

Welcome to [E-XFL.COM](http://E-XFL.COM)

### What is "[Embedded - Microcontrollers](#)"?

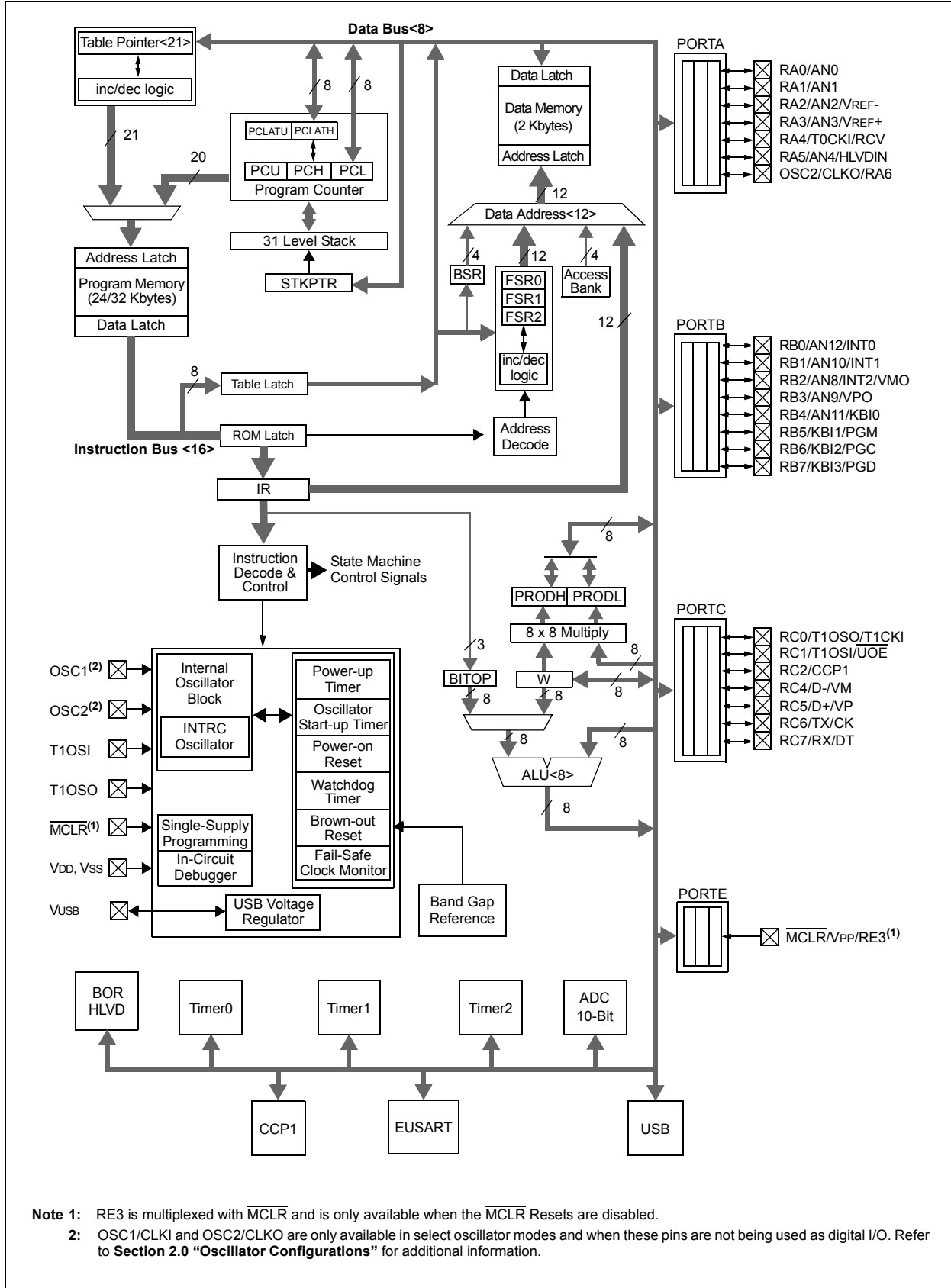
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

