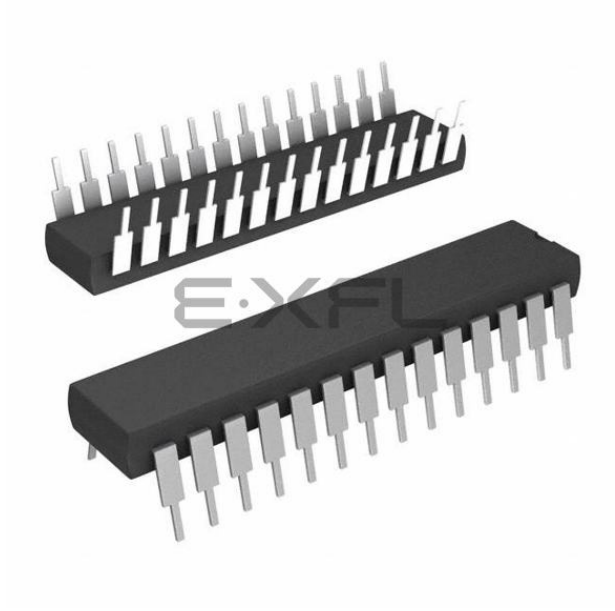


Welcome to E-XFL.COM

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	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1512-e-sp

