by writing a '1' to the ECCPASE bit.

16.2.8 START-UP CONSIDERATIONS

When the ECCP module is used in the PWM mode, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins. When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The CCP1M1:CCP1M0 bits (CCP1CON<1:0>) allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pins are configured as outputs. Changing the polarity configuration while the PWM pins are configured as outputs is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pins for output at the same time as the ECCP module may cause damage to the application circuit. The ECCP module must be enabled in the proper Output mode and complete a full PWM cycle before configuring the PWM pins as outputs. The completion of a full PWM cycle is indicated by the TMR2IF bit being set as the second PWM period begins.

FIGURE 16-11: PWM AUTO-SHUTDOWN (PRSEN = 1, AUTO-RESTART ENABLED)

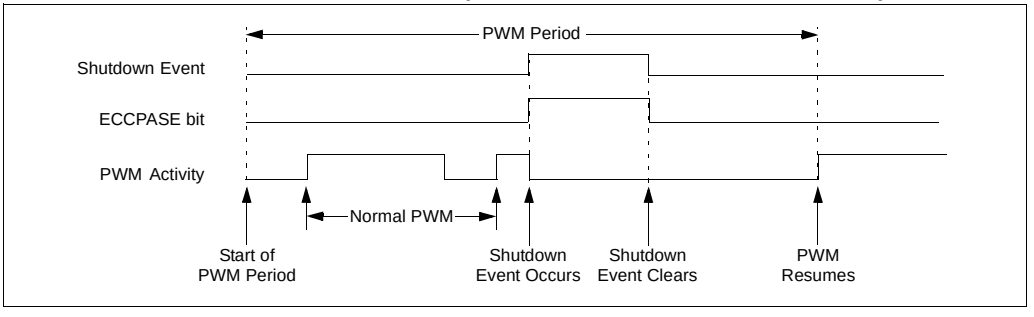
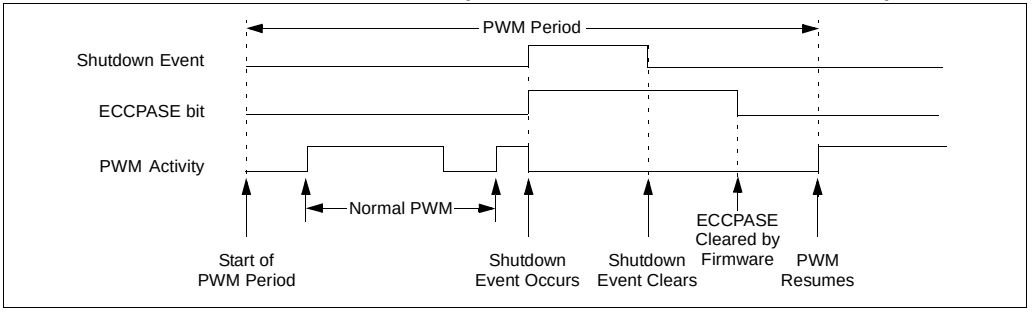


FIGURE 16-12: PWM AUTO-SHUTDOWN (PRSEN = 0, AUTO-RESTART DISABLED)



16.2.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the ECCP1 module for PWM operation:

1. Configure the PWM pins, P1A and P1B (and P1C and P1D, if used), as inputs by setting the corresponding TRISB bits.
2. Set the PWM period by loading the PR2 register.
3. Configure the ECCP1 module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M1:P1M0 bits.
   - Select the polarities of the PWM output signals with the CCP1M3:CCP1M0 bits.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
5. For Half-Bridge Output mode, set the dead-band delay by loading ECCP1DEL<6:0> with the appropriate value.
6. If auto-shutdown operation is required, load the ECCPAS register:
   - Select the auto-shutdown sources using the ECCPAS<2:0> bits.
   - Select the shutdown states of the PWM output pins using PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits.
   - Set the ECCPASE bit (ECCPAS<7>).
   - Configure the comparators using the CMCON register.
   - Configure the comparator inputs as analog inputs.
7. If auto-restart operation is required, set the PRSEN bit (ECCP1DEL<7>).
8. Configure and start TMR2:
   - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
   - Set the TMR2 prescale value by loading the T2CKPS bits (T2CON<1:0>).
   - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
9. Enable PWM outputs after a new PWM cycle has started:
   - Wait until TMR2 overflows (TMR2IF bit is set).
   - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRISB bits.
   - Clear the ECCPASE bit (ECCP1AS<7>).

16.2.10 EFFECTS OF A RESET

Both Power-on and subsequent Resets will force all ports to Input mode and the CCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.