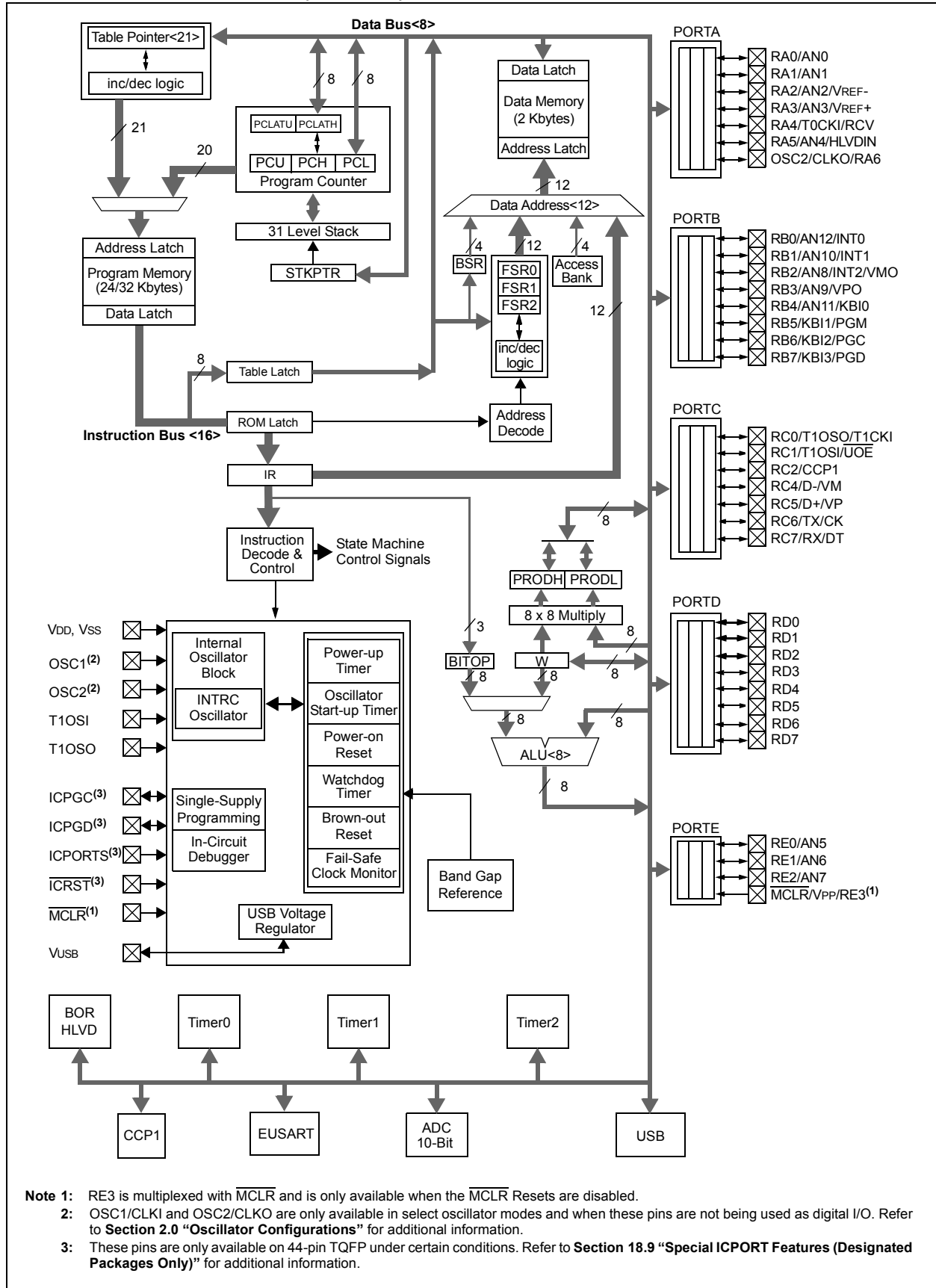
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f4450-i-ml">https://www.e-xfl.com/product-detail/microchip-technology/pic18f4450-i-ml</a>

# PIC18F2450/4450

FIGURE 1-1: PIC18F2450 (28-PIN) BLOCK DIAGRAM



**FIGURE 1-2: PIC18F4450 (40/44-PIN) BLOCK DIAGRAM**



**TABLE 1-2: PIC18F2450 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	SPDIP, SOIC	QFN			
RA0/AN0 RA0 AN0	2	27	I/O I	TTL Analog	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	28	I/O I	TTL Analog	Digital I/O. Analog input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	4	1	I/O I I	TTL Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	2	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (high) input.
RA4/T0CKI/RCV RA4 T0CKI RCV	6	3	I/O I I	ST ST TTL	Digital I/O. Timer0 external clock input. External USB transceiver RCV input.
RA5/AN4/HLVDIN RA5 AN4 HLVDIN	7	4	I/O I I	TTL Analog Analog	Digital I/O. Analog input 4. High/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      I = Input  
 O = Output      P = Power

# PIC18F2450/4450

**TABLE 1-3: PIC18F4450 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
RB0/AN12/INT0 RB0 AN12 INT0	33	9	8	I/O I I	TTL Analog ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. Analog input 12. External interrupt 0.
RB1/AN10/INT1 RB1 AN10 INT1	34	10	9	I/O I I	TTL Analog ST	Digital I/O. Analog input 10. External interrupt 1.
RB2/AN8/INT2/VMO RB2 AN8 INT2 VMO	35	11	10	I/O I I O	TTL Analog ST —	Digital I/O. Analog input 8. External interrupt 2. External USB transceiver VMO output.
RB3/AN9/VPO RB3 AN9 VPO	36	12	11	I/O I O	TTL Analog —	Digital I/O. Analog input 9. External USB transceiver VPO output.
RB4/AN11/KBI0 RB4 AN11 KBI0	37	14	14	I/O I I	TTL Analog TTL	Digital I/O. Analog input 11. Interrupt-on-change pin.
RB5/KBI1/PGM RB5 KBI1 PGM	38	15	15	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC RB6 KBI2 PGC	39	16	16	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD RB7 KBI3 PGD	40	17	17	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin.

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
 ST = Schmitt Trigger input with CMOS levels      I = Input  
 O = Output      P = Power

**Note 1:** These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

# PIC18F2450/4450

## REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 <sup>(1)</sup>	U-0	R/W-1	R-1	R-1	R/W-0 <sup>(2)</sup>	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