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1512 PARTS

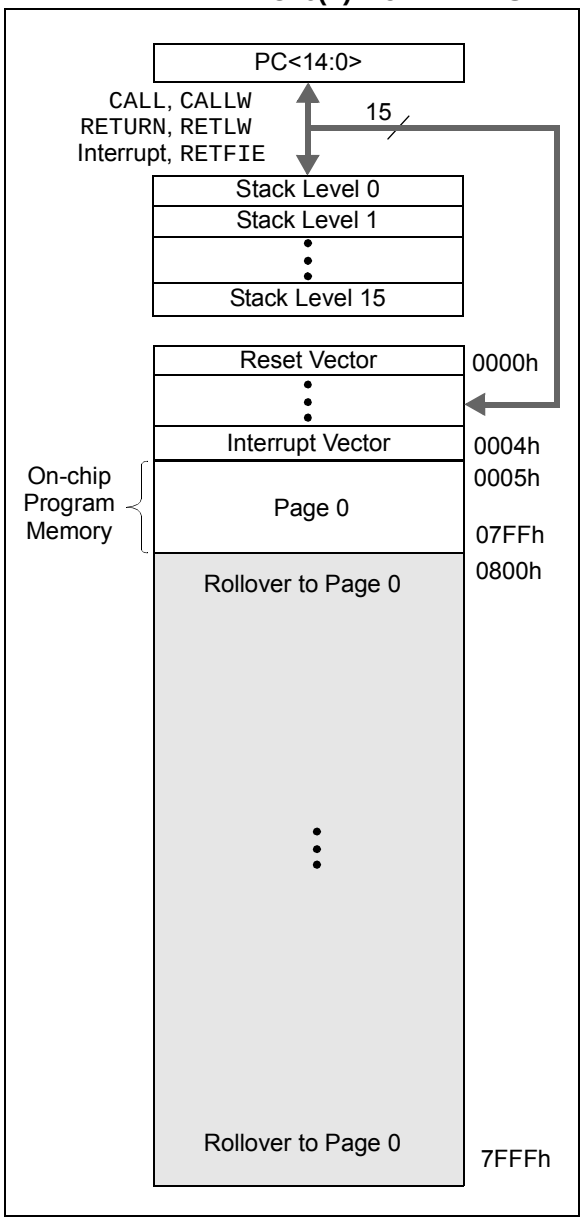


FIGURE 3-2: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1513 PARTS

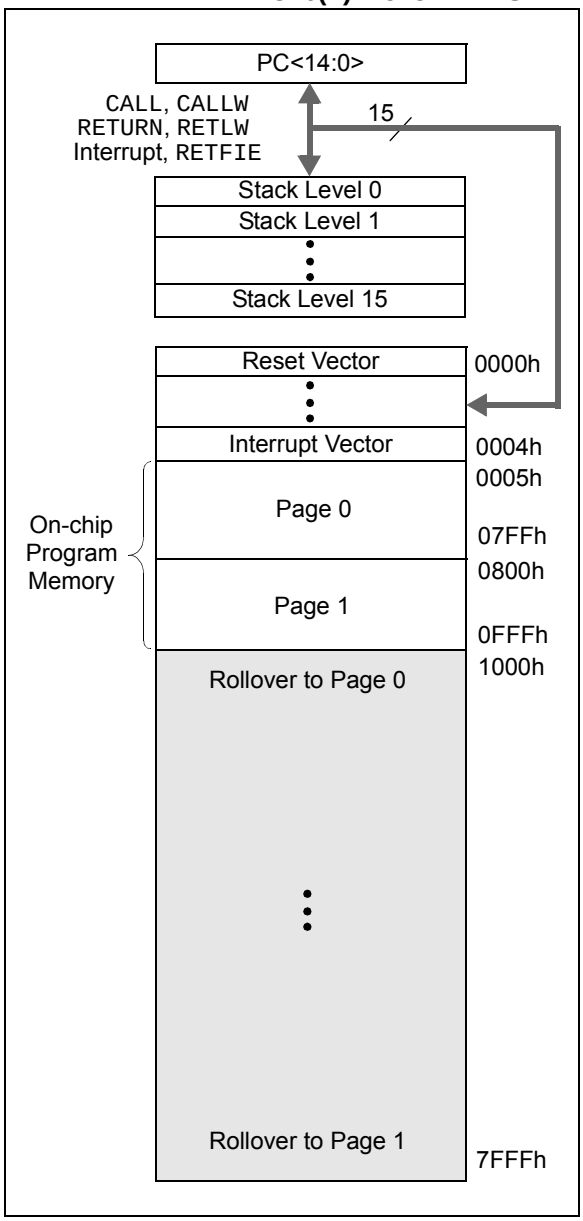
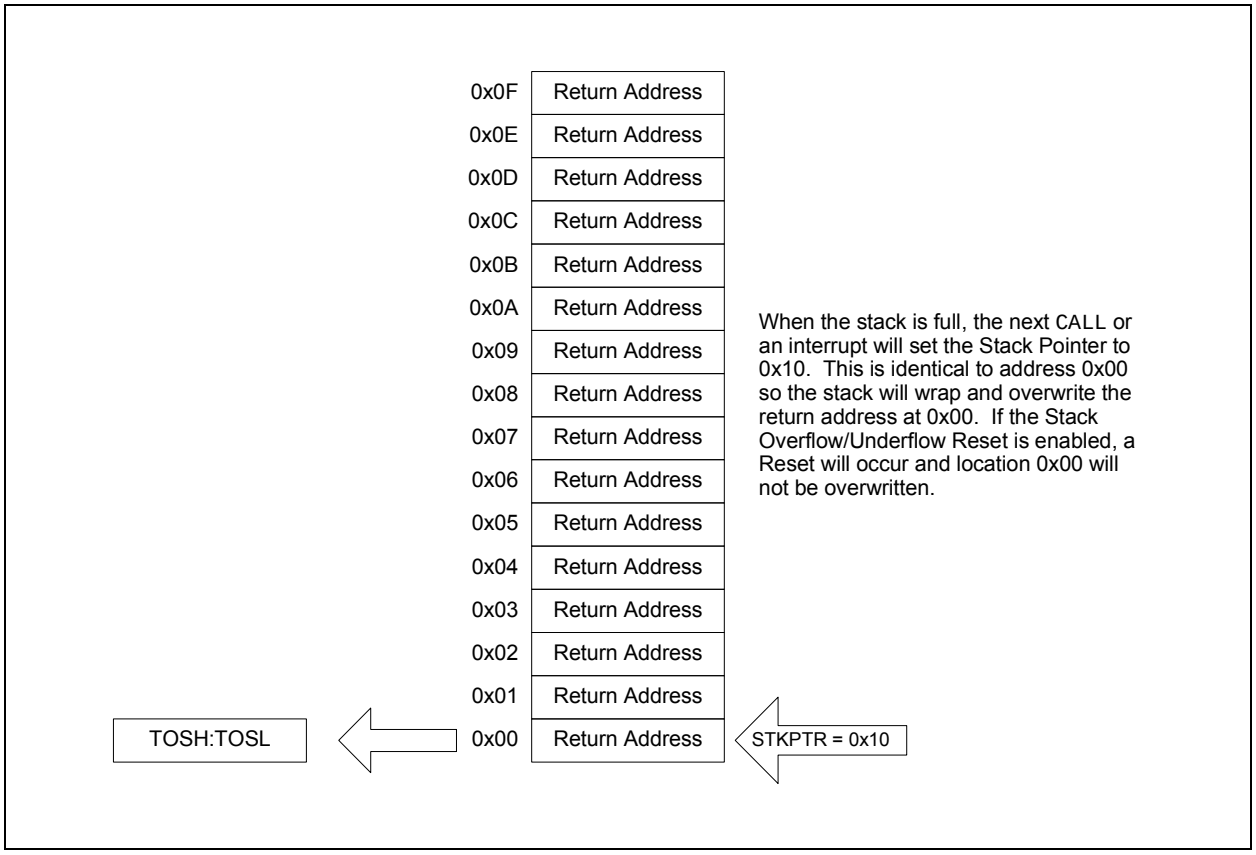


FIGURE 3-8: ACCESSING THE STACK EXAMPLE 4



3.4.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Word 2 is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

