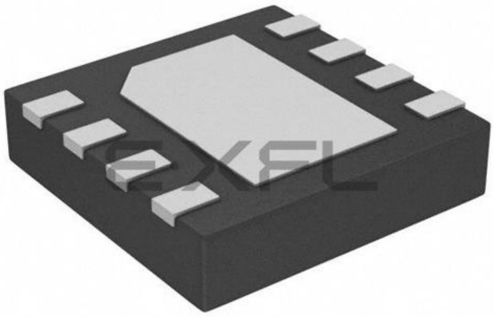


Welcome to E-XFL.COM

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.

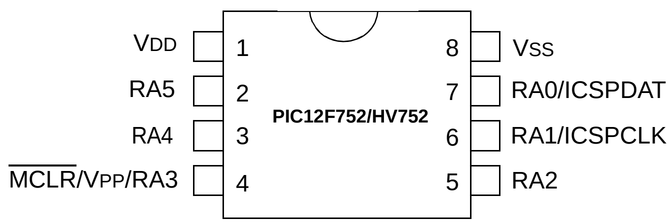
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	5
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f752-e-mf

TABLE 1: PIC12F752/HV752 FEATURE SUMMARY

Device	Flash Program Memory (User) (words)	Self Read/Write Flash Memory	SRAM (bytes)	I/Os	10-bit A/D (ch)	Comparators	Timers 8/16-bit	CCP	Complementary Output Generator (COG)	Shunt Regulator	XLP
PIC12F752	1024	Y	64	6	4	2	3/1	1	Y	N	Y
PIC12HV752	1024	Y	64	6	4	2	3/1	1	Y	Y	Y

FIGURE 1: 8-PIN PDIP, SOIC, DFN



Note: See Table 1 for the location of all peripheral functions.

2.3.3 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, IOCIE change and external RA2/INT pin interrupts.

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

REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF
bit 7							bit 0

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

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** RA2/INT External Interrupt Enable bit
1 = Enables the RA2/INT external interrupt
0 = Disables the RA2/INT external interrupt

bit 3 **IOCIE:** Interrupt-on-Change Interrupt Enable bit(1)
1 = Enables the IOC change interrupt
0 = Disables the IOC change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = Timer0 register has overflowed (must be cleared in software)
0 = Timer0 register did not overflow

bit 1 **INTF:** RA2/INT External Interrupt Flag bit
1 = The RA2/INT external interrupt occurred (must be cleared in software)
0 = The RA2/INT external interrupt did not occur

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit
1 = An IOC pin has changed state and generated an interrupt
0 = No pin interrupts have been generated

Note 1: IOC register must also be enabled.
2: T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.

2.3.8 PCON REGISTER

The Power Control (PCON) register (see Table 17-2) contains flag bits to differentiate between:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Reset ($\overline{\text{BOR}}$)
- Watchdog Timer Reset (WDT)
- External $\overline{\text{MCLR}}$ Reset

The PCON register also controls the software enable of the $\overline{\text{BOR}}$.

The PCON register bits are shown in Register 2-8.

REGISTER 2-8: PCON: POWER CONTROL REGISTER

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

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-2 **Unimplemented:** Read as ‘0’

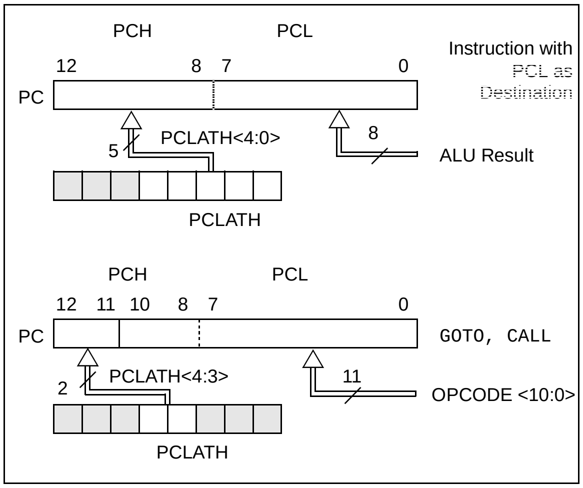
bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

