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	26
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18455-e-ss

8. Resets

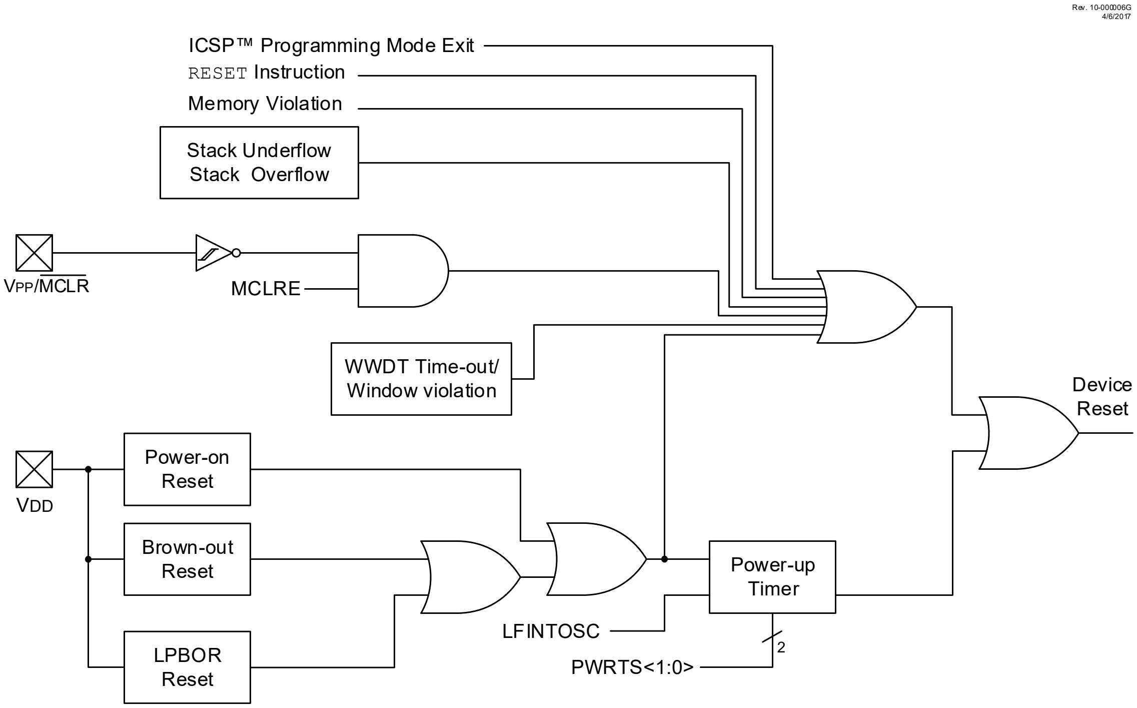
There are multiple ways to reset this device:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- Low-Power Brown-out Reset (LPBOR)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- `RESET` instruction
- Stack Overflow
- Stack Underflow
- Programming mode exit

To allow V_{DD} to stabilize, an optional Power-up Timer can be enabled to extend the Reset time after a BOR or POR event.

A simplified block diagram of the On-Chip Reset Circuit is shown in the block diagram below.

Figure 8-1. Simplified Block Diagram of On-Chip Reset Circuit



Note: See *"BOR Operating Conditions"* table for BOR active conditions.

Related Links

8.2.3 BOR Controlled by Software

14. I/O Ports

14.1 PORT Availability

Table 14-1. PORT Availability Per Device

PORTs	PORT Description	PIC16(L)F18455	PIC16(L)F18456
PORTA	8-bit wide, bidirectional port.	•	•
PORTB	8-bit wide, bidirectional port.	•	•
PORTC	8-bit wide, bidirectional port.	•	•
PORTE	1-bit wide, available only when Master Clear functionality is disabled (MCLRE = 0).	•	•

14.2 I/O Ports Description

Each port has standard registers for its operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate control)
- ODCONx registers (open-drain control)

Most port pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSELx bit is set, the digital input buffer associated with that bit is disabled.