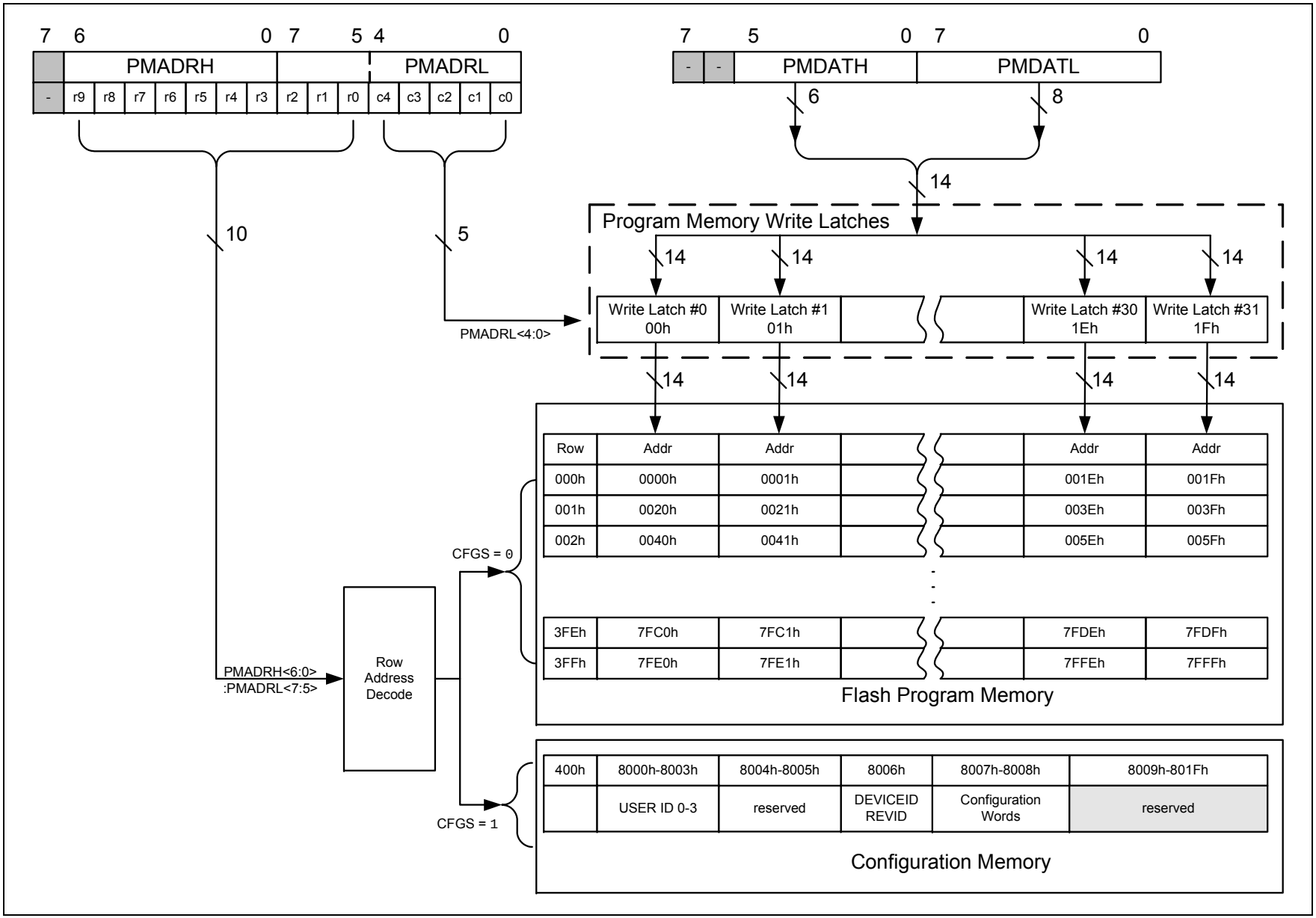
3.5 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

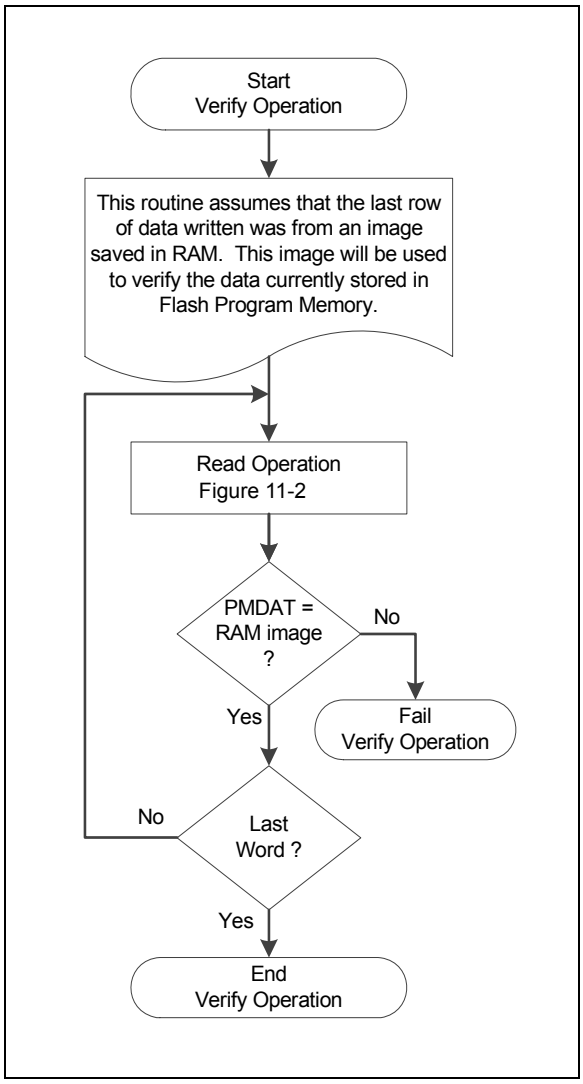
FIGURE 11-5: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 32 WRITE LATCHES



11.5 Write Verify

It is considered good programming practice to verify that program memory writes agree with the intended value. Since program memory is stored as a full page then the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

FIGURE 11-8: FLASH PROGRAM MEMORY VERIFY FLOWCHART



11.6 Flash Program Memory Control Registers

REGISTER 11-2: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
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 7-0 **PMDAT<7:0>**: Read/write value for Least Significant bits of program memory

REGISTER 11-3: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **PMDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 11-4: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PMADR<7:0>							
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 7-0 **PMADR<7:0>**: Specifies the Least Significant bits for program memory address

REGISTER 11-5: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	PMADR<14:8>						
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 **Unimplemented:** Read as '1'

bit 6-0 **PMADR<14:8>**: Specifies the Most Significant bits for program memory address

13.6 Interrupt-On-Change Registers

REGISTER 13-1: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0
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 7-0 **IOCBP<7:0>:** Interrupt-on-Change PORTB Positive Edge Enable bits

1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-2: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
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 7-0 **IOCBN<7:0>:** Interrupt-on-Change PORTB Negative Edge Enable bits

1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-3: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0
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	HS - Bit is set in hardware

bit 7-0 **IOCBF7:0>:** Interrupt-on-Change PORTB Flag bits

1 = An enabled change was detected on the associated pin.
Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.

