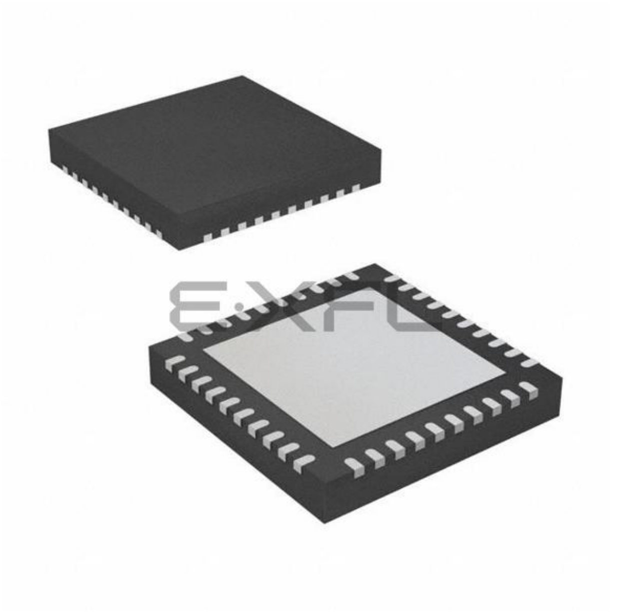
Microchip Technology - PIC16LF1567-I/MV Datasheet



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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 34x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1567-i-mv

2.0 ENHANCED MID-RANGE CPU

This family of devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative Addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

FIGURE 2-1: CORE BLOCK DIAGRAM

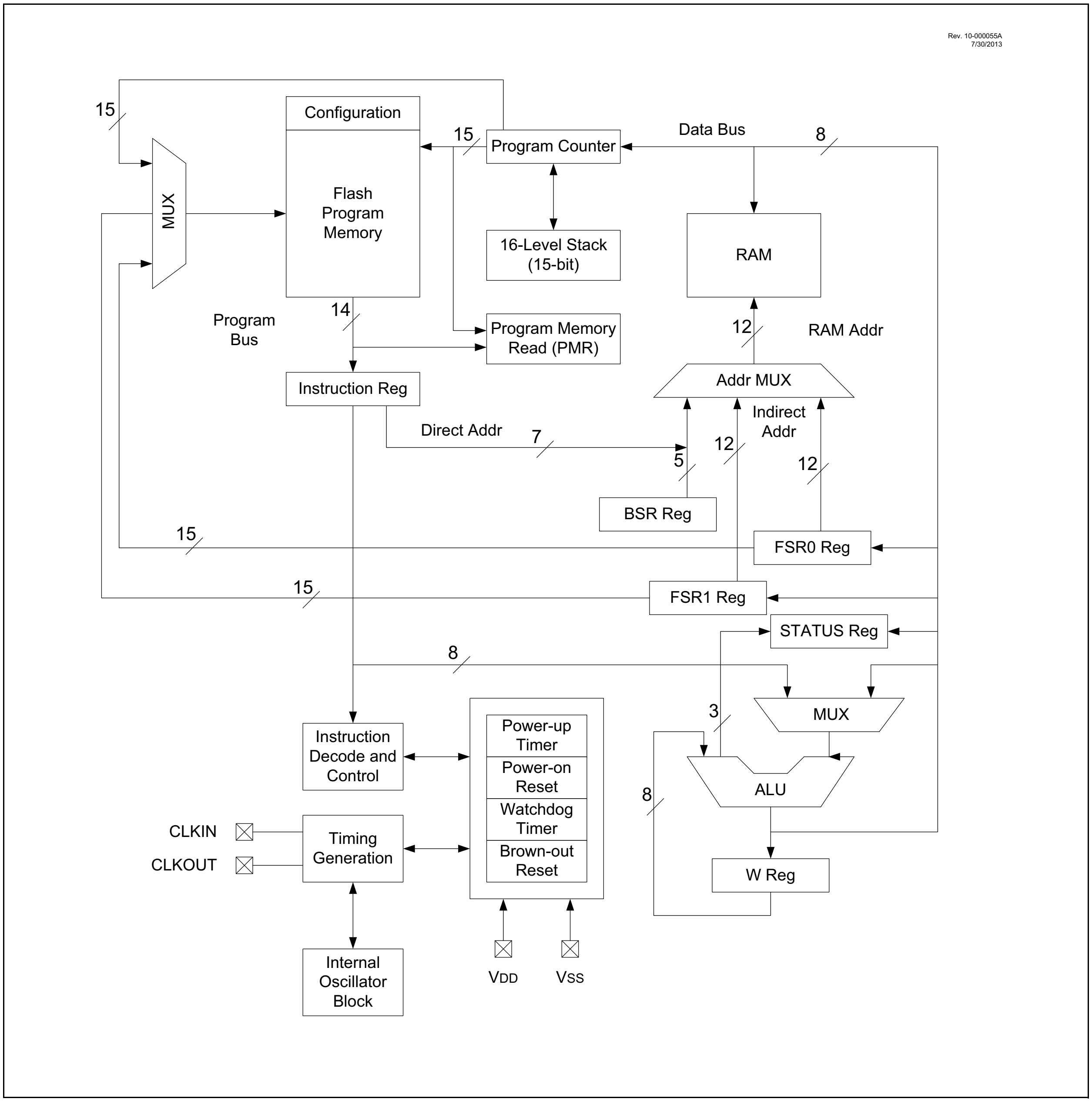


TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 7											
380h	INDF0[(1)]	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
381h	INDF1[(1)]	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
382h	PCL[(1)]	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
383h	STATUS[(1)]	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	---1 1000	---q quuu
384h	FSR0L[(1)]	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
385h	FSR0H[(1)]	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
386h	FSR1L[(1)]	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
387h	FSR1H[(1)]	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
388h	BSR[(1)]	—	—	—	BSR<4:0>					---0 0000	---0 0000
389h	WREG[(1)]	Working Register								0000 0000	uuuu uuuu
38Ah	PCLATH[(1)]	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
38Bh	INTCON[(1)]	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
38Ch	—	Unimplemented								—	—
38Dh	—	Unimplemented								—	—
38Eh	—	Unimplemented								—	—
38Fh	—	Unimplemented								—	—
390h	—	Unimplemented								—	—
391h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	0000 0000	0000 0000
392h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	0000 0000	0000 0000
393h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	0000 0000	0000 0000
394h	—	Unimplemented								—	—
395h	—	Unimplemented								—	—
396h	—	Unimplemented								—	—
397h	—	Unimplemented								—	—
398h	—	Unimplemented								—	—
399h	—	Unimplemented								—	—
39Ah	—	Unimplemented								—	—
39Bh	—	Unimplemented								—	—
39Ch	—	Unimplemented								—	—
39Dh	—	Unimplemented								—	—
39Eh	—	Unimplemented								—	—
39Fh	—	Unimplemented								—	—