14.7.16 WPUC

Name: WPUC
Address: 0x1F4F

Weak Pull-up Register

Bit	7	6	5	4	3	2	1	0
	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – WPUCn Weak Pull-up PORTC Control bits

Value	Description
1	Weak Pull-up enabled
0	Weak Pull-up disabled

15.9.2 Pin Rxy Output Source Selection Register

Name: RxyPPS

Important: See 15.8 Register Summary - PPS for the address offset of each individual register.

Bit	7	6	5	4	3	2	1	0
			RxyPPS[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – RxyPPS[5:0] Pin Rxy Output Source Selection bits
See output source selection table for source codes.

17.6.6 IOCBN

Name: IOCBN
Address: 0x1F49

Interrupt-on-Change Negative Edge Register Example

Bit	7	6	5	4	3	2	1	0
	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – IOCBNn Interrupt-on-Change Negative Edge Enable bits

Value	Description
1	Interrupt-on-Change enabled on the IOCA pin for a negative-going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0	Interrupt-on-Change disabled for the associated pin.

17.6.11 IOCCP

Name: IOCCP
Address: 0x1F53

Interrupt-on-Change Positive Edge Register

Bit	7	6	5	4	3	2	1	0
	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – IOCCPn Interrupt-on-Change Positive Edge Enable bits

Value	Description
1	Interrupt-on-Change enabled on the IOCC 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.

22. Numerically Controlled Oscillator (NCO) Module

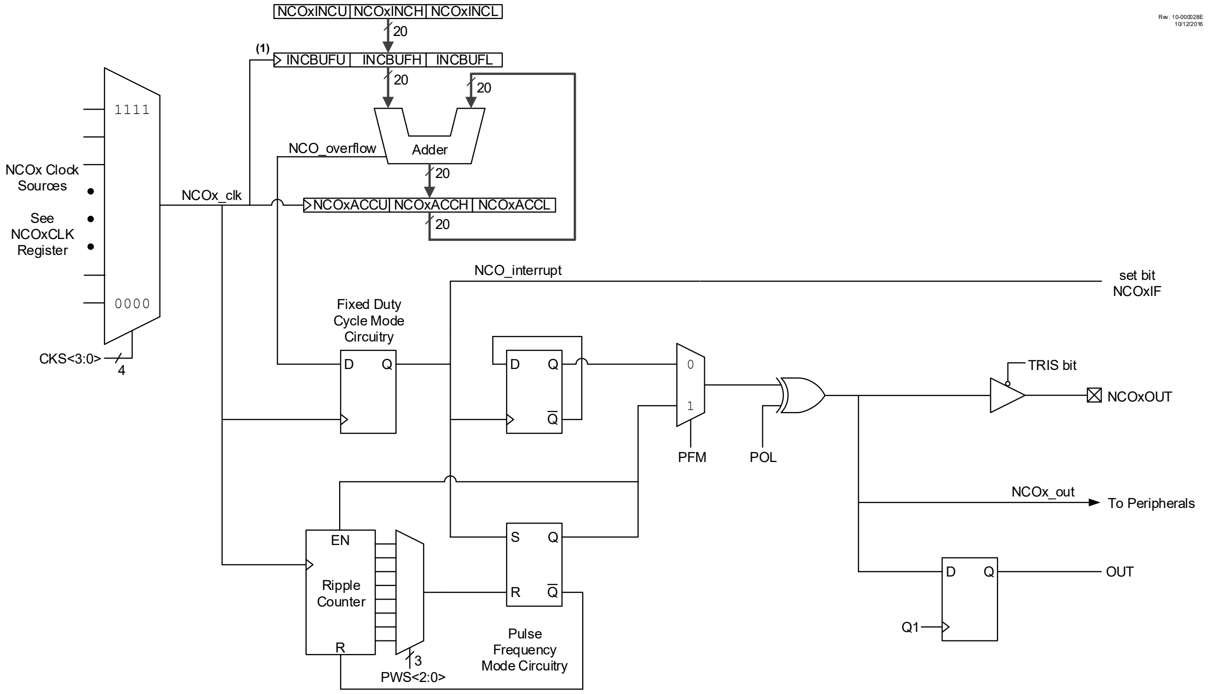
The Numerically Controlled Oscillator (NCO) module is a timer that uses overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the output frequency resolution does not vary with the divider value. The NCO is most useful for application that requires frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO include:

- 20-bit Increment Function
- Fixed Duty Cycle mode (FDC) mode
- Pulse Frequency (PF) mode
- Output Pulse Width Control
- Multiple Clock Input Sources
- Output polarity Control
- Interrupt Capability

The following figure is a simplified block diagram of the NCO module.

Figure 22-1. Numerically Controlled Oscillator Module Simplified Block Diagram



Note:

1. The increment registers are double-buffered to allow for value changes to be made without first disabling the NCO module. The full increment value is loaded into the buffer registers on the second rising edge of the NCOx_clk signal that occurs immediately after a write to NCOxINCL register. The buffers are not user-accessible and are shown here for reference.

23.15.2 CMxCON1

Name: CMxCON1
Address: 0x991,0x995

Comparator x Control Register 1

Bit	7	6	5	4	3	2	1	0
							INTP	INTN
Access							R/W	R/W
Reset							0	0

Bit 1 – INTP Comparator Interrupt on Positive-Going Edge Enable bit

Value	Description
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 0 – INTN Comparator Interrupt on Negative-Going Edge Enable bit

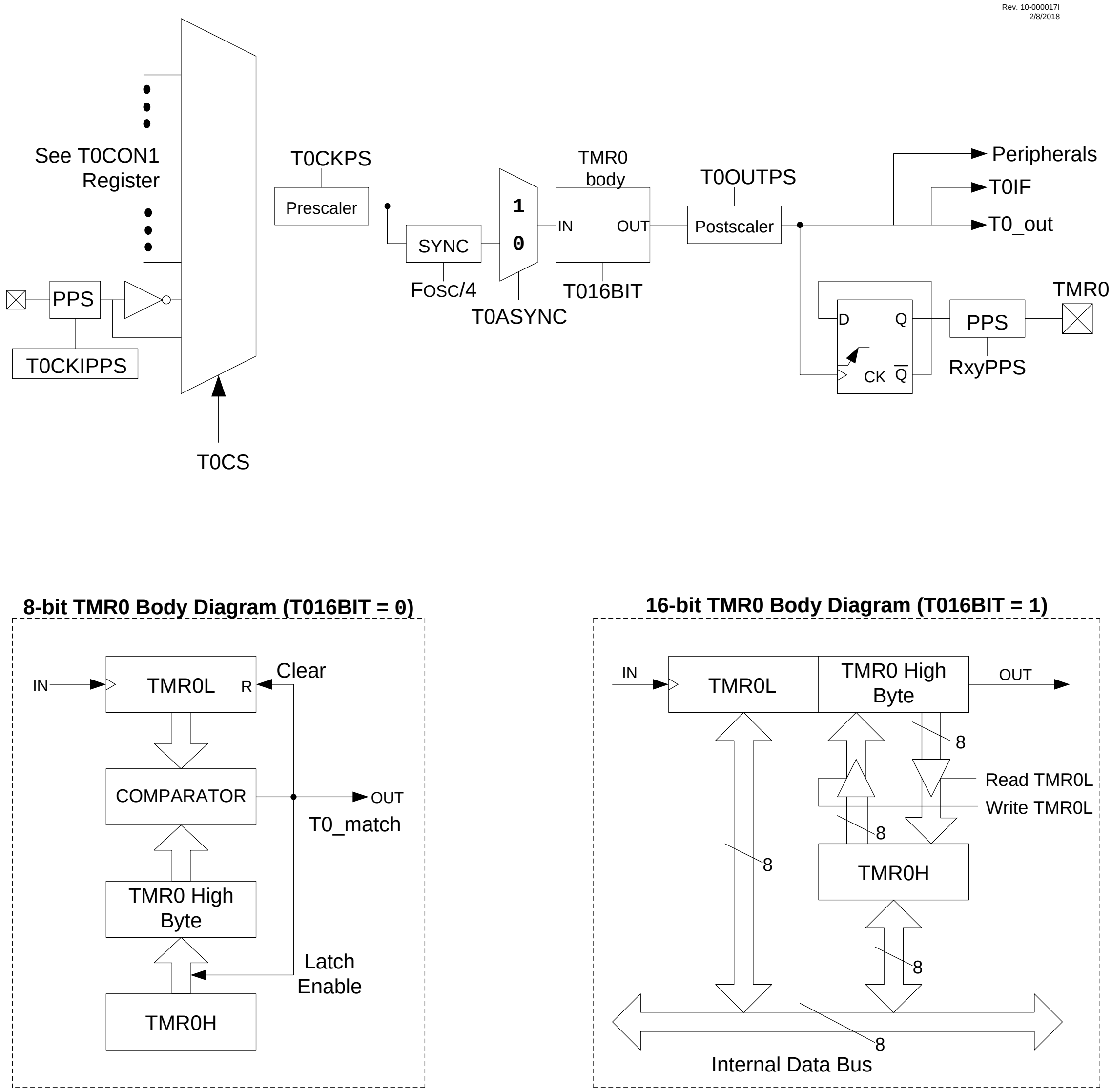
Value	Description
1	The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit
0	No interrupt flag will be set on a negative-going edge of the CxOUT bit