0 = No change was detected, or the user cleared the detected change.

16.2.6 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 (refer to the TRIS register)
   - Configure pin as analog (refer to the ANSEL register)
   - Disable weak pull-ups either globally (refer to the OPTION_REG register) or individually (Refer to the appropriate WPUx register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - 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 in ADRES0H and ADRES0L.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

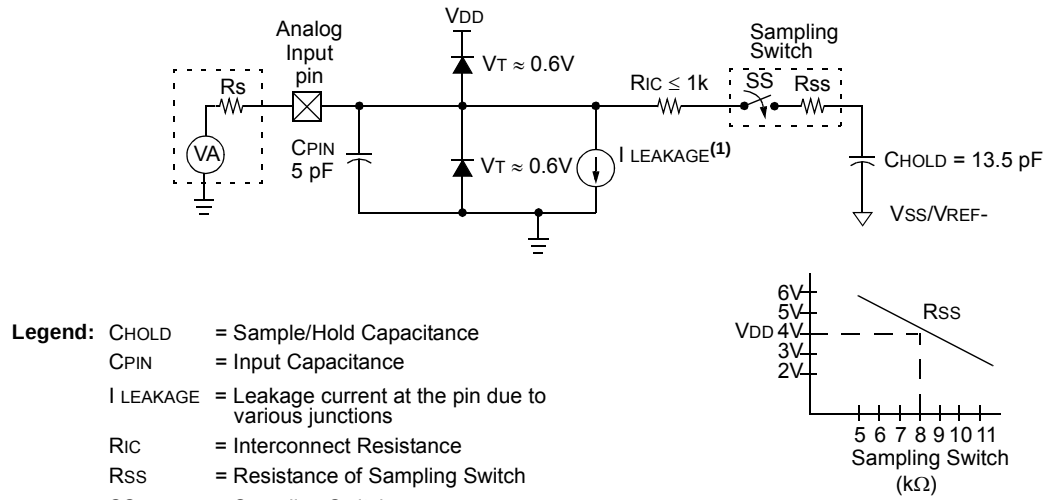
Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to **Section 16.4 "A/D Acquisition Requirements"**.