### Legend:

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

- bit 7            **IPEN:** Interrupt Priority Enable bit  
                   1 = Enable priority levels on interrupts  
                   0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6            **SBOREN:** BOR Software Enable bit<sup>(1)</sup>  
                   If BOREN1:BOREN0 = 01:  
                   1 = BOR is enabled  
                   0 = BOR is disabled  
                   If BOREN1:BOREN0 = 00, 10 or 11:  
                   Bit is disabled and read as '0'.
- bit 5            **Unimplemented:** Read as '0'
- bit 4             **$\overline{RI}$ :** RESET Instruction Flag bit  
                   1 = The RESET instruction was not executed (set by firmware only)  
                   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 = Set by power-up, CLRWD<sub>T</sub> instruction or SLEEP instruction  
                   0 = A WDT time-out occurred
- bit 2             **$\overline{PD}$ :** Power-Down Detection Flag bit  
                   1 = Set by power-up or by the CLRWD<sub>T</sub> instruction  
                   0 = Set by execution of the SLEEP instruction
- bit 1             **$\overline{POR}$ :** Power-on Reset Status bit<sup>(2)</sup>  
                   1 = A Power-on Reset has not occurred (set by firmware only)  
                   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 (set by firmware only)  
                   0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

- Note 1:** If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'.  
**Note 2:** The actual Reset value of  $\overline{POR}$  is determined by the type of device Reset. See the notes following this register and **Section 4.6 "Reset State of Registers"** for additional information.

- 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.  
**Note 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 a Power-on Rest).

**TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2450/4450) (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:
PORTC	RC7	RC6	RC5 <sup>(6)</sup>	RC4 <sup>(6)</sup>	—	RC2	RC1	RC0	xxxx -xxx	51, 106
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	51, 100
PORTA	—	RA6 <sup>(4)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	-x0x 0000	51, 100
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	00-0 0000	52, 132
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	-000 0000	52, 136
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	-0x0 000-	52, 130
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	-xxx xxx-	52, 134
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0--0 0000	52, 148
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0--0 0000	52, 147
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	-000 0000	52, 146
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	-000 0000	52, 144
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	---- -xxx	52, 136
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	xxxx xxxx	52, 136

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

- Note**
- 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.
  - 2: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.
  - 3: These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.  
4: RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.
  - 5: RE3 is only available as a port pin when the MCLRE Configuration bit is clear; otherwise, the bit reads as '0'.
  - 6: RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3> = 0).

## EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

PROGRAM_MEMORY			
	BCF	EECON1, CFGS	; access Flash program memory
	BSF	EECON1, WREN	; enable write to memory
	BCF	INTCON, GIE	; disable interrupts
<b>Required Sequence</b>	MOVLW	55h	
	MOVWF	EECON2	; write 55h
	MOVLW	0AAh	
	MOVWF	EECON2	; write 0AAh
	BSF	EECON1, WR	; start program (CPU stall)
	DECSZ	COUNTER1	
	BRA	WRITE_BUFFER_BACK	
	BSF	INTCON, GIE	; re-enable interrupts
	BCF	EECON1, WREN	; disable write to memory

### 6.5.2 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.

### 6.5.3 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. If the write operation is interrupted by a MCLR Reset or a WDT time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

### 6.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 18.0 “Special Features of the CPU”** for more detail.

## 6.6 Flash Program Operation During Code Protection

See **Section 18.5 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

**TABLE 6-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					49
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								49
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								49
TABLAT	Program Memory Table Latch								49
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
EECON2	Data Memory Control Register 2 (not a physical register)								51
EECON1	—	CFGFS	—	FREE	WRERR	WREN	WR	—	51
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	51
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	51
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	51

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during Flash access.



# PIC18F2450/4450

## 11.1 Timer1 Operation

Timer1 can operate in one of these modes:

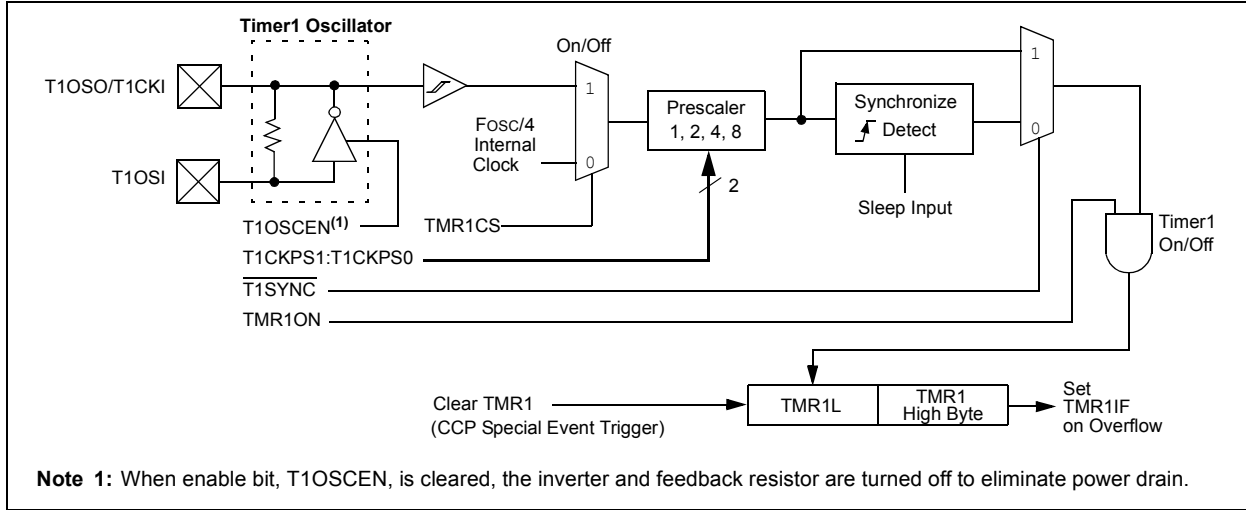
- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction

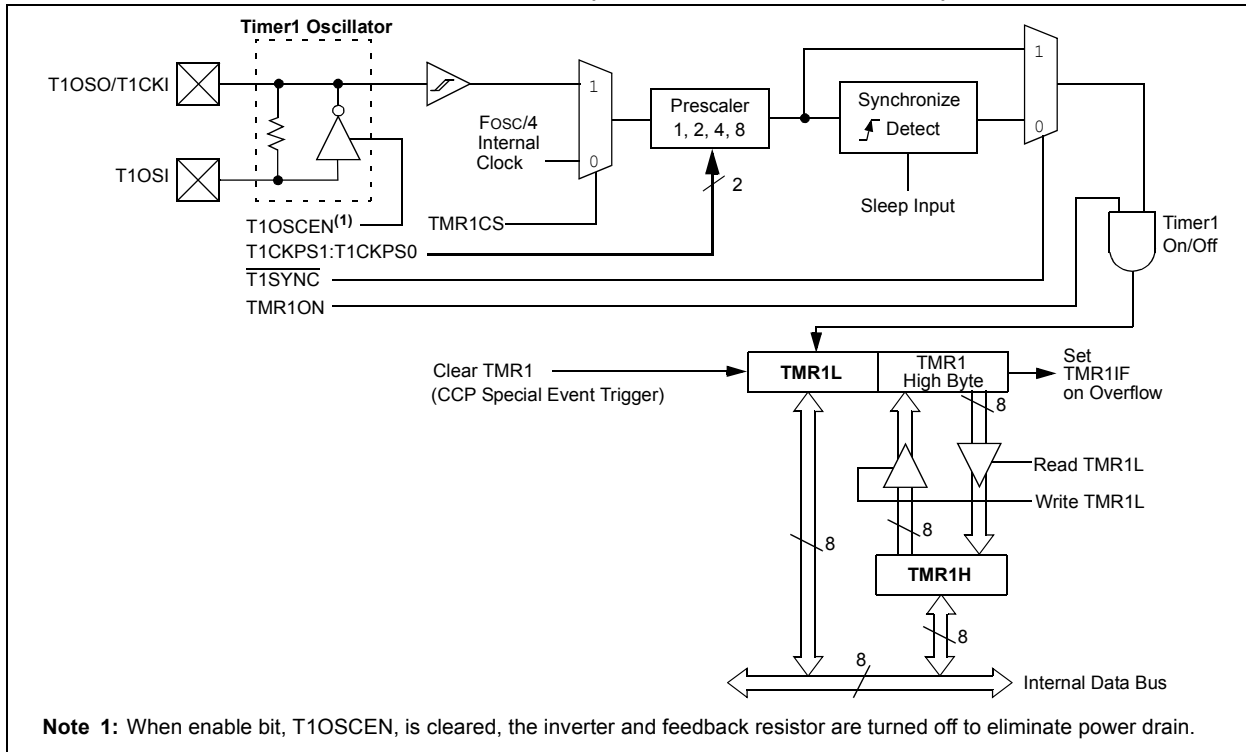
cycle ( $F_{osc}/4$ ). When the bit is set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

When Timer1 is enabled, the RC1/T1OSI/ $\overline{UOE}$  and RC0/T1OSO/T1CKI pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

**FIGURE 11-1: TIMER1 BLOCK DIAGRAM**



**FIGURE 11-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)**



# PIC18F2450/4450

## 14.5.1 USB INTERRUPT STATUS REGISTER (UIR)

The USB Interrupt Status register (Register 14-7) contains the flag bits for each of the USB status interrupt sources. Each of these sources has a corresponding interrupt enable bit in the UIE register. All of the USB status flags are ORed together to generate the USBIF interrupt flag for the microcontroller's interrupt funnel.

Once an interrupt bit has been set by the SIE, it must be cleared by software by writing a '0'. The flag bits can also be set in software which can aid in firmware debugging.

When the USB module is in the Low-Power Suspend mode (UCON<1> = 1), the SIE does not get clocked. When in this state, the SIE cannot process packets, and therefore, cannot detect new interrupt conditions other than the Activity Detect Interrupt, Flag ACTVIF. The ACTVIF bit is typically used by USB firmware to detect when the microcontroller should bring the USB module out of the Low-Power Suspend mode (UCON<1> = 0).

### REGISTER 14-7: UIR: USB INTERRUPT STATUS REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
—	SOFIF	STALLIF	IDLEIF <sup>(1)</sup>	TRNIF <sup>(2)</sup>	ACTVIF <sup>(3)</sup>	UERRIF <sup>(4)</sup>	URSTIF
bit 7							bit 0

#### Legend:

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

- bit 7      **Unimplemented:** Read as '0'
- bit 6      **SOFIF:** Start-of-Frame Token Interrupt bit  
 1 = A Start-of-Frame token received by the SIE  
 0 = No Start-of-Frame token received by the SIE
- bit 5      **STALLIF:** A STALL Handshake Interrupt bit  
 1 = A STALL handshake was sent by the SIE  
 0 = A STALL handshake has not been sent
- bit 4      **IDLEIF:** Idle Detect Interrupt bit<sup>(1)</sup>  
 1 = Idle condition detected (constant Idle state of 3 ms or more)  
 0 = No Idle condition detected
- bit 3      **TRNIF:** Transaction Complete Interrupt bit<sup>(2)</sup>  
 1 = Processing of pending transaction is complete; read USTAT register for endpoint information  
 0 = Processing of pending transaction is not complete or no transaction is pending
- bit 2      **ACTVIF:** Bus Activity Detect Interrupt bit<sup>(3)</sup>  
 1 = Activity on the D+/D- lines was detected  
 0 = No activity detected on the D+/D- lines
- bit 1      **UERRIF:** USB Error Condition Interrupt bit<sup>(4)</sup>  
 1 = An unmasked error condition has occurred  
 0 = No unmasked error condition has occurred.
- bit 0      **URSTIF:** USB Reset Interrupt bit  
 1 = Valid USB Reset occurred; 00h is loaded into UADDR register  
 0 = No USB Reset has occurred

- Note 1:** Once an Idle state is detected, the user may want to place the USB module in Suspend mode.
- Note 2:** Clearing this bit will cause the USTAT FIFO to advance (valid only for IN, OUT and SETUP tokens).
- Note 3:** This bit is typically unmasked only following the detection of a UIDLE interrupt event.
- Note 4:** Only error conditions enabled through the UEIE register will set this bit. This bit is a status bit only and cannot be set or cleared by the user.