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

2.4 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-2 shows the two situations for the loading of the PC. The upper example in Figure 2-2 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-2 shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

FIGURE 2-2: LOADING OF PC IN DIFFERENT SITUATIONS



2.4.1 MODIFYING PCL

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

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

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

2.4.2 STACK

The PIC12F752/HV752 Family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN, RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

Note 1:	There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL, RETURN, RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

2.5 Indirect Addressing, INDF and FSR Registers

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

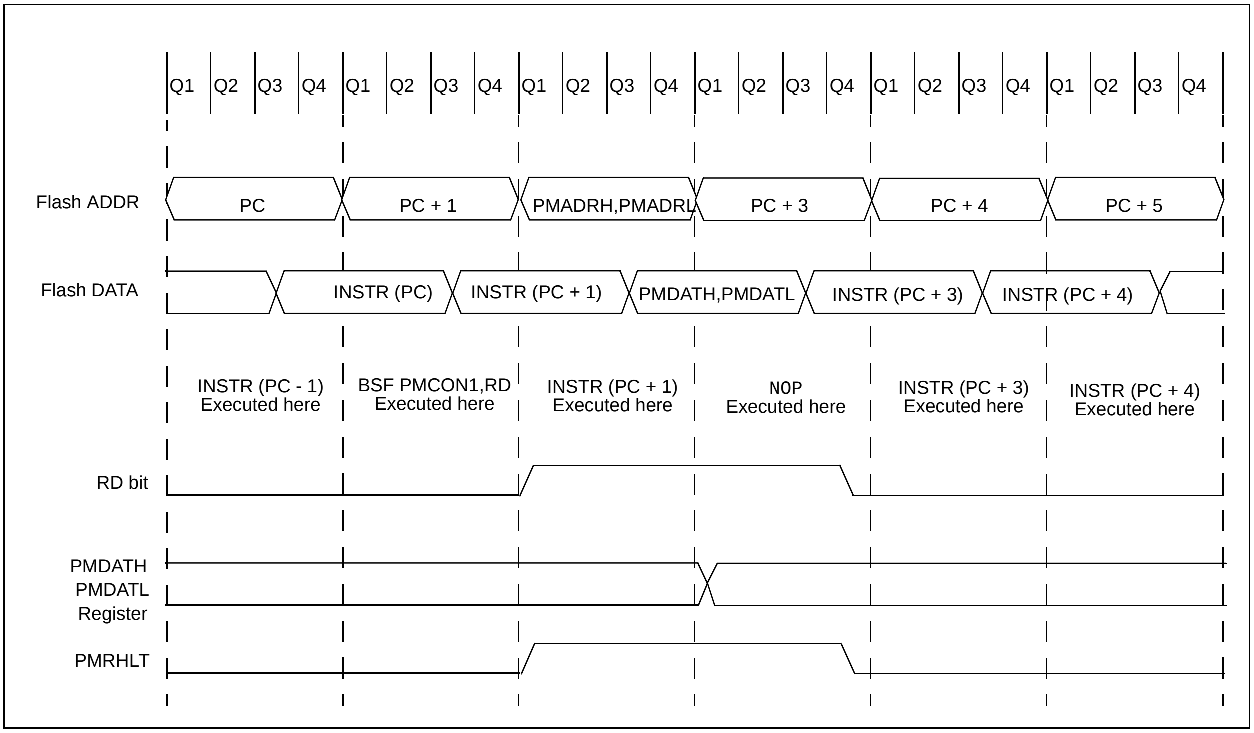
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit of the STATUS register, as shown in Figure 2-3.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 2-1.