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

Note 1: These registers can be accessed from any bank.

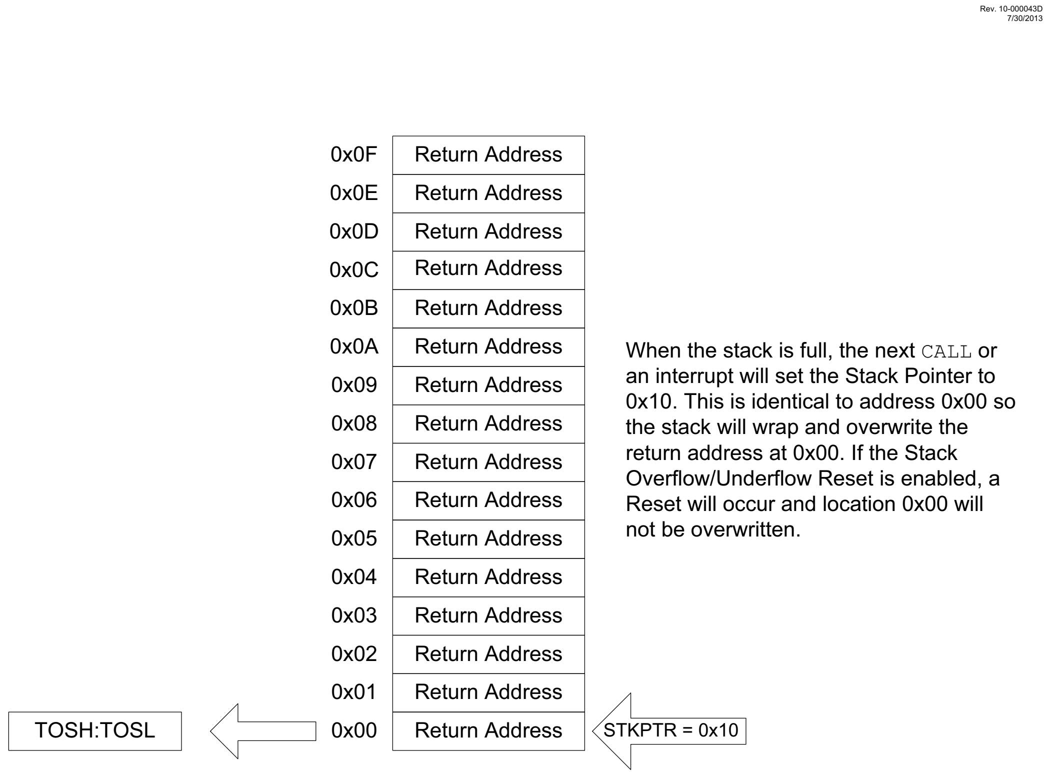
2: PIC16LF1567.

3: These registers/bits are available at two address locations, in Bank 1 and Bank 14.

4: PIC16LF1566 only.

5: Unimplemented, read as '1'.

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



When the stack is full, the next `CALL` or an interrupt will set the Stack Pointer to 0x10. This is identical to address 0x00 so the stack will wrap and overwrite the return address at 0x00. If the Stack Overflow/Underflow Reset is enabled, a Reset will occur and location 0x00 will not be overwritten.

3.4.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words 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.

5.2.2.3 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The outputs of the 16 MHz HFINTOSC postscaler and the LFINTOSC connect to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

> **Note:** Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to ‘0111’ and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

5.2.2.4 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-3). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. Clock switch is complete.

See Figure 5-3 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected.

Start-up delay specifications are located in the oscillator tables of **Section 25.0 “Electrical Specifications”**.

5.2.2.5 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Word 1 must be set to use the INTOSC source as the device system clock (FOSC<1:0> = 00).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<1:0> in Configuration Word 1 (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4x PLL.

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to ‘1x’. The SCS bits must be set to ‘00’ to use the 4x PLL with the internal oscillator.

9.6 Register Definitions: Watchdog Control

REGISTER 9-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
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 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits[1]

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)
•
•
•
10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)
10001 = 1:4194304 (2^{22}) (Interval 128s nominal)
10000 = 1:2097152 (2^{21}) (Interval 64s nominal)
01111 = 1:1048576 (2^{20}) (Interval 32s nominal)
01110 = 1:524288 (2^{19}) (Interval 16s nominal)
01101 = 1:262144 (2^{18}) (Interval 8s nominal)
01100 = 1:131072 (2^{17}) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.

REGISTER 10-3: 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 10-4: 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

14.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "*Use and Calibration of the Internal Temperature Indicator*" (DS01333) for more details regarding the calibration process.

14.1 Circuit Operation

Figure 14-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 14-1 describes the output characteristics of the temperature indicator.