EXAMPLE 16-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'11110000'  ;Right justify, Frc
                       ;clock
MOVWF     ADCON1       ;Vdd and Vss Vref
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   WPUA         ;
BCF       WPUA, 0      ;Disable weak
                        pull-up on RA0
BANKSEL   ADCON0       ;
MOVLW     B'00000001'  ;Select channel AN0
MOVWF     ADCON0       ;Turn ADC On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,ADGO  ;Start conversion
BTFSC     ADCON0,ADGO  ;Is conversion done?
GOTO      $-1          ;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
```

FIGURE 16-4: ANALOG INPUT MODEL



Legend:

CHOLD	= Sample/Hold Capacitance
CPIN	= Input Capacitance
I LEAKAGE	= Leakage current at the pin due to various junctions
RIC	= Interconnect Resistance
RSS	= Resistance of Sampling Switch
SS	= Sampling Switch
VT	= Threshold Voltage

Note 1: Refer to **Section 25.0 "Electrical Specifications"**.

FIGURE 16-5: ADC TRANSFER FUNCTION

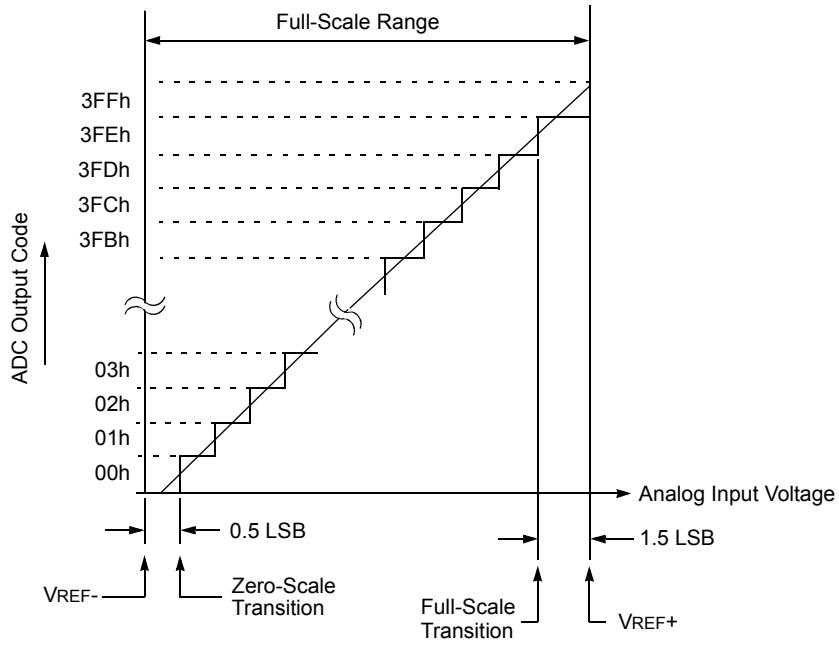
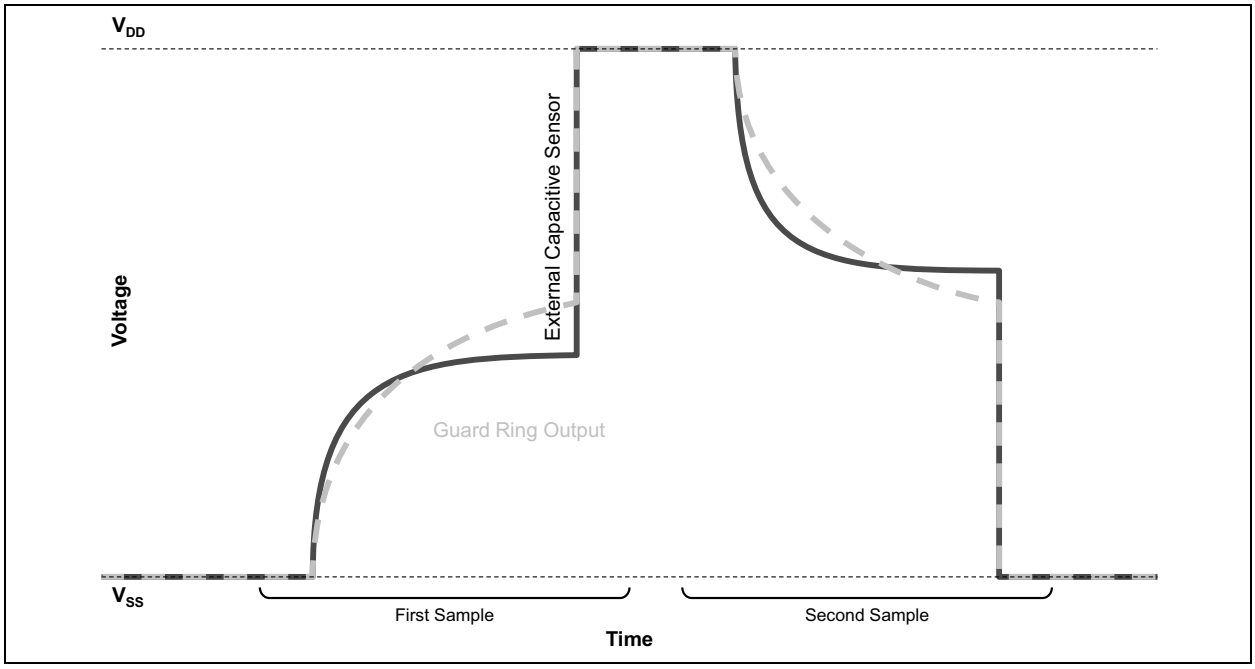


TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	130
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		131
ADRES0H	A/D Result Register High								132, 133
ADRES0L	A/D Result Register Low								132, 133
ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	104
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	108
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	111
CCP1CON	—	—	DC1B<1:0>		CCP1M<3:0>				236
CCP2CON	—	—	DC2B<1:0>		CCP2M<3:0>				236
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		120
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	69
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	70
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	72
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	103
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	107
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	110

Legend: — = unimplemented read as '0'. Shaded cells are not used for ADC module.

FIGURE 16-9: DIFFERENTIAL CVD WITH GUARD RING OUTPUT WAVEFORM



16.6.8 ADDITIONAL SAMPLE AND HOLD CAPACITOR

Additional capacitance can be added in parallel with the sample and hold capacitor (CHOLD) by setting the ADDCAP<2:0> bits of the AADCAP register. This bit connects a digitally programmable capacitance to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion. See Figure 16-6.

16.6.9 ANALOG BUS VISIBILITY

The ADOEN bit or the ADOLEN bit of the AADCON3 register can be used to connect the ADC conversion bus (CHOLD) to the ADOUT pin. This connection can be used to monitor the state and behavior of the internal analog bus and it also can be used to improve the match between internal and external capacitance by connecting a external capacitor to increase the effective internal capacitance. The ADOEN bit provides the connection via a standard channel passgate, while the ADOLEN bit enables a lower-impedance passgate.

The ADOUT pin function can be overridden during the pre-charge stage of conversion. This override function is controlled by the ADOOEN bit. The polarity of the override is set by the ADIPPOL bit. It should be noted that, outside of the pre-charge phase, no ADOUT override is in effect. Therefore, the user must manage the state of the ADOUT pin via the relevant TRIS bit in order to avoid unintended affects on conversion results. If the user wishes to have the ADOUT path be active during conversions, then the relevant TRIS bit should be set to ensure that the ADOUT pin logic is in the input mode during the acquisition phase of conversions.

16.6.10 HARDWARE CVD DOUBLE CONVERSION PROCEDURE

This is an example procedure for using hardware CVD to perform a double conversion for differential CVD measurement with active guard drive.

1. Configure Port:
   - Enable pin output driver (Refer to the TRIS register).
   - Configure pin output low (Refer to the LAT register).
   - Disable weak pull-up (Refer to the WPU register).
2. Configure the ADC module:
   - Select an appropriate ADC conversion clock for your oscillator frequency.
   - Configure voltage reference.
   - Select ADC input channel.
   - Turn on the ADC module.
3. Configure the hardware CVD module:
   - Configure charge polarity and double conversion.
   - Configure pre-charge and acquisition timer.
   - Configure guard ring (optional).
   - Select additional capacitance (optional).
4. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit or by enabling the Special Event Trigger in the ADDCON2 register.
6. Wait for the 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:
   - Conversion 1 result in ADDRES0H and ADDRES0L
   - Conversion 2 result in ADDRES1H and ADDRES1L
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.

EXAMPLE 16-2: HARDWARE CVD DOUBLE CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Fosc/16
;clock and AN0 input.
;
; The Hardware CVD will perform an inverted
; double conversion, Guard A and B drive are
; both enabled.
;Conversion start & polling for completion
are included.
;
BANKSEL    TRISA
BCF        TRISA,0      ;Set RA0 to output
BANKSEL    LATA
BCF        LATA,0       ;RA0 output low
BANKSEL    ANSELA
BCF        ANSELA,0     ;Set RA0 to digital
BANKSEL    WPUA
BCF        WPUA,0       ;Disable pull-up on
RA0
; Initialize ADC and Hardware CVD

BANKSEL    AADCON0
MOVLW      B'00000001’  ;Select channel AN0
MOVWF      AADCON0
BANKSEL    AADCON1
MOVLW      B'11010000'  ;Vdd and Vss Vref
MOVWF      AADCON1

BANKSEL    AADCON3
MOVLW      B'01000011'  ;Double and inverted
MOVWF      AADCON3      ;ADOUT disabled
BANKSEL    AADPRE
MOVLW      .10
MOVWF      AADPRE       ;Pre-charge Timer
BANKSEL    AADACQ
MOVLW      .10
MOVWF      AADACQ       ;Acquisition Timer
BANKSEL    AADGRD
MOVLW      B'11000000'  ;Guard on A and B
MOVWF      AADGRD
BANKSEL    AADCAP
MOVLW      B'00000000'
MOVWF      AADCAP       ;No additional
                        ;Capacitor
BANKSEL    ADCON0
BSF        ADCON0, GO
BTFSC      ADCON0, GO
GOTO       $-1          ;No, test again

;RESULTS OF CONVERIONS 1.
BANKSEL AADRES0H        ;
MOVF       AADRES0H,W   ;Read upper 2 bits
MOVWF      RESULT0H     ;store in GPR space
MOVF       AADRES0L,W   ;Read lower 8 bits
MOVWF      RESULT0L     ;Store in GPR space

;RESULTS OF CONVERIONS 2.
BANKSEL    AADRES1H     ;
MOVF       AADRES1H,W   ;Read upper 2 bits
MOVWF      RESULT1H     ;store in GPR space
MOVF       AADRES1L,W   ;Read lower 8 bits
MOVWF      RESULT1L     ;Store in GPR space
```