EXAMPLE 2-1: INDIRECT ADDRESSING

```
         MOVLW   0x40    ;initialize pointer
         MOVWF   FSR     ;to RAM
NEXT     CLRF    INDF    ;clear INDF register
         INCF    FSR     ;inc pointer
         BTFSS   FSR,7   ;all done?
         GOTO    NEXT    ;no clear next
CONTINUE                 ;yes continue
```


FIGURE 3-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION



4.2 Clock Source Modes

Clock Source modes can be classified as external or internal:

- The External Clock mode relies on an external clock for the clock source, such as a clock module or clock output from another circuit.
- Internal clock sources are contained internally within the oscillator module. The oscillator module has four selectable clock frequencies:
  - 8 MHz
  - 4 MHz
  - 1 MHz
  - 31 kHz

The system clock can be selected between external or internal clock sources via the FOSC0 bit of the Configuration Word register (CONFIG).

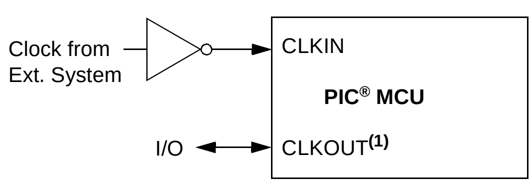
4.2.1 EC MODE

The External Clock (EC) mode allows an externally generated logic as the system clock source. The EC clock mode is selected when the FOSC0 bit of the Configuration Word is set.

When operating in this mode, an external clock source must be connected to the CLKIN input. The CLKOUT is available for either general purpose I/O or system clock output. Figure 4-3 shows the pin connections for EC mode.

Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 4-3: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Alternate pin functions are listed in the **Section 1.0 "Device Overview"**.

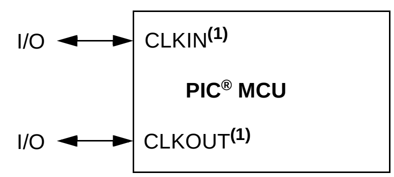
4.2.2 INTERNAL CLOCK MODE

Internal Clock mode configures the internal oscillators as the system clock source. The Internal Clock mode is selected when the FOSC0 bit of the Configuration Word is cleared. The source and frequency are selected with the IRCF<1:0> bits of the OSCCON register.

When one of the HFINTOSC frequencies is selected, the frequency of the internal oscillator can be trimmed by adjusting the TUN<4:0> bits of the OSCTUNE register.

Operation after a Power-on Reset (POR) or wake-up from Sleep is delayed by the oscillator start-up time. Delays are typically longer for the LFINTOSC than HFINTOSC because of the very low-power operation and relatively narrow bandwidth of the LF internal oscillator. However, when another peripheral keeps the oscillator running during Sleep, the start-up time is delayed to allow the memory bias to stabilize.