TABLE 16-3: REGISTERS ASSOCIATED WITH PWM AND TIMER2

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
TRISE	PORTE Data Direction Register								1111 1111	1111 1111
TRISG	—	—	—	PORTG Data Direction Register					---1 1111	---1 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
ECCP1DEL	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	uuuu uuuu

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

NOTES:

17.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

17.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

17.2 Control Registers

The MSSP module has three associated registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

17.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

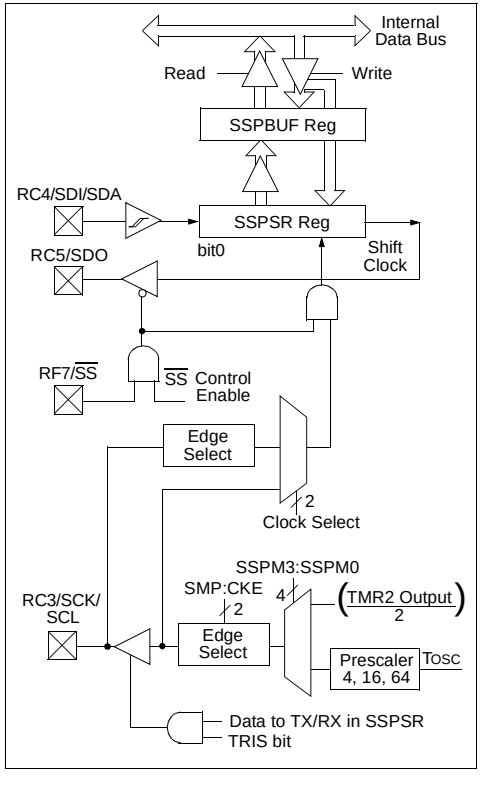
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) – RF7/$\overline{SS}$

Figure 17-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 17-1: MSSP BLOCK DIAGRAM (SPI MODE)



17.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON1 and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 17-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Edge Select bit
When CKP = 0:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK
When CKP = 1:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
Used in I^2C mode only.

bit 4 **P:** Stop bit
Used in I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.

bit 3 **S:** Start bit
Used in I^2C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ bit Information
Used in I^2C mode only.

bit 1 **UA:** Update Address bit
Used in I^2C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

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

REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit
SPI Slave mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode.The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
0 = No overflow

Note: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
1 = Enables serial port and configures SCK, SDO, SDI, and $\overline{SS}$ as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0011 = SPI Master mode, clock = TMR2 output/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

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

17.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received(transmit complete)?
        BRA     LOOP            ;No
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

17.3.3 ENABLING SPI I/O

To enable the serial port, SSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$ must have TRISF<7> bit set

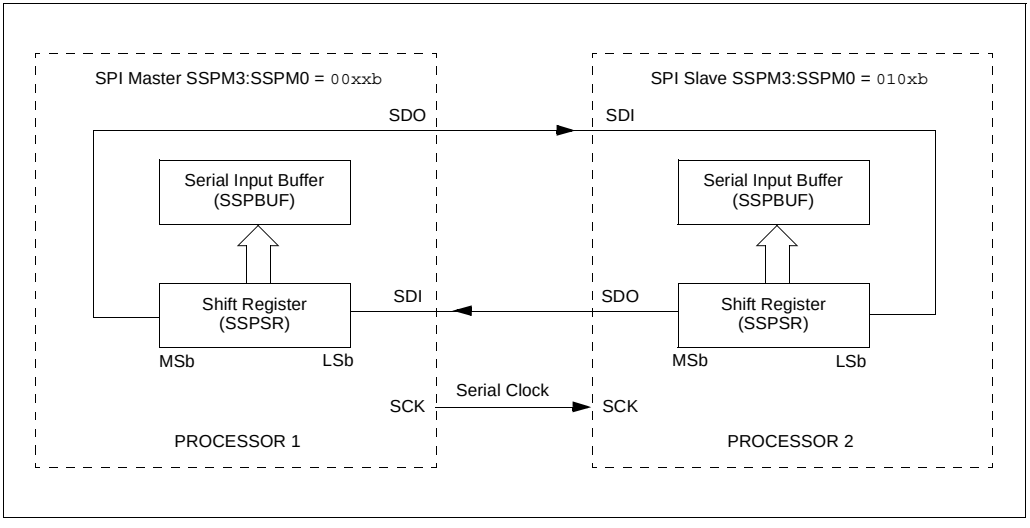
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

17.3.4 TYPICAL CONNECTION

Figure 17-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data - Slave sends dummy data
- Master sends data - Slave sends data
- Master sends dummy data - Slave sends data

FIGURE 17-2: SPI MASTER/SLAVE CONNECTION



17.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 17-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a “Line Activity Monitor” mode.