**TABLE 15-3: BAUD RATES FOR ASYNCHRONOUS MODES**

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

## 17.2 HLVD Setup

The following steps are needed to set up the HLVD module:

1. Disable the module by clearing the HLVDEN bit (HLVDCON<4>).
2. Write the value to the HLVDL3:HLVDL0 bits that selects the desired HLVD trip point.
3. Set the VDIRMAG bit to detect high voltage (VDIRMAG = 1) or low voltage (VDIRMAG = 0).
4. Enable the HLVD module by setting the HLVDEN bit.
5. Clear the HLVD Interrupt Flag, HLVDIF (PIR2<2>), which may have been set from a previous interrupt.
6. Enable the HLVD interrupt, if interrupts are desired, by setting the HLVDIE and GIE/GIEH bits (PIE2<2> and INTCON<7>). An interrupt will not be generated until the IRVST bit is set.

## 17.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and will consume static current. The total current consumption, when enabled, is specified in electrical specification parameter D022 (Section 270 “DC Characteristics”).

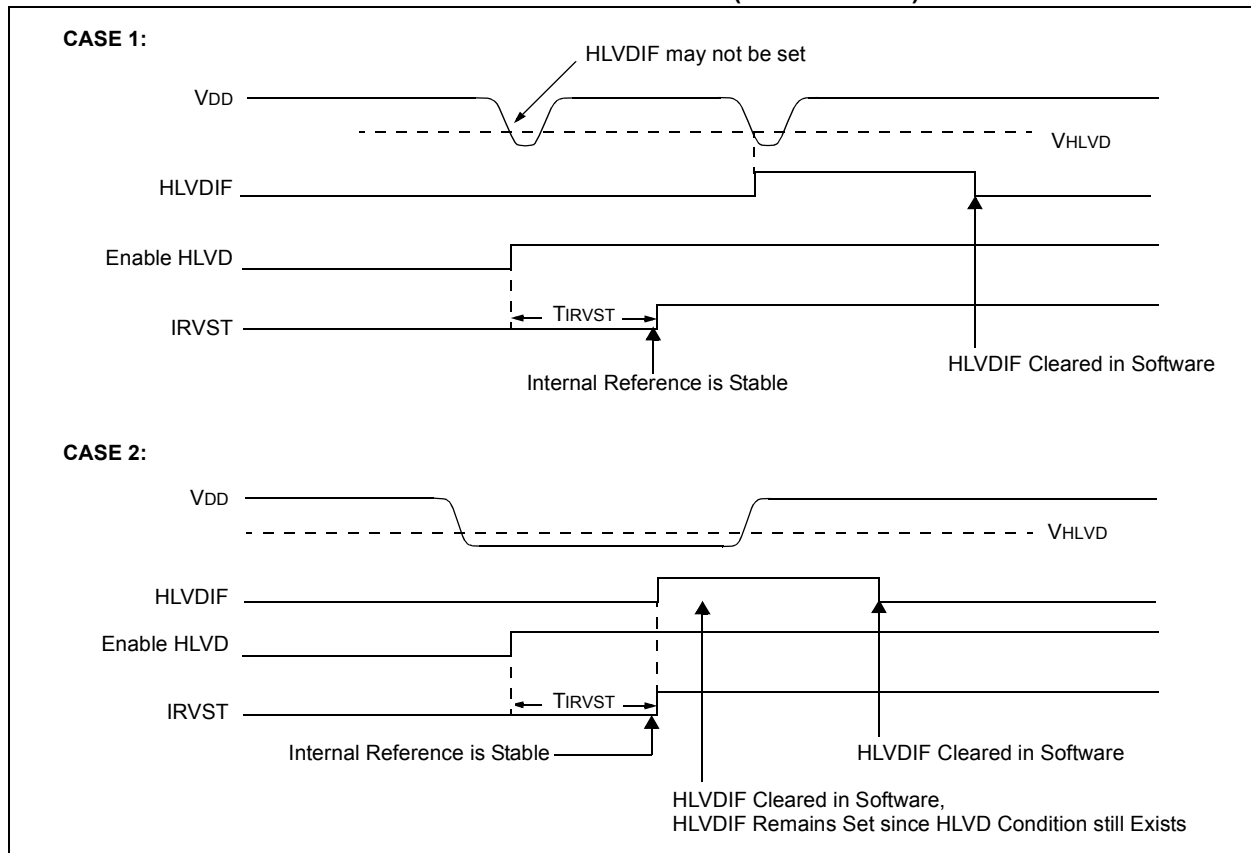
Depending on the application, the HLVD module does not need to be operating constantly. To decrease the current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After doing the check, the HLVD module may be disabled.

## 17.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification parameter D420 (see Table 21-4 in Section 21.0 “Electrical Characteristics”), may be used by other internal circuitry, such as the Programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device’s current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time,  $T_{IRVST}$ , is an interval that is independent of device clock speed. It is specified in electrical specification parameter 36 (Table 21-10).