FIGURE 4-4: INTERNAL CLOCK MODE OPERATION

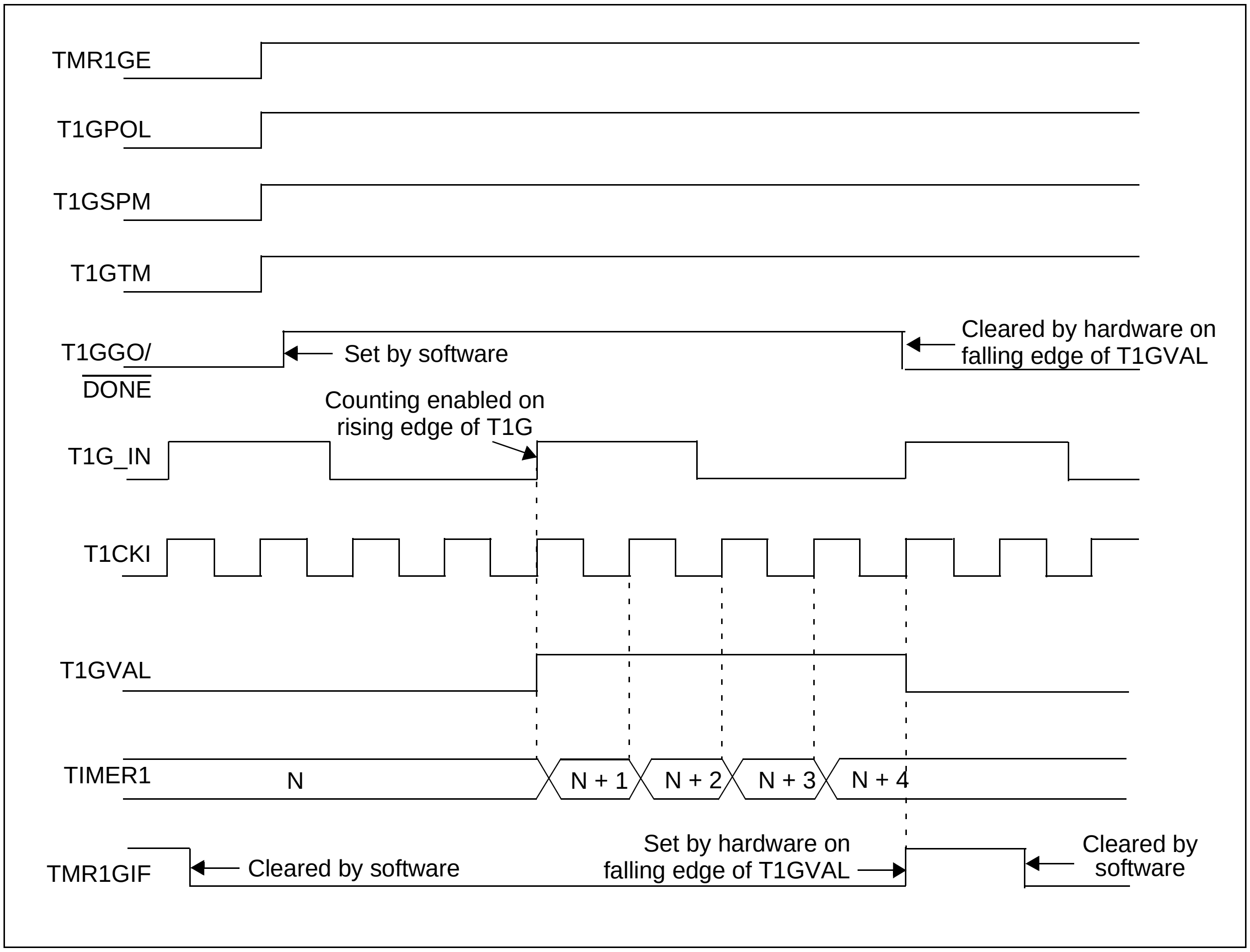


Note 1: Alternate pin functions are listed in the **Section 1.0 "Device Overview"**.

4.2.2.1 Oscillator Ready Bits

The HTS and LTS bits of the OSCCON register indicate the status of the HFINTOSC and LFINTOSC, respectively. When either bit is set, it indicates that the corresponding oscillator is running and stable.

FIGURE 7-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



8.2 Timer2 Control Registers

REGISTER 8-1: T2CON: TIMER 2 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
—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
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 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits
- 0000 = 1:1 Postscaler
- 0001 = 1:2 Postscaler
- 0010 = 1:3 Postscaler
- 0011 = 1:4 Postscaler
- 0100 = 1:5 Postscaler
- 0101 = 1:6 Postscaler
- 0110 = 1:7 Postscaler
- 0111 = 1:8 Postscaler
- 1000 = 1:9 Postscaler
- 1001 = 1:10 Postscaler
- 1010 = 1:11 Postscaler
- 1011 = 1:12 Postscaler
- 1100 = 1:13 Postscaler
- 1101 = 1:14 Postscaler
- 1110 = 1:15 Postscaler
- 1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
- 1 = Timer2 is on
- 0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
- 00 = Prescaler is 1
- 01 = Prescaler is 4
- 1x = Prescaler is 16

TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	15
PIE1	TMR1GIE	ADIE	—	—	—	HLTMR1IE	TMR2IE	TMR1IE	16
PIR1	TMR1GIF	ADIF	—	—	—	HLTMR1IF	TMR2IF	TMR1IF	18
PR2	Timer2 Module Period Register								61*
TMR2	Holding Register for the 8-bit TMR2 Register								61*
T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		62

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.
* Page provides register information.