25. Timer0 Module

Timer0 module has the following features:

- 8-Bit B\Timer with Programmable Period
- 16-Bit Timer
- Selectable Clock Sources
- Synchronous and Asynchronous Operation
- Programmable Prescaler and Postscaler
- Interrupt on Match or Overflow
- Output on I/O Pin (via PPS) or to Other Peripherals
- Operation During Sleep

Figure 25-1. Timer0 Block Diagram



33.8.5 CLCxSEL2

Name: CLCxSEL2
Address: 0x1E14,0x1E1E,0x1E28,0x1E32

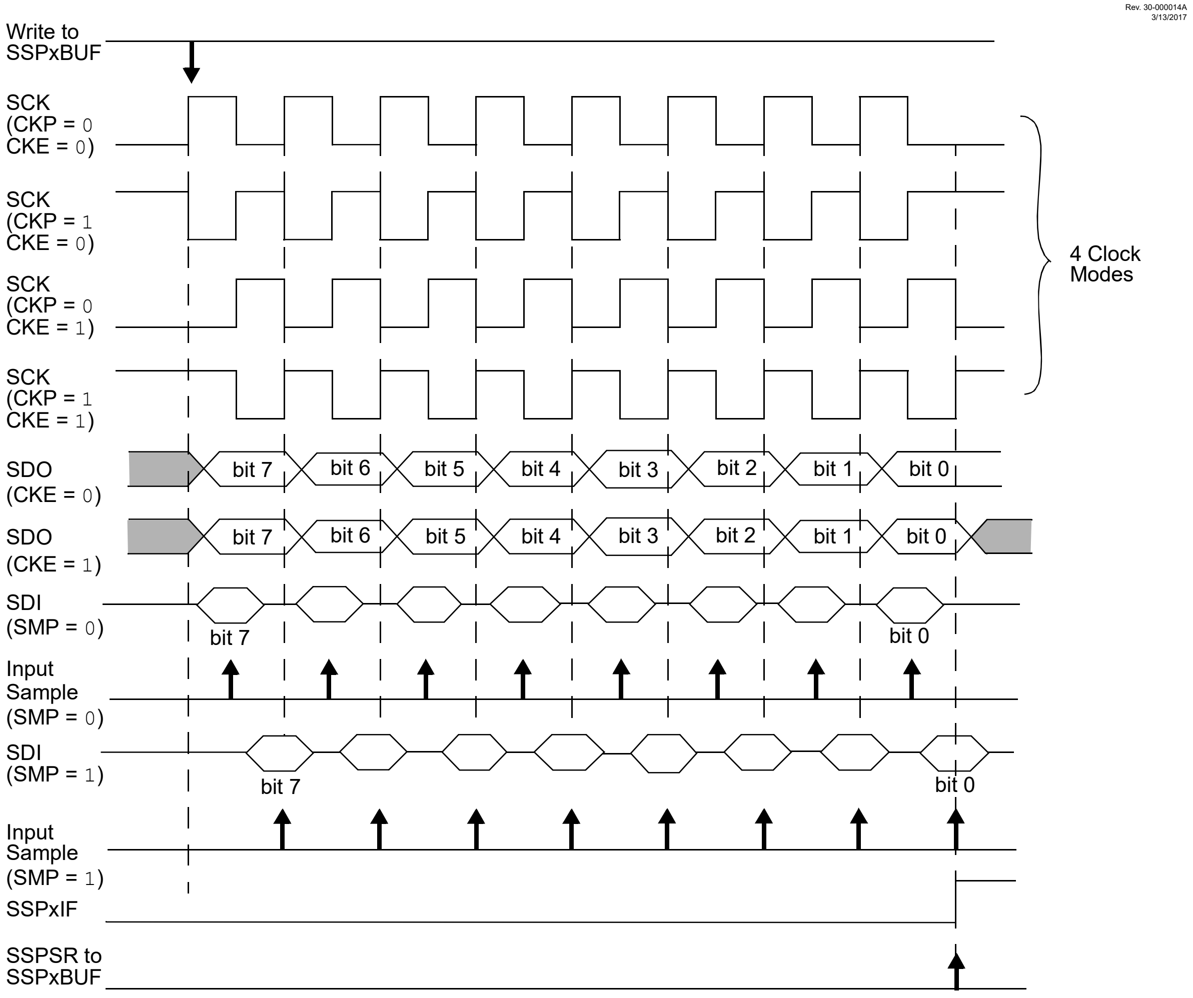
Generic CLCx Data 1 Select Register

Bit	7	6	5	4	3	2	1	0
			D3S[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bits 5:0 – D3S[5:0]
CLCx Data3 Input Selection bits
Reset States: POR/BOR = xxxxxx
All Other Resets = uuuuuu

Value	Description
n	Refer to CLC Input Sources for input selections

Figure 35-4. SPI Mode Waveform (Master Mode)



35.2.2 SPI Slave Mode

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

35.2.3 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 35-16 displays a module using both address and data holding. Figure 35-17 includes the operation with the SEN bit of the SSPxCON2 register set.

1. S bit is set; SSPxIF is set if interrupt on Start detect is enabled.
2. Matching address with R/W bit clear is clocked in. SSPxIF is set and CKP cleared after the eighth falling edge of SCL.
3. Slave clears the SSPxIF.
4. Slave can look at the ACKTIM bit to determine if the SSPxIF was after or before the $\overline{\text{ACK}}$.
5. Slave reads the address value from SSPxBUF, clearing the BF flag.
6. Slave sets $\overline{\text{ACK}}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPxIF is set after an $\overline{\text{ACK}}$, not after a NACK.
9. If SEN = `1`, the slave hardware will stretch the clock after the $\overline{\text{ACK}}$.
10. Slave clears SSPxIF.

> **Important:** SSPxIF is still set after the ninth falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPxIF not set

11. SSPxIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{\text{ACK}}$ = `1`, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit.

the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

35.6.6.1 BF Status Flag

In Transmit mode, the BF bit is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

35.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

The WCOL bit must be cleared by software before the next transmission.

35.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit is cleared when the slave has sent an Acknowledge ($\overline{\text{ACK}}$ = 0) and is set when the slave does not Acknowledge ($\overline{\text{ACK}}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

35.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSP module shifts in the $\overline{\text{ACK}}$ bit from the slave device and writes its value into the ACKSTAT bit.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the $\overline{\text{ACK}}$ bit from the slave device and writes its value into the ACKSTAT bit.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits. Interrupt is generated once the Stop/Restart condition is complete.