The HLVD interrupt flag is not enabled until  $T_{IRVST}$  has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval. Refer to Figure 17-2 or Figure 17-3.

**FIGURE 17-2: LOW-VOLTAGE DETECT OPERATION (VDIRMAG = 0)**



# PIC18F2450/4450

## REGISTER 18-15: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN <sup>(1)</sup>
bit 7							bit 0

### Legend:

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

bit 7-1                      **Unimplemented:** Read as '0'  
 bit 0                      **SWDTEN:** Software Controlled Watchdog Timer Enable bit<sup>(1)</sup>  
                                     1 = Watchdog Timer is on  
                                     0 = Watchdog Timer is off

**Note 1:** This bit has no effect if the Configuration bit, WDTE, is enabled.

## TABLE 18-2: SUMMARY OF WATCHDOG TIMER REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
RCON	IPEN	SBOREN <sup>(1)</sup>	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	50
WDTCON	—	—	—	—	—	—	—	SWDTEN	50

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

**Note 1:** The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

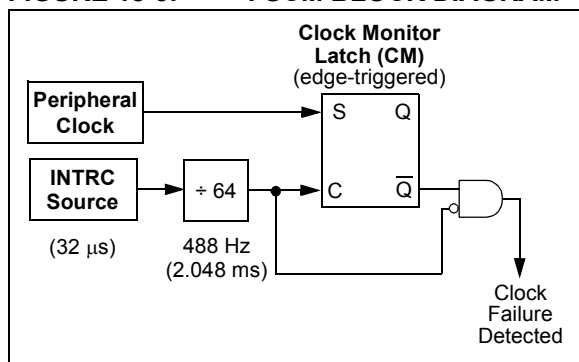
# PIC18F2450/4450

## 18.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator. The FSCM function is enabled by setting the FCMEN Configuration bit.

When FSCM is enabled, the INTRC oscillator runs at all times to monitor clocks to peripherals and provide a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 18-3) is accomplished by creating a sample clock signal, which is the INTRC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor latch (CM). The CM is set on the falling edge of the device clock source, but cleared on the rising edge of the sample clock.

**FIGURE 18-3: FSCM BLOCK DIAGRAM**



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 18-4). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- the device clock source is switched to the internal oscillator (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- the WDT is reset.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator fails, no failure would be detected, nor would any action be possible.

### 18.4.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock Monitor events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

### 18.4.2 EXITING FAIL-SAFE OPERATION

The fail-safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INTRC provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTRC. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

# PIC18F2450/4450

**COMF**                      **Complement f**

---

Syntax:                    COMF f {,d {,a}}

Operands:                 $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

Operation:                 $\bar{f} \rightarrow \text{dest}$

Status Affected:        N, Z

Encoding:                

0001	11da	ffff	ffff
------	------	------	------

Description:             The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words:                    1

Cycles:                   1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example:                COMF        REG, 0, 0

Before Instruction  
REG = 13h

After Instruction  
REG = 13h  
W = ECh

**CPFSEQ**                    **Compare f with W, Skip if f = W**

---

Syntax:                    CPFSEQ f {,a}

Operands:                 $0 \leq f \leq 255$   
 $a \in [0,1]$

Operation:                 $(f) - (W)$ ,  
skip if  $(f) = (W)$   
(unsigned comparison)

Status Affected:        None

Encoding:                

0110	001a	ffff	ffff
------	------	------	------

Description:             Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If  $f = W$ , then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words:                    1

Cycles:                   1(2)

**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:                HERE        CPFSEQ REG, 0  
NEQUAL        :  
EQUAL        :

Before Instruction  
PC Address = HERE  
W = ?  
REG = ?

After Instruction  
If REG = W;  
PC = Address (EQUAL)  
If REG  $\neq$  W;  
PC = Address (NEQUAL)

# PIC18F2450/4450

LFSR	Load FSR															
Syntax:	LFSR f, k															
Operands:	$0 \leq f \leq 2$ $0 \leq k \leq 4095$															
Operation:	$k \rightarrow \text{FSRf}$															
Status Affected:	None															
Encoding:	<table border="1"> <tr> <td>1110</td> <td>1110</td> <td>00ff</td> <td><math>k_{11}kkk</math></td> </tr> <tr> <td>1111</td> <td>0000</td> <td><math>k_7kkk</math></td> <td><math>kkkk</math></td> </tr> </table>	1110	1110	00ff	$k_{11}kkk$	1111	0000	$k_7kkk$	$kkkk$							
1110	1110	00ff	$k_{11}kkk$													
1111	0000	$k_7kkk$	$kkkk$													
Description:	The 12-bit literal 'k' is loaded into the File Select Register pointed to by 'f'.															
Words:	2															
Cycles:	2															
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read literal 'k' MSB</td> <td>Process Data</td> <td>Write literal 'k' MSB to FSRfH</td> <td></td> </tr> <tr> <td>Decode</td> <td>Read literal 'k' LSB</td> <td>Process Data</td> <td>Write literal 'k' to FSRfL</td> <td></td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSRfH		Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSRfL	
	Q1	Q2	Q3	Q4												
Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSRfH													
Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSRfL													