EQUATION 14-1: VOUT RANGES

$$\text{High Range: } V_{OUT} = V_{DD} - 4V_T$$

$$\text{Low Range: } V_{OUT} = V_{DD} - 2V_T$$

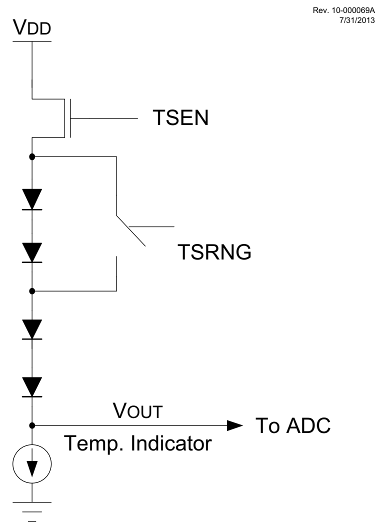
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher VDD is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 14-1: TEMPERATURE CIRCUIT DIAGRAM



14.2 Minimum Operating VDD

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, VDD, must be high enough to ensure that the temperature circuit is correctly biased.

Table 14-1 shows the recommended minimum VDD vs. range setting.

TABLE 14-1: RECOMMENDED VDD VS. RANGE

Min. VDD, TSRNG = 1	Min. VDD, TSRNG = 0
3.6V	1.8V

14.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 15.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

REGISTER 15-3: ADCON1[1]/ADCOMCON[2]: ADC CONTROL REGISTER 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	R/W-0/0
ADFM	ADCS<2:0>			ADNREF	GO/$\overline{\text{DONE}}$_ALL	ADPREF<1:0>	
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 **ADFM:** ADC Result Format Select bit
1 = Right justified. Six Most Significant bits of ADxRESxH are set to ‘0’ when the conversion result is loaded.
0 = Left justified. Six Least Significant bits of ADxRESxL are set to ‘0’ when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
111 = FRC (clock supplied from an internal RC oscillator)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock supplied from an internal RC oscillator)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

bit 3 **ADNREF:** ADC Negative Voltage Reference Configuration bit
1 = VREFL is connected to external VREF- pin[4]
0 = VREFL is connected to AVSS.

bit 2 **GO/$\overline{\text{DONE}}$_ALL[3]:** Synchronized ADC Conversion Status bit
1 = Synchronized ADC conversion in progress. Setting this bit starts conversion in any ADC with ADxON = 1.
0 = Synchronized ADC conversion completed/ not in progress.

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
11 = VREFH is connected to internal Fixed Voltage Reference.
10 = VREFH is connected to external VREF+ pin[4]
01 = Reserved
00 = VREFH is connected to VDD

Note 1: Bank 1 name is ADCON1.
2: Bank 14 name is ADCOMCON.
3: Setting this bit triggers the GO/$\overline{\text{DONE}}$x bits in both ADCs. Each ADC will run a conversion according to its control register settings. This bit reads as an OR of the individual GO/$\overline{\text{DONE}}$x bits.
4: When selecting the VREF+ or VREF- pin as the source of the positive or negative reference, be aware that a minimum voltage specification exists. See **Section 25.0 “Electrical Specifications”** for details.

REGISTER 16-2: AD2CON0: ANALOG-TO-DIGITAL (ADC) 2 CONTROL REGISTER 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	R/W-0/0
CHS25	CHS24	CHS23	CHS22	CHS21	CHS20	GO/$\overline{\text{DONE2}}$[(2)]	AD2ON
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 **CHS25<5:0>:** Analog Channel Select bits for ADC2
When AD2ON = 0, all multiplexer inputs are disconnected.
111111 = Fixed Voltage Reference (FVREF)
111101 = Temperature Indicator
111011 = VREFH (ADC Positive Reference)
101110 - 111010 = Reserved
101001 - 101101 = Channel 41 through 45, (AN41 through AN45)[(1)]
101000 = Channel 40, (AN40)
011110 - 100111 = Reserved
010100 - 011101 = Channel 20 through 29, (AN20 through AN29)
000011 - 010011 = Reserved
000010 = Channel 2, (AN2)
000001 = Channel 1, (AN1)
000000 = Channel 0, (AN0)

bit 1 **GO/$\overline{\text{DONE2}}$:** ADC2 Conversion Status bit[(2)]
If AD2ON = 1
1 = ADC conversion in progress. Setting this bit starts the ADC conversion. When the RC clock source is selected, the ADC Module waits one instruction before starting the conversion.
0 = ADC conversion not in progress (This bit is automatically cleared by hardware when the ADC conversion is complete.)
If this bit is cleared while a conversion is in progress, the conversion will stop and the results of the conversion up to this point will be transferred to the result registers, but the AD2IF interrupt flag bit will not be set.
If AD2ON = 0
0 = ADC conversion not in progress

bit 0 **AD2ON:** ADC Module 2 Enable bit
1 = ADC converter module 2 is operating
0 = ADC converter module 2 is shut off and consumes no operating current. All Analog channels are disconnected.

Note 1: PIC16LF1567 only. Not implemented on PIC16LF1566.
2: When the AD2DSEN bit is set; the GO/$\overline{\text{DONE}}$ bit will clear after a second conversion has completed.

REGISTER 16-19: AADxRESxH: HARDWARE CVD RESULT REGISTER MSB ADFM = 0(1)

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
ADRESx<9:2>							
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 7-0 **ADRESx<9:2>:** ADC Result Register bits
Upper eight bits of 10-bit conversion result

Note 1: See **Section 16.1.13 “Hardware CVD Register Mapping”** for more information.

REGISTER 16-20: AADxRESxL: HARDWARE CVD RESULT REGISTER LSL ADFM = 0(1)

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
ADRESx<1:0>		—	—	—	—	—	—
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 7-6 **ADRESx<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

Note 1: See **Section 16.1.13 “Hardware CVD Register Mapping”** for more information.

20.4.9 ACKNOWLEDGE SEQUENCE

The ninth SCLx pulse for any transferred byte in I^{2}C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDAx line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{\text{ACK}}$) is an active-low signal, pulling the SDAx line low indicated to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{\text{ACK}}$ is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{\text{ACK}}$ value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{\text{ACK}}$ response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an $\overline{\text{ACK}}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCLx on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

20.5 I^{2}C SLAVE MODE OPERATION

The MSSPx Slave mode operates in one of four modes selected in the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operated the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

20.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 20-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSPx Mask register (Register 20-5) affects the address matching process. See **Section 20.5.9 "SSPx Mask Register"** for more information.