35.9.2 SSPxCON1

Name: SSPxCON1
Address: 0x190,0x19A

MSSP Control Register 1

Bit	7	6	5	4	3	2	1	0
	WCOL	SSPOV	SSPEN	CKP	SSPM[3:0]			
Access	R/W/HS	R/W/HS	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – WCOL
Write Collision Detect bit

Value	Mode	Description
1	SPI	A write to the SSPxBUF register was attempted while the previous byte was still transmitting (must be cleared by software)
1	I^{2}C Master transmit	A write to the SSPxBUF register was attempted while the I^{2}C conditions were not valid for a transmission to be started (must be cleared by software)
1	I^{2}C Slave transmit	The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0	SPI or I^{2}C Master or Slave transmit	No collision
x	Master or Slave receive	Don't care

Bit 6 – SSPOV
Receive Overflow Indicator bit[1]

Value	Mode	Description
1	SPI Slave	A byte is received while the SSPxBUF register is still holding the previous byte. The user must read SSPxBUF, even if only transmitting data, to avoid setting overflow. (must be cleared in software)
1	I^{2}C Receive	A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)
0	SPI Slave or I^{2}C Receive	No overflow
x	SPI Master or I^{2}C Master transmit	Don't care

Bit 5 – SSPEN
Master Synchronous Serial Port Enable bit.[2]

Value	Mode	Description
1	SPI	Enables the serial port. The SCKx, SDOx, SDIx, and $\overline{\text{SSx}}$ pin selections must be made with the PPS controls. Each signal must be configured with the corresponding TRIS control to the direction appropriate for the mode selected.
1	I^{2}C	Enables the serial port. The SDAx and SCLx pin selections must be made with the PPS controls. Since both signals are bidirectional the PPS input pin and PPS output pin selections must be made that specify the same pin. Both pins must be configured as inputs with the corresponding TRIS controls.
0	All	Disables serial port and configures these pins as I/O port pins

Bit 4 – CKP
SCK Release Control bit

Value	Mode	Description
1	SPI	Idle state for the clock is a high level
0	SPI	Idle state for the clock is a low level
1	I^{2}C Slave	Releases clock
0	I^{2}C Slave	Holds clock low (clock stretch), used to ensure data setup time
x	I^{2}C Master	Unused in this mode

Bits 3:0 – SSPM[3:0]
Master Synchronous Serial Port Mode Select bits[4]

Value	Description
1111	I^{2}C Slave mode: 10-bit address with Start and Stop bit interrupts enabled
1110	I^{2}C Slave mode: 7-bit address with Start and Stop bit interrupts enabled
1101	Reserved - do not use
1100	Reserved - do not use
1011	I^{2}C Firmware Controlled Master mode (slave Idle)
1010	SPI Master mode: Clock = F_{OSC}/(4*(SSPxADD+1)). SSPxADD must be greater than 0.[3]
1001	Reserved - do not use
1000	I^{2}C Master mode: Clock = F_{OSC}/(4 * (SSPxADD + 1))
0111	I^{2}C Slave mode: 10-bit address
0110	I^{2}C Slave mode: 7-bit address
0101	SPI Slave mode: Clock = SCKx pin. $\overline{\text{SSx}}$ pin control is disabled
0100	SPI Slave mode: Clock = SCKx pin. $\overline{\text{SSx}}$ pin control is enabled
0011	SPI Master mode: Clock = TMR2 output/2
0010	SPI Master mode: Clock = Fosc/64
0001	SPI Master mode: Clock = Fosc/16
0000	SPI Master mode: Clock = Fosc/4

Note:

1. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
2. When enabled, these pins must be properly configured as inputs or outputs.
3. SSPxADD = 0 is not supported.
4. Bit combinations not specifically listed here are either reserved or implemented in I^{2}C mode only.