**Example:** LFSR 2, 3ABh

After Instruction

FSR2H = 03h  
FSR2L = ABh

MOVF	Move f										
Syntax:	MOVF f {,d {,a}}										
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$										
Operation:	$f \rightarrow \text{dest}$										
Status Affected:	N, Z										
Encoding:	<table border="1"> <tr> <td>0101</td> <td>00da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0101	00da	ffff	ffff						
0101	00da	ffff	ffff								
Description:	<p>The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"</b> for details.</p>										
Words:	1										
Cycles:	1										
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write W</td> <td></td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write W	
	Q1	Q2	Q3	Q4							
Decode	Read register 'f'	Process Data	Write W								

**Example:** MOVF REG, 0, 0

Before Instruction

REG = 22h  
W = FFh

After Instruction

REG = 22h  
W = 22h



**TSTFSZ**                    **Test f, Skip if 0**

---

Syntax:                    TSTFSZ f {,a}

Operands:                 $0 \leq f \leq 255$   
 $a \in [0,1]$

Operation:                skip if  $f = 0$

Status Affected:        None

Encoding:                

0110	011a	ffff	ffff
------	------	------	------

Description:             If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words:                    1

Cycles:                    1(2)  
**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

**Example:**

```

HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction  
PC = Address (HERE)

After Instruction  
If CNT = 00h,  
PC = Address (ZERO)  
If CNT  $\neq$  00h,  
PC = Address (NZERO)

**XORLW**                    **Exclusive OR Literal with W**

---

Syntax:                    XORLW k

Operands:                 $0 \leq k \leq 255$

Operation:                (W) .XOR. k  $\rightarrow$  W

Status Affected:        N, Z

Encoding:                

0000	1010	kkkk	kkkk
------	------	------	------

Description:             The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.

Words:                    1

Cycles:                    1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

**Example:**                    XORLW 0AFh

Before Instruction  
W = B5h

After Instruction  
W = 1Ah

# PIC18F2450/4450

---

## 19.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set of the PIC18F2450/4450 family of devices. This includes the MPLAB C18 C compiler, MPASM Assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing mode. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option, or dialog box within the environment, that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

# PIC18F2450/4450

---

## 20.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

## 20.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 20.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) for the complete list of demonstration, development and evaluation kits.

# PIC18F2450/4450

MOVWF .....	239
MULLW .....	240
MULWF .....	240
NEGF .....	241
NOP .....	241
Opcode Field Descriptions .....	214
POP .....	242
PUSH .....	242
RCALL .....	243
RESET .....	243
RETFIE .....	244
RETLW .....	244
RETURN .....	245
RLCF .....	245
RLNCF .....	246
RRCF .....	246
RRNCF .....	247
SETF .....	247
SETF (Indexed Literal Offset mode) .....	261
SLEEP .....	248
Standard Instructions .....	213
SUBFWB .....	248
SUBLW .....	249
SUBWF .....	249
SUBWFB .....	250
SWAPF .....	250
TBLRD .....	251
TBLWT .....	252
TSTFSZ .....	253
XORLW .....	253
XORWF .....	254
INTCON Register	
RBIF Bit .....	101
INTCON Registers .....	87
Internal Oscillator Block	
INTHS, INTXT, INTCKO and INTIO Modes .....	27
Internal RC Oscillator	
Use with WDT .....	203
Internet Address .....	319
Interrupt Sources .....	191
A/D Conversion Complete .....	179
Capture Complete (CCP) .....	124
Compare Complete (CCP) .....	125
Interrupt-on-Change (RB7:RB4) .....	101
INTx Pin .....	97
PORTB, Interrupt-on-Change .....	97
TMR0 .....	97
TMR0 Overflow .....	113
TMR1 Overflow .....	115
TMR2 to PR2 Match (PWM) .....	127
Interrupts .....	85
USB .....	85
Interrupts, Flag Bits	
Interrupt-on-Change (RB7:RB4)	
Flag (RBIF Bit) .....	101
INTOSC, INTRC. See Internal Oscillator Block.	
IORLW .....	236
IORWF .....	236
IPR Registers .....	94
<b>L</b>	
LFSR .....	237
Low-Voltage ICSP Programming. See Single-Supply ICSP Programming.	
<b>M</b>	
Master Clear Reset ( $\overline{\text{MCLR}}$ ) .....	43
Memory Organization .....	53
Data Memory .....	59
Program Memory .....	53
Memory Programming Requirements .....	280
Microchip Internet Web Site .....	319
Migration from Baseline to Enhanced Devices .....	309
Migration from High-End to Enhanced Devices .....	310
Migration from Mid-Range to Enhanced Devices .....	310
MOVF .....	237
MOVFF .....	238
MOVLB .....	238
MOVLW .....	239
MOVSF .....	257
MOVSS .....	258
MOVWF .....	239
MPLAB ASM30 Assembler, Linker, Librarian .....	264
MPLAB ICD 2 In-Circuit Debugger .....	265
MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator .....	265