10.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM modules is a peripheral which allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

10.1 Capture Mode

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCP1 pin, the 16-bit CCPR1H:CCPR1L register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCP1M<3:0> bits of the CCP1CON register:

- Every Falling Edge
- Every Rising Edge
- Every 4th Rising Edge
- Every 16th Rising Edge

When a capture is made, the Interrupt Request Flag bit CCP1IF of the PIR2 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPR1H, CCPR1L register pair is read, the old captured value is overwritten by the new captured value.

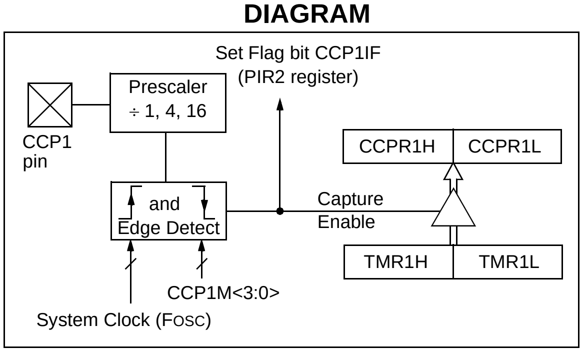
Figure 10-1 shows a simplified diagram of the Capture operation.

10.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the CCP1 pin should be configured as an input by setting the associated TRIS control bit.

> **Note:** If the CCP1 pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 10-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



10.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP1 module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 7.0 "Timer1 Module with Gate Control"** for more information on configuring Timer1.

10.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit of the PIE2 register clear to avoid false interrupts. Additionally, the user should clear the CCP1IF interrupt flag bit of the PIR2 register following any change in Operating mode.

> **Note:** Clocking Timer1 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCP1 pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

10.1.4 CCP1 PRESCALER

There are four prescaler settings specified by the CCP1M<3:0> bits of the CCP1CON register. Whenever the CCP1 module is turned off or the CCP1 module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCP1CON register before changing the prescaler. Example 10-1 demonstrates the code to perform this function.

EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCP1CON     ;Set Bank bits to point
                    ;to CCP1CON
CLRF    CCP1CON     ;Turn CCP1 module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP1 ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ;value
```

11.7 Phase Delay

It is possible to delay the assertion of the rising event. This is accomplished by placing a non-zero value in COGxPH register. Refer to Register 11-6 and Figure 11-3 for COG operation with CCP1 and phase delay. The delay from the input rising event signal switching to the actual assertion of the events is calculated the same as the dead-band and blanking delays. Please see Equation 11-1.

When the COGxPH value is '0', phase delay is disabled and the phase delay counter output is true, thereby, allowing the event signal to pass straight through to complementary output driver flop.

11.7.1 CUMULATIVE UNCERTAINTY

It is not possible to create more than one COG_clock of uncertainty by successive stages. Consider that the phase delay stage comes after the blanking stage, the dead-band stage comes after either the blanking or phase delay stages, and the blanking stage comes after the dead-band stage. When the preceding stage is enabled, the output of that stage is necessarily synchronous with the COG_clock, which removes any possibility of uncertainty in the succeeding stage.

EQUATION 11-1: PHASE, DEAD-BAND, AND BLANKING TIME CALCULATION

$$T_{\text{min}} = \frac{\text{Count}}{F_{COG\_clock}}$$

$$T_{\text{max}} = \frac{\text{Count} + 1}{F_{COG\_\text{clock}}}$$

$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}}$$

Also:

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}}$$

Where:

T	Count
Phase Delay	GxPH<3:0>
Rising Dead Band	GxDBR<3:0>
Falling Dead Band	GxDBF<3:0>
Rising Event Blanking	GxBLKR<3:0>
Falling Event Blanking	GxBLKF<3:0>

EXAMPLE 11-1: TIMER UNCERTAINTY

Given:

$$Count = Ah = 10d$$
$$F_{COG\_\text{Clock}} = 8\text{MHz}$$

Therefore:

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}} = \frac{1}{8MHz} = 125ns$$

Proof:

$$T_{\text{min}} = \frac{Count}{F_{COG\_\text{clock}}} = 125ns \bullet 10d = 1.25\mu s$$

$$T_{\text{max}} = \frac{Count + 1}{F_{COG\_\text{clock}}} = 125ns \bullet (10d + 1) = 1.375\mu s$$

Therefore:

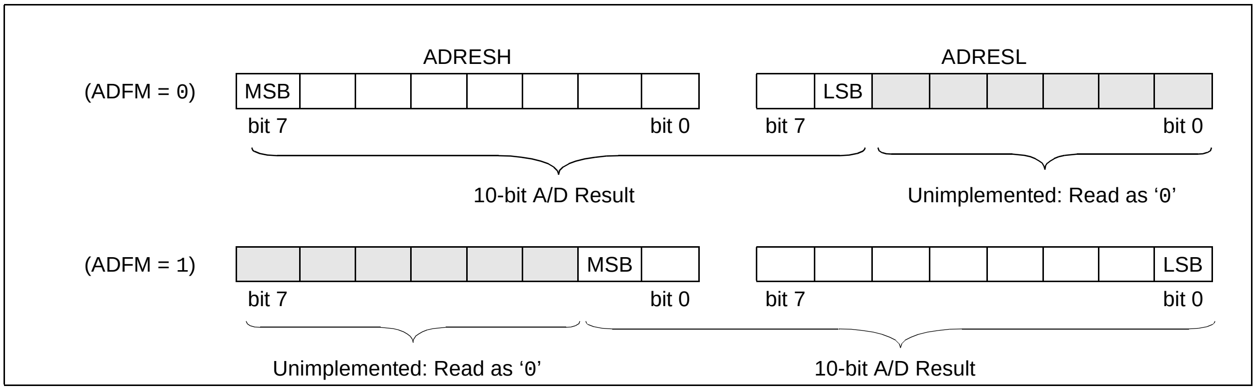
$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}} = 1.375\mu s - 1.25\mu s = 125ns$$

12.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 12-4 shows the two output formats.

FIGURE 12-3: 10-BIT A/D CONVERSION RESULT FORMAT



12.2 ADC Operation

12.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 12.2.6 "A/D Conversion Procedure"**.

12.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

12.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 T_{AD} delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

12.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

12.2.5 SPECIAL EVENT TRIGGER

The CCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See **Section 10.0 "Capture/Compare/PWM Modules"** for more information.

12.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (See TRIS register)
   - Configure pin as analog
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Select result format
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2)
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result
8. Clear the ADC interrupt flag (required if interrupt is enabled)

Note 1:	The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.
2:	See **Section 12.4 "A/D Acquisition Requirements"**.

EXAMPLE 12-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and RA0 input.
;
;Conversion start & polling for completion
; are included.
;
  BANKSEL ADCON1      ;
  BSF     TRISA,0     ;Set RA0 to input
  BANKSEL ADCON1      ;
  MOVLW   B'01110000' ;ADC Frc clock,
  IORWF   ADCON1      ; and RA0 as analog
  BANKSEL ADCON0      ;
  MOVLW   B'10000001' ;Right justify,
  MOVWF   ADCON0      ;Vdd Vref, AN0, On
  CALL    SampleTime  ;Acquisiton delay
  BSF     ADCON0,GO   ;Start conversion
TEST AGAIN
  BTFSC   ADCON0,GO   ;Is conversion done?
  GOTO    TEST AGAIN  ;No, test again
  BANKSEL ADRESH      ;
  MOVF    ADRESH,W    ;Read upper 2 bits
  MOVWF   RESULTHI    ;Store in GPR space
  BANKSEL ADRESL      ;
  MOVF    ADRESL,W    ;Read lower 8 bits
  MOVWF   RESULTLO    ;Store in GPR space
```