16.6.11 HARDWARE CVD REGISTER MAPPING

The hardware CVD module is an enhanced expansion of the standard ADC module as stated in **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** and is backward compatible with the other devices in this family. Control of the standard ADC module uses Bank 1 registers, see Table 16-4. This set of registers is mapped into Bank 14 with the control registers for the hardware CVD module. Although this subset of registers has different names, they are identical. Since the registers for the standard ADC are mapped into the Bank 14 address space, any changes to registers in Bank 1 will be reflected in Bank 14 and vice-versa.

TABLE 16-4: HARDWARE CVD REGISTER MAPPING

[Bank 14 Address]	[Bank 1 Address]
Hardware CVD	**ADC**
[711h] AADCON0(1)	[09Dh] ADCON0(1)
[712h] AADCON1(1)	[09Eh] ADCON1(1)
[713h] AADCON2	
[714h] AADCON3	
[715h] AADSTAT	
[716h] AADPRE	
[717h] AADACQ	
[718h] AADGRD	
[719h] AADCAP	
[71Ah] AADRES0L(1)	[09Bh] ADRES0L(1)
[71Bh] AADRES0H(1)	[09Ch] ADRES0H(1)
[71Ch] AADRES1L	
[71Dh] AADRES1H	

Note 1: Register is mapped in Bank 1 and Bank 14, using different names in each bank.

REGISTER 16-13: AADACQ: HARDWARE CVD ACQUISITION TIME CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	ADACQ<6:0>						
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 **Unimplemented:** Read as '0'

bit 6-0 **ADACQ<6:0>:** Acquisition/Charge Share Time Select bits(1)

111 1111 = Acquisition/charge share for 127 instruction cycles

111 1110 = Acquisition/charge share for 126 instruction cycles

•

•

•

000 0001 = Acquisition/charge share for one instruction cycle (FOSC/4)

000 0000 = ADC Acquisition/charge share time is disabled

Note 1: When the FRC clock is selected as the conversion clock source, it is also the clock used for the pre-charge and acquisition times.

REGISTER 16-14: AADGRD: HARDWARE CVD GUARD RING CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
GRDBOE(2)	GRDAOE(2)	GRDPOL(1,2)			—	—	—
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 **GRDBOE:** Guard Ring B Output Enable bit(2)

1 = ADC guard ring output is enabled to ADGRDB pin. Its corresponding TRISx bit must be clear.