MPLAB Integrated Development Environment Software .....	263
MPLAB PM3 Device Programmer .....	265
MPLAB REAL ICE In-Circuit Emulator System .....	265
MPLINK Object Linker/MPLIB Object Librarian .....	264
MULLW .....	240
MULWF .....	240
<b>N</b>	
NEGF .....	241
NOP .....	241
<b>O</b>	
Oscillator Configuration .....	23
EC .....	23
ECIO .....	23
ECPIO .....	23
ECPLL .....	23
HS .....	23
HSPLL .....	23
INTCKO .....	23
Internal Oscillator Block .....	27
INTHS .....	23
INTIO .....	23
INTXT .....	23
Oscillator Modes and USB Operation .....	24
XT .....	23
XTPLL .....	23
Oscillator Selection .....	191
Oscillator Settings for USB .....	27
Oscillator Start-up Timer (OST) .....	32, 45
Oscillator Switching .....	30
Oscillator Transitions .....	30
Oscillator, Timer1 .....	115
<b>P</b>	
Packaging Information .....	295
Details .....	297
Marking .....	295
PICkit 2 Development Programmer .....	266
PICSTART Plus Development Programmer .....	266
PIE Registers .....	92

# PIC18F2450/4450

PRI_IDLE Mode .....	38	HLVDCON (High/Low-Voltage Detect Control) .....	185
PRI_RUN Mode .....	34	INTCON (Interrupt Control) .....	87
Program Counter .....	54	INTCON2 (Interrupt Control 2) .....	88
PCL, PCH and PCU Registers .....	54	INTCON3 (Interrupt Control 3) .....	89
PCLATH and PCLATU Registers .....	54	IPR1 (Peripheral Interrupt Priority 1) .....	94
Program Memory .....		IPR2 (Peripheral Interrupt Priority 2) .....	95
and the Extended Instruction Set .....	70	OSCCON (Oscillator Control) .....	31
Code Protection .....	209	PIE1 (Peripheral Interrupt Enable 1) .....	92
Instructions .....	58	PIE2 (Peripheral Interrupt Enable 2) .....	93
Two-Word .....	58	PIR1 (Peripheral Interrupt Request (Flag) 1) .....	90
Interrupt Vector .....	53	PIR2 (Peripheral Interrupt Request (Flag) 2) .....	91
Look-up Tables .....	56	PORTE .....	109
Map and Stack (diagram) .....	53	RCON (Reset Control) .....	42, 96
Reset Vector .....	53	RCSTA (Receive Status and Control) .....	155
Program Verification and Code Protection .....	208	STATUS .....	66
Associated Registers .....	208	STKPTR (Stack Pointer) .....	55
Programming, Device Instructions .....	213	T0CON (Timer0 Control) .....	111
Pulse-Width Modulation. See PWM (CCP Module).		T1CON (Timer1 Control) .....	115
PUSH .....	242	T2CON (Timer2 Control) .....	121
PUSH and POP Instructions .....	55	TXSTA (Transmit Status and Control) .....	154
PUSHL .....	258	UCFG (USB Configuration) .....	132
PWM (CCP Module) .....		UCON (USB Control) .....	130
Associated Registers .....	128	UEIE (USB Error Interrupt Enable) .....	148
Duty Cycle .....	127	UEIR (USB Error Interrupt Status) .....	147
Example Frequencies/Resolutions .....	128	UEPn (USB Endpoint n Control) .....	135
Period .....	127	UIE (USB Interrupt Enable) .....	146
Setup for PWM Operation .....	128	UIR (USB Interrupt Status) .....	144
TMR2 to PR2 Match .....	127	USTAT (USB Status) .....	134
		WDTCON (Watchdog Timer Control) .....	204
<b>Q</b>		RESET .....	243
Q Clock .....	128	Reset State of Registers .....	48
		Reset Timers .....	45
<b>R</b>		Oscillator Start-up Timer (OST) .....	45
RAM. See Data Memory.		PLL Lock Time-out .....	45
RC_IDLE Mode .....	39	Power-up Timer (PWRT) .....	45
RC_RUN Mode .....	36	Resets .....	41, 191
RCALL .....	243	Brown-out Reset (BOR) .....	191
RCON Register .....		Oscillator Start-up Timer (OST) .....	191
Bit Status During Initialization .....	48	Power-on Reset (POR) .....	191
Reader Response .....	320	Power-up Timer (PWRT) .....	191
Register File Summary .....	63–65	RETFIE .....	244
Registers .....		RETLW .....	244
ADCON0 (A/D Control 0) .....	175	RETURN .....	245
ADCON1 (A/D Control 1) .....	176	Return Address Stack .....	54
ADCON2 (A/D Control 2) .....	177	and Associated Registers .....	54
BAUDCON (Baud Rate Control) .....	156	Return Stack Pointer (STKPTR) .....	55
BDnSTAT (Buffer Descriptor n Status, CPU Mode) .....	139	Revision History .....	307
BDnSTAT (Buffer Descriptor n Status, SIE Mode) .....	140	RLCF .....	245
CCP1CON (Capture/Compare/PWM Control) .....	123	RLNCF .....	246
CONFIG1H (Configuration 1 High) .....	194	RRCF .....	246
CONFIG1L (Configuration 1 Low) .....	193	RRNCF .....	247
CONFIG2H (Configuration 2 High) .....	196		
CONFIG2L (Configuration 2 Low) .....	195	<b>S</b>	
CONFIG3H (Configuration 3 High) .....	197	SEC_IDLE Mode .....	38
CONFIG4L (Configuration 4 Low) .....	198	SEC_RUN Mode .....	34
CONFIG5H (Configuration 5 High) .....	199	SETF .....	247
CONFIG5L (Configuration 5 Low) .....	199	Single-Supply ICSP Programming .....	212
CONFIG6H (Configuration 6 High) .....	200	SLEEP .....	248
CONFIG6L (Configuration 6 Low) .....	200	Sleep .....	
CONFIG7H (Configuration 7 High) .....	201	OSC1 and OSC2 Pin States .....	32
CONFIG7L (Configuration 7 Low) .....	201	Sleep Mode .....	37
DEVID1 (Device ID 1) .....	202	Software Simulator (MPLAB SIM) .....	264
DEVID2 (Device ID 2) .....	202	Special Event Trigger. See Compare (CCP Module).	
EECON1 (Memory Control 1) .....	75	Special Features of the CPU .....	191
		Special ICPORT Features .....	211