20.5.1.1 I^{2}C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

20.5.1.2 I^{2}C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCLx is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCLx is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{\text{W}}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

FIGURE 20-16: I^2C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)

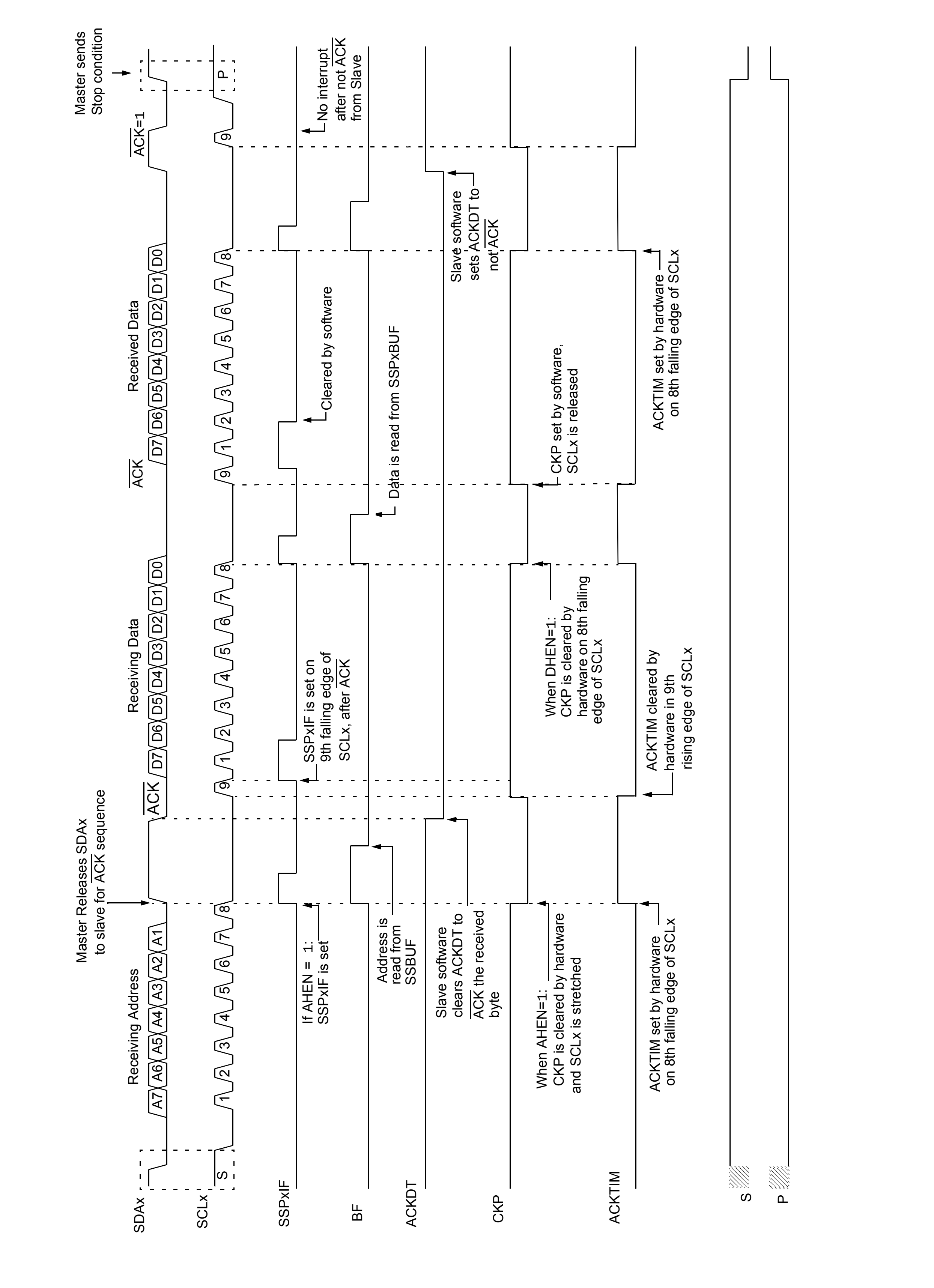
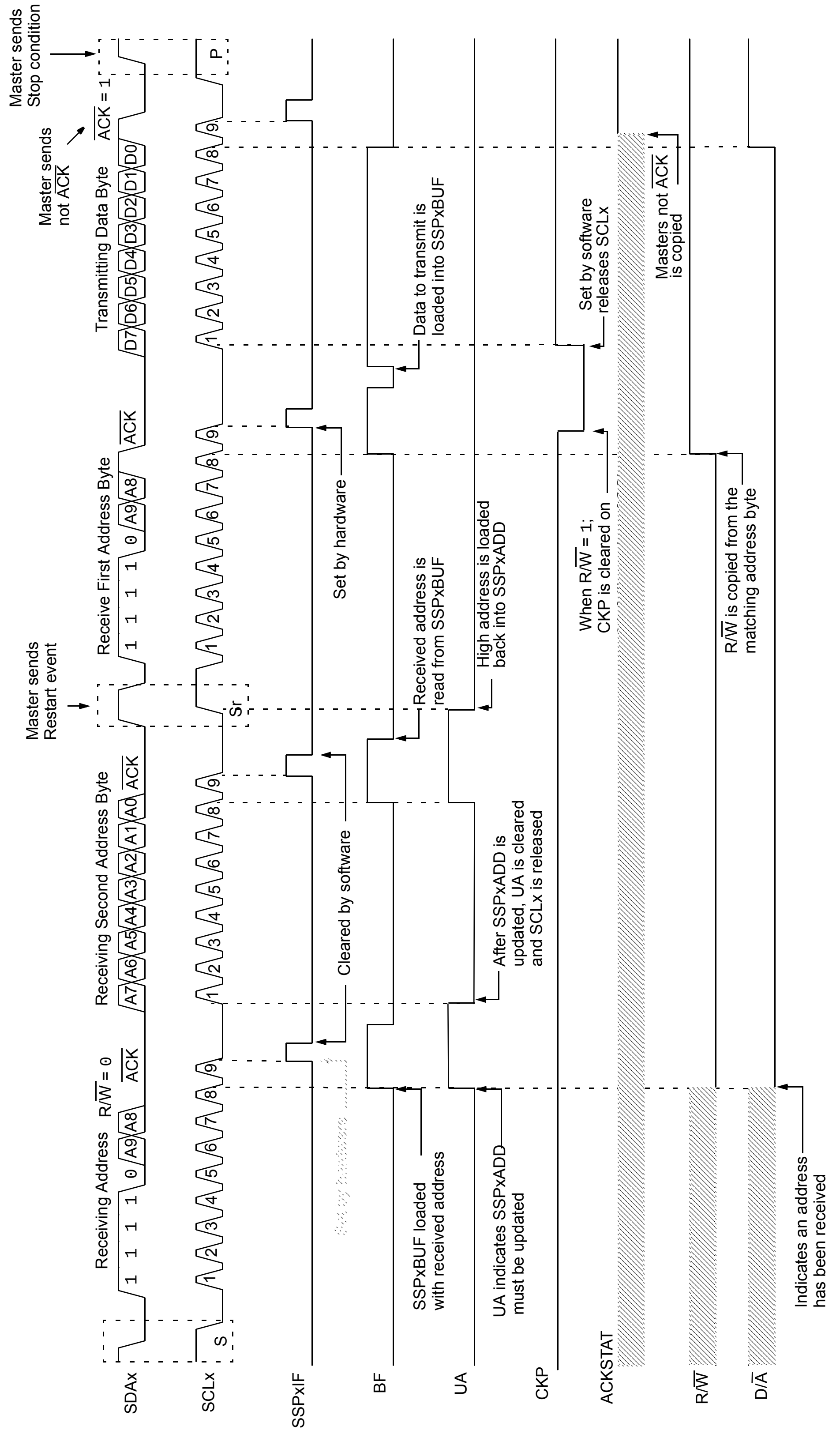