36.6.1 RCxSTA

Name: RCxSTA
Address: 0x11D,0xA1D

Receive Status and Control Register

Bit	7	6	5	4	3	2	1	0
	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
Access	R/W	R/W	R/W	R/W	R/W	RO	R/HC	R/HC
Reset	0	0	0	0	0	0	0	0

Bit 7 – SPEN Serial Port Enable bit

Value	Description
1	Serial port enabled
0	Serial port disabled (held in Reset)

Bit 6 – RX9 9-Bit Receive Enable bit

Value	Description
1	Selects 9-bit reception
0	Selects 8-bit reception

Bit 5 – SREN Single Receive Enable bit
Controls reception. This bit is cleared by hardware when reception is complete

Value	Condition	Description
1	SYNC = 1 AND CSRC = 1	Start single receive
0	SYNC = 1 AND CSRC = 1	Single receive is complete
X	SYNC = 0 OR CSRC = 0	Don't care

Bit 4 – CREN Continuous Receive Enable bit

Value	Condition	Description
1	SYNC = 1	Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0	SYNC = 1	Disables continuous receive
1	SYNC = 0	Enables receiver
0	SYNC = 0	Disables receiver

Bit 3 – ADDEN Address Detect Enable bit

Value	Condition	Description
1	SYNC = 0 AND RX9 = 1	The receive buffer is loaded and the interrupt occurs only when the ninth received bit is set
0	SYNC = 0 AND RX9 = 1	All bytes are received and interrupt always occurs. Ninth bit can be used as parity bit
X	RX9 = 0 OR SYNC = 1	Don't care

37.3.7 SMTxTMR

Name: SMTxTMR
Address: 0x48C,0x50C

SMT Timer Register

Bit	23	22	21	20	19	18	17	16
	TMRU[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TMRH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TMRL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – TMRU[7:0] Upper byte of the SMT timer register

Bits 15:8 – TMRH[7:0] High byte of the SMT timer register

Bits 7:0 – TMRL[7:0] Lower byte of the SMT timer register

Address	Name	Bit Pos.								
		15:8	FSRH[7:0]							
0x0688	BSR	7:0			BSR[5:0]					
0x0689	WREG	7:0	WREG[7:0]							
0x068A	PCLATH	7:0		PCLATH[6:0]						
0x068B	INTCON	7:0	GIE	PEIE						INTEDG
0x068C	CWG3CLK	7:0								CS
0x068D	CWG3ISM	7:0					ISM[3:0]			
0x068E	CWG3DBR	7:0			DBR[5:0]					
0x068F	CWG3DBF	7:0			DBF[5:0]					
0x0690	CWG3CON0	7:0	EN	LD				MODE[2:0]		
0x0691	CWG3CON1	7:0			IN		POLD	POLC	POLB	POLA
0x0692	CWG3AS0	7:0	SHUTDOWN	REN	LSBD[1:0]		LSAC[1:0]			
0x0693	CWG3AS1	7:0			AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
0x0694	CWG3STR	7:0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA
0x0695 ... 0x06FF	Reserved									
0x0700	INDF0	7:0	INDF0[7:0]							
0x0701	INDF1	7:0	INDF1[7:0]							
0x0702	PCL	7:0	PCL[7:0]							
0x0703	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0704	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0706	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0708	BSR	7:0			BSR[5:0]					
0x0709	WREG	7:0	WREG[7:0]							
0x070A	PCLATH	7:0		PCLATH[6:0]						
0x070B	INTCON	7:0	GIE	PEIE						INTEDG
0x070C	PIR0	7:0			TMR0IF	IOCIF				INTF
0x070D	PIR1	7:0	OSFIF	CSWIF					ADTIF	ADIF
0x070E	PIR2	7:0		ZCDIF					C2IF	C1IF
0x070F	PIR3	7:0	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF
0x0710	PIR4	7:0			TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF
0x0711	PIR5	7:0	CLC4IF	CLC3IF	CL24IF	CLC1IF		TMR5GIF	TMR3GIF	TMR1GIF
0x0712	PIR6	7:0				CCP5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
0x0713	PIR7	7:0			NVMIF	NCO1IF		CWG3IF	CWG2IF	CWG1IF
0x0714	PIR8	7:0			SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
0x0715	Reserved									
0x0716	PIE0	7:0			TMR0IE	IOCIE				INTE
0x0717	PIE1	7:0	OSFIE	CSWIE					ADTIE	ADIE
0x0718	PIE2	7:0		ZCDIE					C2IE	C1IE
0x0719	PIE3	7:0	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE
0x071A	PIE4	7:0			TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE
0x071B	PIE5	7:0	CLC4IE	CLC3IE	CLC2IE	CLC1IE		TMR5GIE	TMR3GIE	TMR1GIE
0x071C	PIE6	7:0				CCP5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE

Address	Name	Bit Pos.								
0x1284	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1286	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1288	BSR	7:0			BSR[5:0]					
0x1289	WREG	7:0	WREG[7:0]							
0x128A	PCLATH	7:0		PCLATH[6:0]						
0x128B	INTCON	7:0	GIE	PEIE						INTEDG
0x128C ... 0x12FF	Reserved									
0x1300	INDF0	7:0	INDF0[7:0]							
0x1301	INDF1	7:0	INDF1[7:0]							
0x1302	PCL	7:0	PCL[7:0]							
0x1303	STATUS	7:0				$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
0x1304	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1306	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1308	BSR	7:0			BSR[5:0]					
0x1309	WREG	7:0	WREG[7:0]							
0x130A	PCLATH	7:0		PCLATH[6:0]						
0x130B	INTCON	7:0	GIE	PEIE						INTEDG
0x130C ... 0x137F	Reserved									
0x1380	INDF0	7:0	INDF0[7:0]							
0x1381	INDF1	7:0	INDF1[7:0]							
0x1382	PCL	7:0	PCL[7:0]							
0x1383	STATUS	7:0				$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
0x1384	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1386	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1388	BSR	7:0			BSR[5:0]					
0x1389	WREG	7:0	WREG[7:0]							
0x138A	PCLATH	7:0		PCLATH[6:0]						
0x138B	INTCON	7:0	GIE	PEIE						INTEDG
0x138C ... 0x13FF	Reserved									
0x1400	INDF0	7:0	INDF0[7:0]							
0x1401	INDF1	7:0	INDF1[7:0]							
0x1402	PCL	7:0	PCL[7:0]							
0x1403	STATUS	7:0				$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
0x1404	FSR0	7:0	FSRL[7:0]							

<table>
<tr><th>MOVWI</th><th colspan="3">Move W to INDFn</th></tr>
<tr><td></td><td colspan="3">-32 ≤ k ≤ 31</td></tr>
<tr><td>Operation:</td><td colspan="3">(W) → INDFn
Effective address is determined by
• FSR + 1 (preincrement)
• FSR - 1 (predecrement)
• FSR + k (relative offset)
After the Move, the FSR value will be either:
• FSR + 1 (all increments)
• FSR - 1 (all decrements)
• Unchanged</td></tr>
<tr><td rowspan="6">Status Affected:</td><td colspan="3">None</td></tr>
<tr><td><b>MODE</b></td><td><b>SYNTAX</b></td><td><b>mm</b></td></tr>
<tr><td>Preincrement</td><td>++FSRn</td><td>00</td></tr>
<tr><td>Predecrement</td><td>--FSRn</td><td>01</td></tr>
<tr><td>Postincrement</td><td>FSRn++</td><td>10</td></tr>
<tr><td>Postdecrement</td><td>FSRn--</td><td>11</td></tr>
<tr><td>Description:</td><td colspan="3">This instruction is used to move data between W and one of the indirect registers (INDFn).
Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/ decrementing it.
The INDFn registers are not physical registers.
Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.
FSRn is limited to the range 0000h-FFFFh.
Incrementing/decrementing it beyond these bounds will cause it to wrap-around.
The increment/decrement operation on FSRn WILL NOT affect any Status bits.</td></tr>
</table>

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