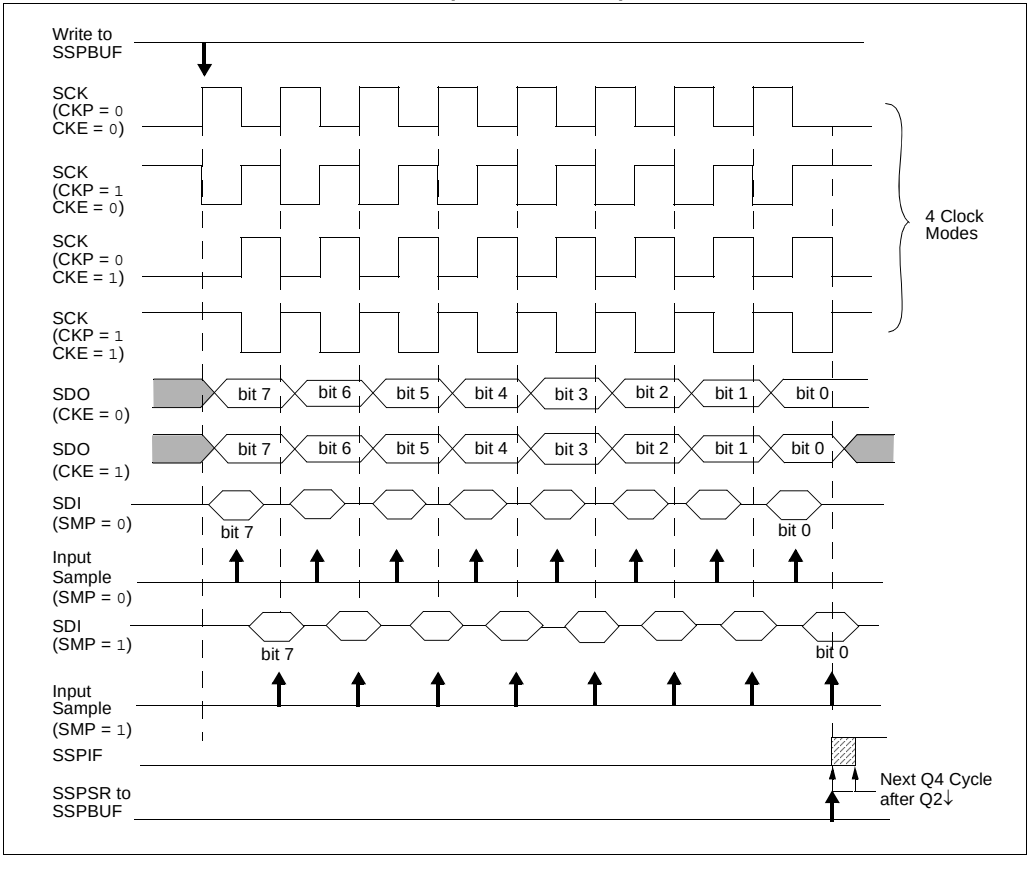
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication, as shown in Figure 17-3, Figure 17-5 and Figure 17-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 17-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 17-3: SPI MODE WAVEFORM (MASTER MODE)



17.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

17.3.7 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.
2:	If the SPI is used in Slave mode with CKE set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 17-4: SLAVE SYNCHRONIZATION WAVEFORM