SDAx
SCLx
SSPxIF
BF
ACKDT
CKP
ACKTIM
S
P
Receiving Address
A7 A6 A5 A4 A3 A2 A1
S
1 2 3 4 5 6 7 8 9
Master Releases SDAx to slave for ACK sequence
ACK
Receiving Data
D7 D6 D5 D4 D3 D2 D1 D0
Received Data
ACK=1
P
Master sends Stop condition
If AHEN = 1: SSPxIF is set
SSPxIF is set on 9th falling edge of SCLx, after ACK
Cleared by software
No interrupt after not ACK from Slave
Address is read from SSBUF
Data is read from SSPxBUF
Slave software clears ACKDT to ACK the received byte
Slave software sets ACKDT to not ACK
When AHEN=1: CKP is cleared by hardware and SCLx is stretched
When DHEN=1: CKP is cleared by hardware on 8th falling edge of SCLx
CKP set by software, SCLx is released
ACKTIM set by hardware on 8th falling edge of SCLx
ACKTIM cleared by hardware in 9th rising edge of SCLx
ACKTIM set by hardware on 8th falling edge of SCLx

FIGURE 20-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)


SDAx
SCLx
SSPxIF
BF
UA
CKP
ACKSTAT
R/W̄
D/Ā
S
Receiving Address
1 1 1 1 0 A9 A8
R/W̄ = 0
ACK̄
1 2 3 4 5 6 7 8 9
Receiving Second Address Byte
A7 A6 A5 A4 A3 A2 A1 A0 ACK̄
Master sends Restart event
Sr
Receive First Address Byte
1 1 1 1 0 A9 A8
ACK̄
Transmitting Data Byte
D7 D6 D5 D4 D3 D2 D1 D0
Master sends not ACK̄
ACK̄ = 1
Master sends Stop condition
P
Set by hardware
Cleared by software
Set by hardware
SSPxBUF loaded with received address
Received address is read from SSPxBUF
Data to transmit is loaded into SSPxBUF
UA indicates SSPxADD must be updated
After SSPxADD is updated, UA is cleared and SCLx is released
High address is loaded back into SSPxADD
When R/W̄ = 1; CKP is cleared on
Set by software releases SCLx
Masters not ACK̄ is copied
R/W̄ is copied from the matching address byte
Indicates an address has been received

20.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCLx line low effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCLx.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. Setting CKP will release SCLx and allow more communication.

20.5.6.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$ if the R/$\overline{\text{W}}$ bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the $\overline{\text{ACK}}$ sequence. Once the slave is ready; CKP is set by software and communication resumes.

> **Note 1:** The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCLx.
>
> **2:** Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCLx. It is now always cleared for read requests.

20.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCLx is stretched without CKP being cleared. SCLx is released immediately after a write to SSPxADD.

> **Note:** Previous versions of the module did not stretch the clock if the second address byte did not match.

20.5.6.3 Byte NACKing

When the AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCLx for a received matching address byte. When the DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCLx for received data.

Stretching after the eighth falling edge of SCLx allows the slave to look at the received address or data and decide if it wants to ACK the received data.

20.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I^{2}C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I^{2}C bus have released SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 20-23).