14.8 DAC Control Registers

REGISTER 14-1: DACCON0: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	U-0	U-0
DACEN	DACRNG	DACOE	—	—	DACPSS	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **DACRNG:** DAC Range Selection bit(1)
1 = DAC is operating in Full Range mode
0 = DAC is operating in Limited Range mode

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC reference output is enabled to the DACOUT pin(2)
0 = DAC reference output is disabled

bit 4-3 **Unimplemented:** Read as ‘0’

bit 2 **DACPSS:** DAC Positive Source Select bits
0 = V_{DD}
1 = $V_{REF}+$ pin

bit 1-0 **Unimplemented:** Read as ‘0’

Note 1: Refer to Equation 14-1.
2: The DACOUT pin configuration requires additional control bits in the FVRCON register (see Figure 14-3).

REGISTER 14-2: DACCON1: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DACR<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **DACR<4:0>:** DAC Voltage Output Select bits
1 1111 = DAC Voltage Maximum Output
•
•
•
0 0000 = DAC Voltage Minimum Output

Note 1: Refer to Equation 14-1 to calculate the value of the DAC Voltage Output.

REGISTER 15-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	—	—	CxNCH0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bit
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-4 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
00 = CxVP connects to CxIN+ pin
01 = CxVP connects to DAC Voltage Reference (dac_ref)
10 = CxVP connects to FVR Voltage Reference (fvr_ref)
11 = CxVP connects to Vss

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **CxNCH0:** Comparator Negative Input Channel Select bits
0 = CxVN connects to CxIN0- pin
1 = CxVN connects to CxIN1- pin

REGISTER 15-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

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

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

TABLE 16-2: PIC12F752/HV752 INSTRUCTION SET

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

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTA, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register 'f' is moved to a destination dependent upon the status of 'd'. If d = 0, destination is W register. If d = 1, the destination is file register 'f' itself. d = 1 is useful to test a file register since Status flag Z is affected.
Words:	1
Cycles:	1
Example:	`MOVF FSR, 0` After Instruction W = value in FSR register Z = 1

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \leq f \leq 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	`MOVW OPTION F` Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	k → (W)
Status Affected:	None
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	`MOVLW 0x5A` After Instruction W = 0x5A

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation
Words:	1
Cycles:	1
Example:	`NOP`

17.3.4 BROWN-OUT RESET (BOR)

The BOREN<1:0> bits in the Configuration Word register select one of three BOR modes. One mode has been added to allow control of the BOR enable for lower current during Sleep. By selecting BOREN<1:0> = 10, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. See Register 17-1 for the Configuration Word definition.

A brown-out occurs when VDD falls below VBOR for greater than parameter TBOR (see **Section 20.0 "Electrical Specifications"**). The brown-out condition will reset the device. This will occur regardless of VDD slew rate. A Brown-out Reset may not occur if VDD falls below VBOR for less than parameter TBOR.

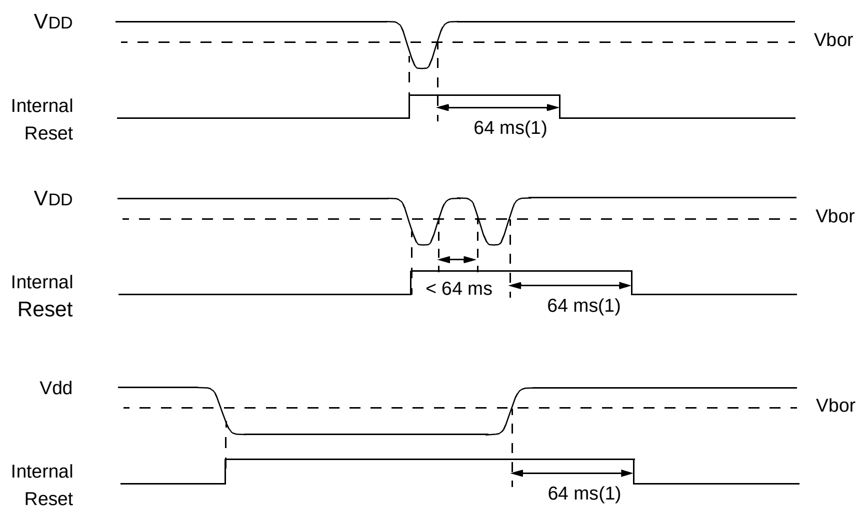
On any Reset (Power-on, Brown-out Reset, Watchdog timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 17-3). If enabled, the Power-up Timer will be invoked by the Reset and keep the chip in Reset an additional 64 ms.