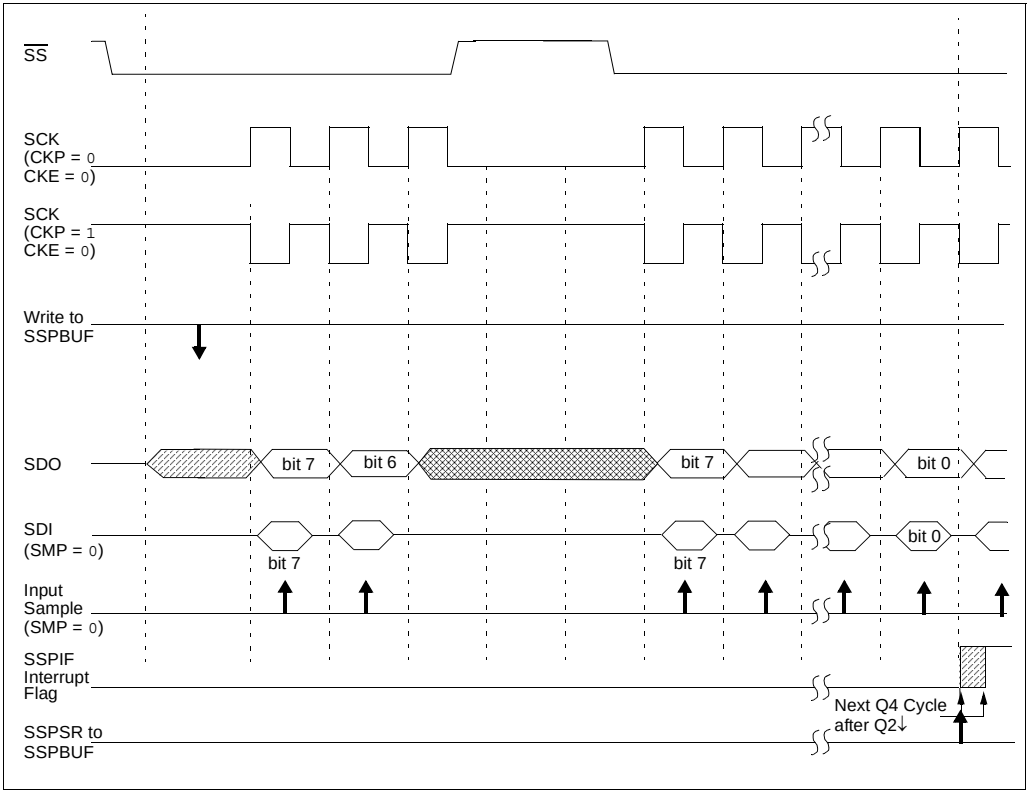


FIGURE 17-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

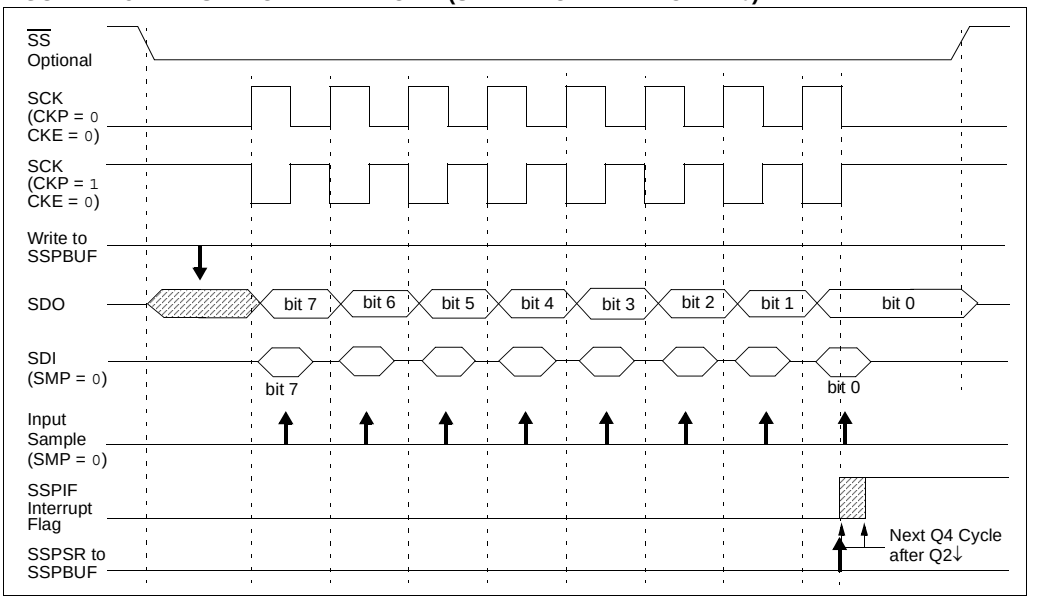
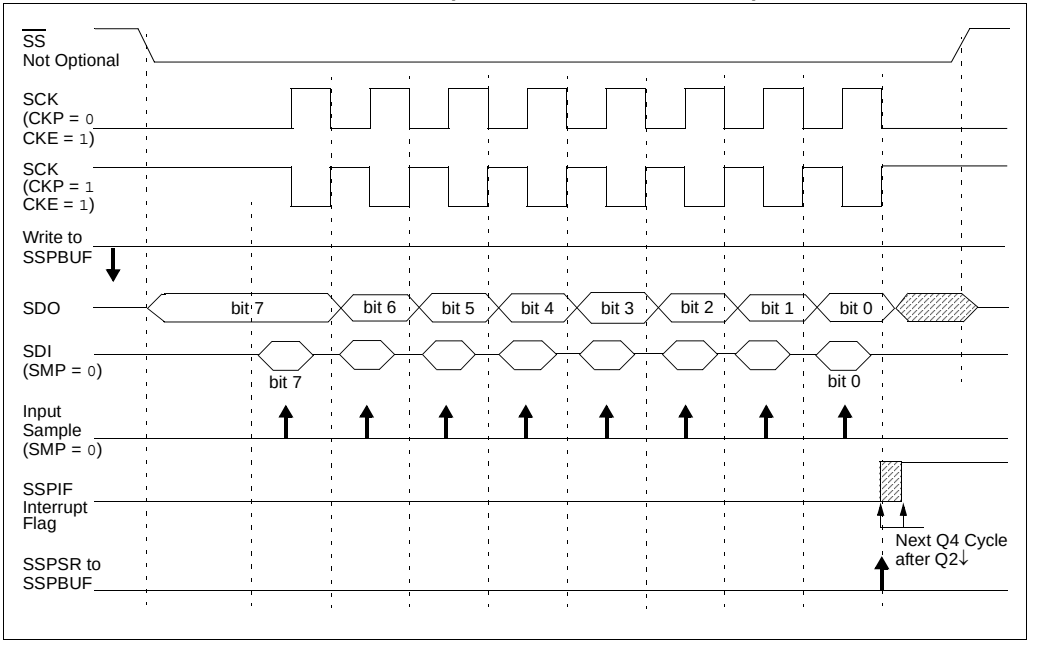