FIGURE 20-23: CLOCK SYNCHRONIZATION TIMING

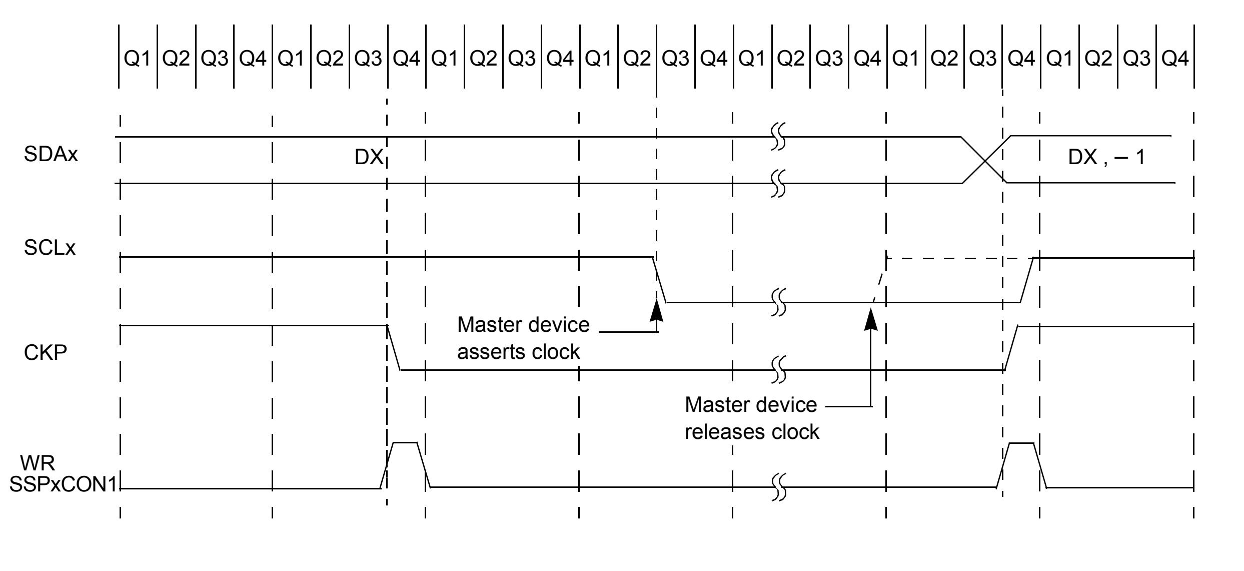

Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
SDAx
DX
DX , – 1
SCLx
CKP
Master device asserts clock
Master device releases clock
WR SSPxCON1

FIGURE 20-34: BUS COLLISION DURING START CONDITION (SCLX = 0)

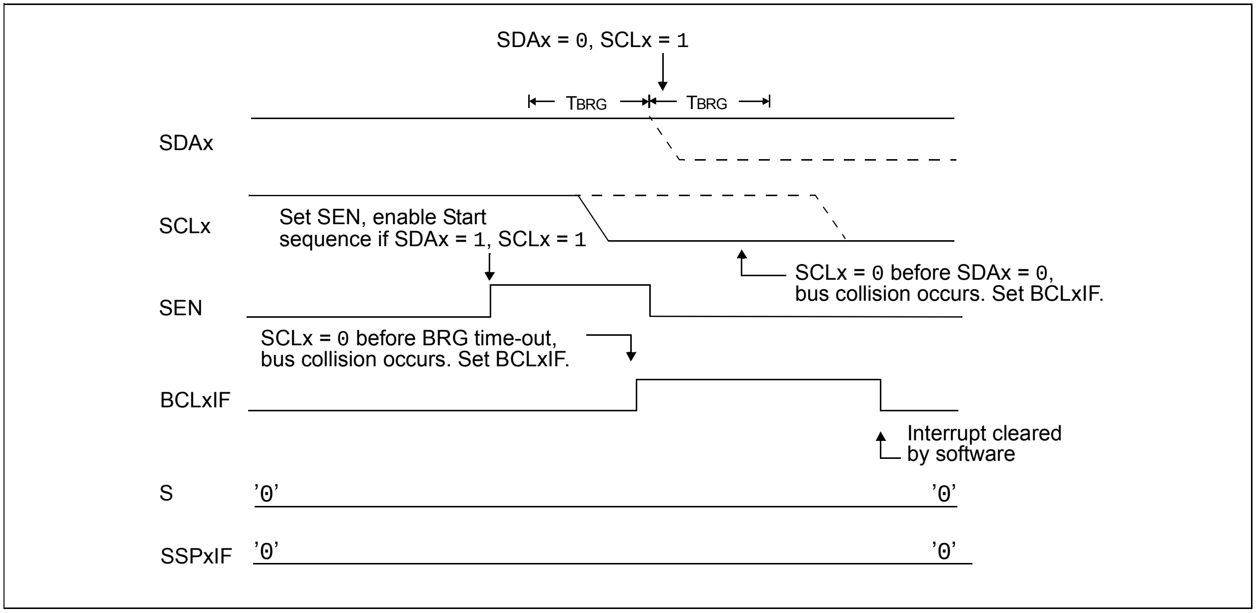

SDAx = 0, SCLx = 1
TBRG
TBRG
SDAx
SCLx
Set SEN, enable Start sequence if SDAx = 1, SCLx = 1
SCLx = 0 before SDAx = 0, bus collision occurs. Set BCLxIF.
SEN
SCLx = 0 before BRG time-out, bus collision occurs. Set BCLxIF.
BCLxIF
Interrupt cleared by software
S
'0'
'0'
SSPxIF
'0'
'0'