Note:	The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word register.

If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

Table 17-3 summarizes the registers associated with BOR.

FIGURE 17-3: BROWN-OUT SITUATIONS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

TABLE 17-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT RESET

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	20
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	13

Legend: u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by BOR.

Note 1: Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

20.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

Ambient temperature under bias .... -40° to +125°C
Storage temperature .... -65°C to +150°C
Voltage on pins with respect to V_{SS}
on V_{DD} pin
PIC12HV752 .... -0.3V to +6.5V
PIC12F752 .... -0.3V to +6.5V
on $\overline{MCLR}$ .... -0.3V to +13.5V
on all other pins .... -0.3V to (V_{DD} + 0.3V)
Maximum current
on V_{SS} pin(1)
-40°C ≤ T_A ≤ +85°C .... 95 mA
-40°C ≤ T_A ≤ +125°C .... 95 mA
on V_{DD} pin(1)
-40°C ≤ T_A ≤ +85°C .... 95 mA
-40°C ≤ T_A ≤ +125°C .... 95 mA
on RA1, RA4, RA5 .... 25 mA
on RA0, RA2 .... 50 mA
Clamp current, I_K (V_{PIN} < 0 or V_{PIN} > V_{DD}) .... ± 20 mA

Note 1: Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characteristics. See Table 20-6 to calculate device specific limitations.

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

20.3 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

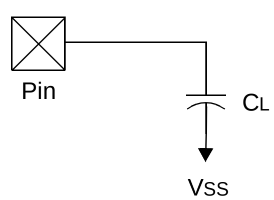
pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	Data in	t0	T0CKI
io	I/O Port	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-Impedance)	V	Valid
L	Low	Z	High-Impedance

FIGURE 20-3: LOAD CONDITIONS

Load Condition



Note: C_L = 50 pF for all pins.

TABLE 20-8: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
OS06	TWARM	Internal Oscillator Switch when running	—	—	—	2	TOSC	
OS07	INTOSC	Internal Calibrated INTOSC Frequency(1) (4 MHz)	±1%	3.96	4.0	4.04	MHz	VDD = 3.5V, TA = 25°C
			±2%	3.92	4.0	4.08	MHz	2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C
			±5%	3.80	4.0	4.20	MHz	2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.)
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency(1)	±1%	7.92	8	8.08	MHz	VDD = 3.5V, TA = 25°C
			±2%	7.84	8	8.16	MHz	2.5V ≤ VDD ≤ 5.5V, 0°C ≤ TA ≤ +85°C
			±5%	7.60	8	8.40	MHz	2.0V ≤ VDD ≤ 5.5V, -40°C ≤ TA ≤ +85°C (Ind.), -40°C ≤ TA ≤ +125°C (Ext.)
OS09	LFOSC	Internal LFINTOSC Frequency	—	—	31	—	kHz	
OS10*	TIOSC ST	HFINTOSC Wake-up from Sleep Start-up Time	—	—	12	24	μs	VDD = 2.0V -40°C ≤ TA ≤ +85°C
				—	7	14	μs	VDD = 3.0V -40°C ≤ TA ≤ +85°C
				—	6	11	μs	VDD = 5.0V -40°C ≤ TA ≤ +85°C

* These parameters are characterized but not tested.

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

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