FIGURE 17-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



17.3.8 SLEEP OPERATION

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

17.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

17.3.10 BUS MODE COMPATIBILITY

Table 17-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 17-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also a SMP bit which controls when the data is sampled.

TABLE 17-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	1111 1111	uuuu uuuu
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

17.4 I²C Mode

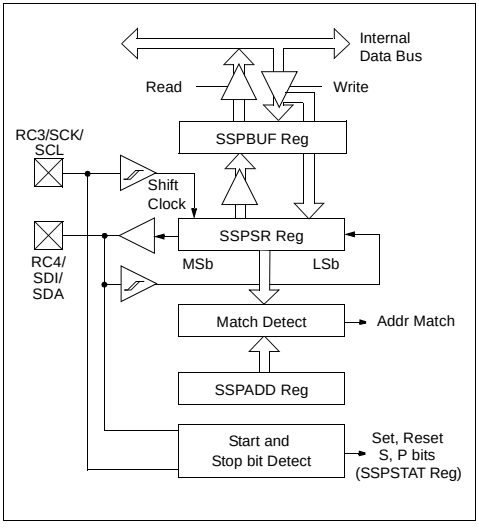
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

FIGURE 17-7: MSSP BLOCK DIAGRAM (I²C MODE)



17.4.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I²C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in I²C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 17-3: SSPSTAT: MSSP STATUS REGISTER (I^2C MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** Slew Rate Control bit

In Master or Slave mode:

1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)

0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

In Master or Slave mode:

1 = Enable SMBus specific inputs

0 = Disable SMBus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit

In Master mode:

Reserved.

In Slave mode:

1 = Indicates that the last byte received or transmitted was data

0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit

1 = Indicates that a Stop bit has been detected last

0 = Stop bit was not detected last

Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 3 **S:** Start bit

1 = Indicates that a Start bit has been detected last

0 = Start bit was not detected last

Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ bit Information (I^2C mode only)

In Slave mode:

1 = Read

0 = Write

Note: This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{ACK}$ bit.

In Master mode:

1 = Transmit is in progress

0 = Transmit is not in progress

Note: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

In Receive mode:

1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is full

0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is empty

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

REGISTER 17-4: SSPCON1: MSSP CONTROL REGISTER 1 (I^2C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPBUF register was attempted while the I^2C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a “don’t care” bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

In Transmit mode:

This is a “don’t care” bit in Transmit mode.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, the SDA and SCL pins must be properly configured as input or output.

bit 4 **CKP:** SCK Release Control bit

In Slave mode:

1 = Release clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1011 = I^2C Firmware Controlled Master mode (slave Idle)

1000 = I^2C Master mode, clock = FOSC/(4 * (SSPADD + 1))

0111 = I^2C Slave mode, 10-bit address

0110 = I^2C Slave mode, 7-bit address

Note: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

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