FIGURE 20-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION

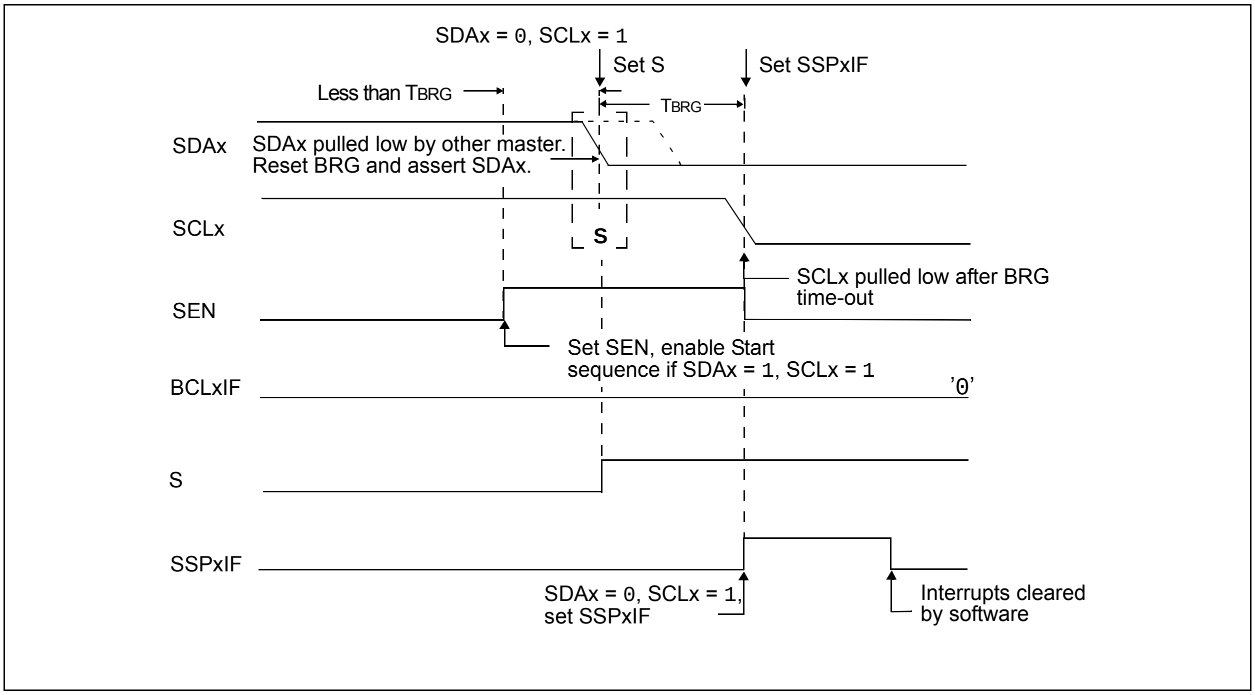

SDAx = 0, SCLx = 1
Set S
Set SSPxIF
Less than TBRG
TBRG
SDAx
SDAx pulled low by other master. Reset BRG and assert SDAx.
SCLx
S
SCLx pulled low after BRG time-out
SEN
Set SEN, enable Start sequence if SDAx = 1, SCLx = 1
BCLxIF
'0'
S
SSPxIF
SDAx = 0, SCLx = 1, set SSPxIF
Interrupts cleared by software

TABLE 20-3: SUMMARY OF REGISTERS ASSOCIATED WITH I^{2}C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	84
PIE1	TMR1GIE	ADIE	RC1IE	TX1IE	SSP1IE	SSP2IE	TMR2IE	TMR1IE	85
PIE2	—	AD2IE	—	—	BCL1IE	BCL2IE	TMR4IE	—	86
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	SSP2IF	TMR2IF	TMR1IF	87
PIR2	—	AD2IF	—	—	BCL1IF	BCL2IF	TMR4IF	—	88
SSP1ADD	ADD<7:0>								253
SSP2ADD	ADD<7:0>								253
SSP1BUF	MSSPx Receive Buffer/Transmit Register								201*
SSP2BUF	MSSPx Receive Buffer/Transmit Register								201*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				248
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				248
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	250
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	250
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	251
SSP2CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	251
SSP1MSK	MSK<7:0>								252
SSP2MSK	MSK<7:0>								252
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	246
SSP2STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	246
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	123
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	126

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by the MSSP module in I^{2}C mode.
* Page provides register information.

Note 1: PIC16F/LF1829 only.

TABLE 21-1: 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	266
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	84
PIE1	TMR1GIE	AD1IE	RCIE	TXIE	SSP1IE	SSP2IE	TMR2IE	TMR1IE	85
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	SSP2IF	TMR2IF	TMR1IF	87
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	265
SPBRGL	BRG<7:0>								267*
SPBRGH	BRG<15:8>								267*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	119
TXREG	EUSART Transmit Data Register								257
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	264

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for asynchronous transmission.
* Page provides register information.

21.4.4 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 21-9 for the timing of the Break character sequence.

21.4.4.1 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.

21.4.5 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 21.4.3 "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 21-9: SEND BREAK CHARACTER SEQUENCE

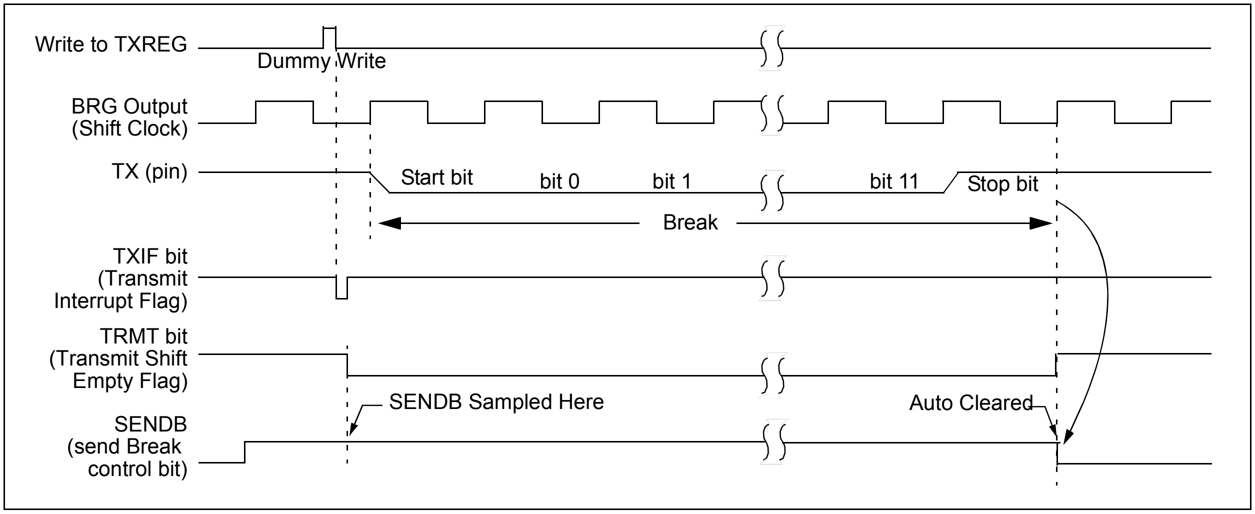
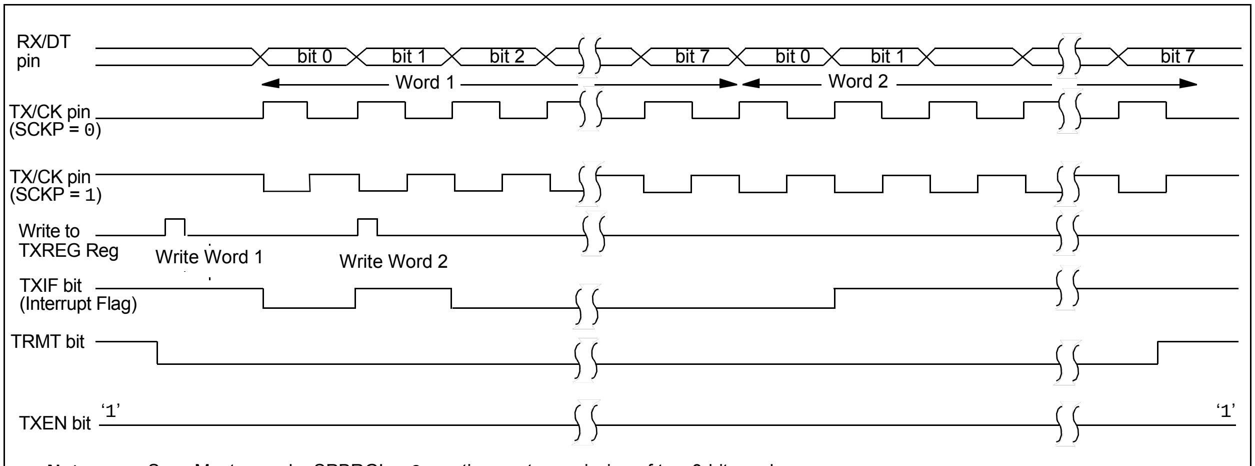