0 = No ADC guard ring function to this pin is enabled

bit 6 **GRDAOE:** Guard Ring A Output Enable bit(2)

1 = ADC Guard Ring Output is enabled to ADGRDA pin. Its corresponding TRISx, x bit must be clear.

0 = No ADC Guard Ring function is enabled

bit 5 **GRDPOL:** Guard Ring Polarity selection bit(1,2)

1 = ADC guard ring outputs start as digital high during pre-charge stage

0 = ADC guard ring outputs start as digital low during pre-charge stage

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

Note 1: When the ADDSEN = 1 and ADIPEN = 1; the polarity of this output is inverted for the second conversion time. The stored bit value does not change.

2: Guard Ring outputs are maintained while ADON = 1. The ADGRDA output switches polarity at the start of the acquisition time.

17.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 17-1 is a block diagram of the Timer0 module.

17.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

17.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

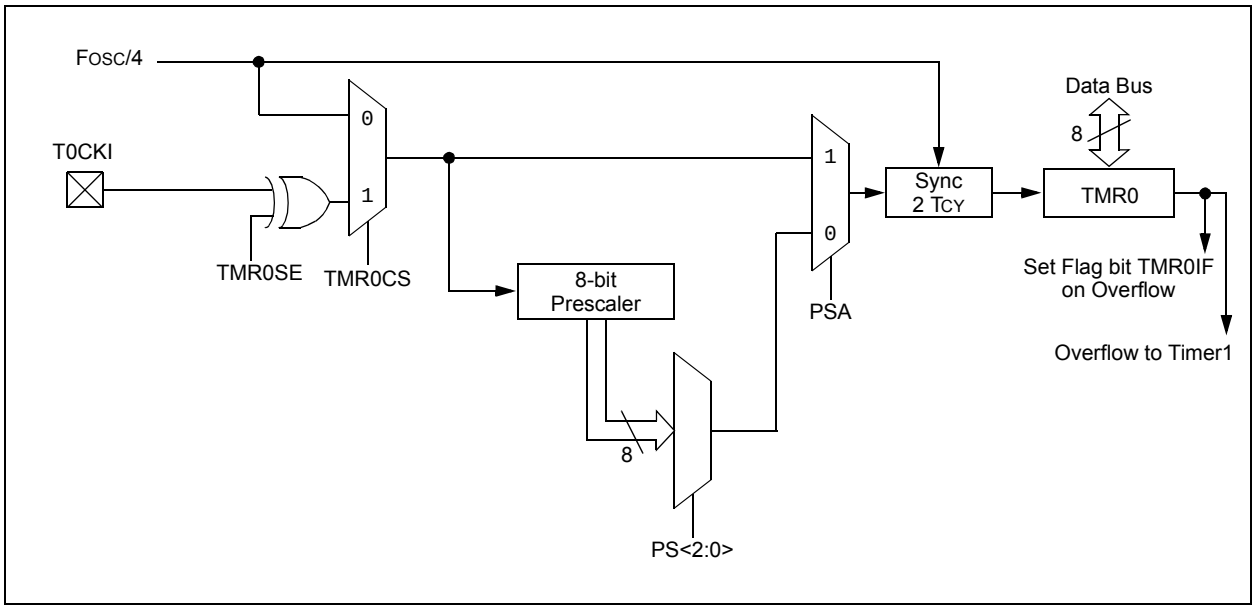
17.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMR0SE bit in the OPTION_REG register.

FIGURE 17-1: BLOCK DIAGRAM OF THE TIMER0



17.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

Note:	The Watchdog Timer (WDT) uses its own independent prescaler.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

17.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

17.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 25.0 "Electrical Specifications"**.

17.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

20.4 I²C MODE OPERATION

All MSSP I²C communication is byte-oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

20.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the 8th falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an Acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

20.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

20.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note: Data is tied to output zero when an I²C mode is enabled.