FIGURE 21-10: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPBRGL = 0, continuous transmission of two 8-bit words.

FIGURE 21-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

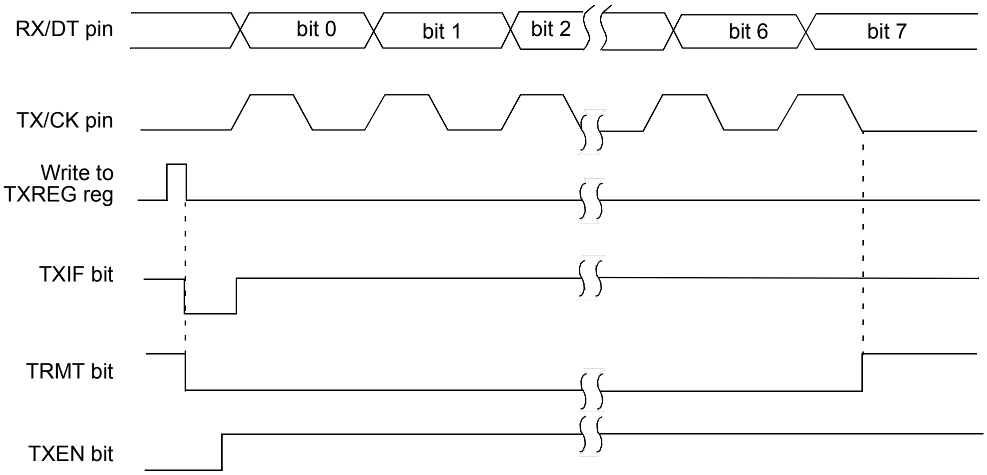


TABLE 21-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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	266
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	84
PIE1	TMR1GIE	AD1IE	RCIE	TXIE	SSP1IE	SSP2IE	TMR2IE	TMR1IE	85
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	SSP2IF	TMR2IF	TMR1IF	87
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	265
SPBRGL	BRG<7:0>								267*
SPBRGH	BRG<15:8>								267*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	119
TXREG	EUSART Transmit Data Register								257*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	264

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master transmission.
* Page provides register information.

TABLE 24-3: ENHANCED MID-RANGE INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
		CONTROL OPERATIONS							
BRA	k	Relative Branch	2	`11`	`001k`	`kkkk`	`kkkk`		
BRW	–	Relative Branch with W	2	`00`	`0000`	`0000`	`1011`		
CALL	k	Call Subroutine	2	`10`	`0kkk`	`kkkk`	`kkkk`		
CALLW	–	Call Subroutine with W	2	`00`	`0000`	`0000`	`1010`		
GOTO	k	Go to address	2	`10`	`1kkk`	`kkkk`	`kkkk`		
RETFIE	–	Return from interrupt	2	`00`	`0000`	`0000`	`1001`		
RETLW	k	Return with literal in W	2	`11`	`0100`	`kkkk`	`kkkk`		
RETURN	–	Return from Subroutine	2	`00`	`0000`	`0000`	`1000`		
		INHERENT OPERATIONS							
CLRWDT	–	Clear Watchdog Timer	1	`00`	`0000`	`0110`	`0100`	$\overline{TO}$, $\overline{PD}$	
NOP	–	No Operation	1	`00`	`0000`	`0000`	`0000`		
OPTION	–	Load OPTION_REG register with W	1	`00`	`0000`	`0110`	`0010`		
RESET	–	Software device Reset	1	`00`	`0000`	`0000`	`0001`		
SLEEP	–	Go into Standby mode	1	`00`	`0000`	`0110`	`0011`	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	`00`	`0000`	`0110`	`0fff`		
		C-COMPILER OPTIMIZED							
ADDFSR	n, k	Add Literal k to FSRn	1	`11`	`0001`	`0nkk`	`kkkk`		

MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	`00`	`0000`	`0001`	`0nmm`
							`kkkk`
								Z	2, 3

	k[n]	Move INDFn to W, Indexed Indirect.	1	`11`	`1111`	`0nkk`	`1nmm`	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	`00`	`0000`	`0001`	`kkkk`		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	`11`	`1111`	`1nkk`			2

Note 1: 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`.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See Table in the MOVIW and MOVWI instruction descriptions.