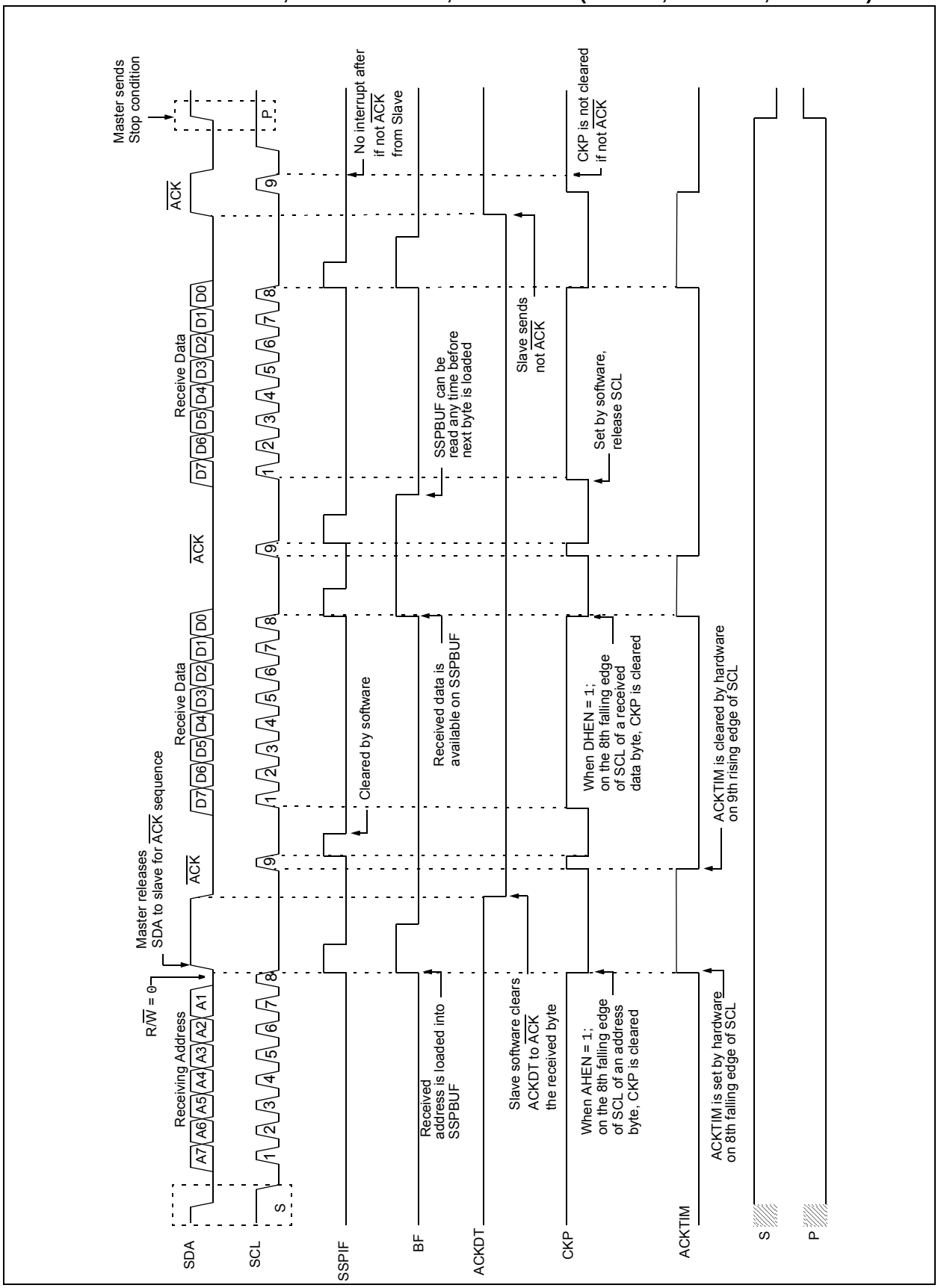
20.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 20-2: I²C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPADD.
Write Request	Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

FIGURE 20-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



20.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN bit of the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C Specification states that a bus collision cannot occur on a Start.

FIGURE 20-26: FIRST START BIT TIMING

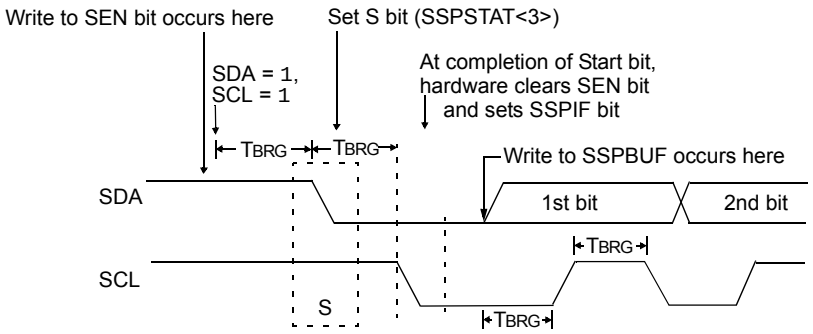
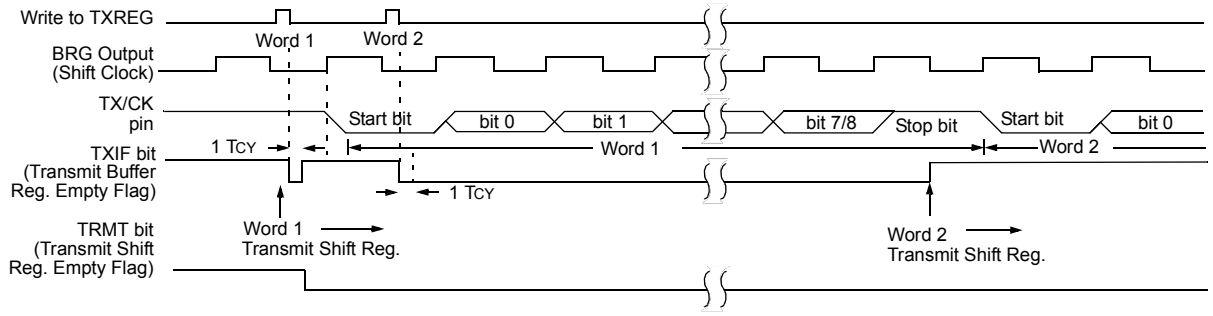


FIGURE 22-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 22-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	249
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	69
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	70
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	72
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	248
SPBRGL	BRG<7:0>								250*
SPBRGH	BRG<15:8>								250*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	110
TXREG	EUSART Transmit Data Register								239*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	247

Legend: — = unimplemented, read as '0'. Shaded cells are not used for asynchronous transmission.

* Page provides register information.

22.4.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 22-9 for the timing of the Break character sequence.

22.4.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

22.4.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 22.4.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCON register before placing the EUSART in Sleep mode.

FIGURE 22-9: SEND BREAK CHARACTER SEQUENCE