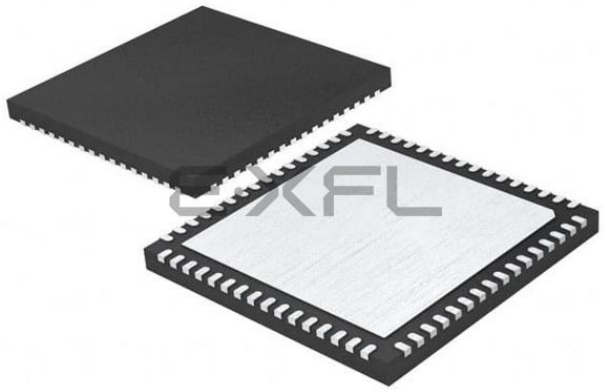


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	60
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 45x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf66k40-i-mr

TABLE 10-4: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F6XK40 DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
F5Fh	ADPCH	F37h	CWG1DAT	F0Fh	SMT1CPRU	EE7h	RCSTA4	EBFh	ODCONH
F5Eh	ADPRE	F36h	CWG1CLK	F0Eh	SMT1CPRH	EE6h	SPBRGH4	EBEh	SLRCONH
F5Dh	ADCAP	F35h	CLKRCLK	F0Dh	SMT1CPRL	EE5h	SPBRGL4	EBDh	INLVLH
F5Ch	ADACQ	F34h	CLKRCON	F0Ch	SMT1TMRU	EE4h	TXREG4	EBCh	ANSELG
F5Bh	ADCON3	F33h	T7CLK	F0Bh	SMT1TMRH	EE3h	RCREG4	EBBh	WPUG
F5Ah	ADCON2	F32h	T7GATE	F0Ah	SMT1TMRL	EE2h	BAUDCON5	EBAh	ODCONG
F59h	ADCON1	F31h	T7GCON	F09h	SMT2WIN	EE1h	TXSTA5	EB9h	SLRCONG
F58h	ADREF	F30h	T7CON	F08h	SMT2SIG	EE0h	RCSTA5	EB8h	INLVLG
F57h	ADCLK	F2Fh	TMR7H	F07h	SMT2CLK	EDFh	SPBRGH5	EB7h	IOCGP
F56h	ADACT	F2Eh	TMR7L	F06h	SMT2STAT	EDEh	SPBRGL5	EB6h	IOCGN
F55h	MDMDCARH	F2Dh	T8ERS	F05h	SMT2CON1	EDDh	TXREG5	EB5h	IOCGF
F54h	MDCARL	F2Ch	T8CLK	F04h	SMT2CON0	EDCh	RCREG5	EB4h	ANSELF
F53h	MDSRC	F2Bh	T8HLT	F03h	SMT2PRU	EDBh	SSP2CON3	EB3h	WPUF
F52h	MDCON1	F2Ah	T8CON	F02h	SMT2PRH	EDAh	SSP2CON2	EB2h	ODCONF
F51h	MDCON0	F29h	T8PR	F01h	SMT2PRL	ED9h	SSP2CON1	EB1h	SLRCONF
F50h	SCANDTI	F28h	T8TMR	F00h	SMT2CPWU	ED8h	SSP2STAT	EB0h	INLVLF
F4Fh	SCANCON0	F27h	CCP3CAP	EFFh	SMT2CPWH	ED7h	SSP2MASK	EAFh	ANSELE
F4Eh	SCANLADDRU	F26h	CCP3CON	EFEh	SMT2CPWL	ED6h	SSP2ADD	EAEh	WPUE
F4Dh	SCANLADDRH	F25h	CCP3H	EFDh	SMT2CPRU	ED5h	SSP2BUF	EADh	ODCONE
F4Ch	SCANLADDRL	F24h	CCP3L	EFCh	SMT2CPRH	ED4h	CMOUT	EACh	SLRCONE
F4Bh	SCANNADDRU	F23h	CCP4CAP	EFBh	SMT2CPRL	ED3h	CM1PCH	EABh	INLVLE
F4Ah	SCANNADDRH	F22h	CCP4CON	EFAh	SMT2TMRU	ED2h	CM1NCH	EAAh	IOCEP
F49h	SCANNADDRL	F21h	CCP4H	EF9h	SMT2TMRH	ED1h	CM1CON1	EA9h	IOCEN
F48h	CRCCON1	F20h	CCP4L	EF8h	SMT2TMRL	ED0h	CM1CON0	EA8h	IOCEF
F47h	CRCCON0	F1Fh	CCP5CAP	EF7h	BAUDCON2	ECFh	CM2PCH	EA7h	ANSELD
F46h	CRCXORH	F1Eh	CCP5CON	EF6h	TXSTA2	ECEh	CM2NCH	EA6h	WPUD
F45h	CRCXORL	F1Dh	CCP5H	EF5h	RCSTA2	ECDh	CM2CON1	EA5h	ODCOND
F44h	CRCSHFTH	F1Ch	CCP5L	EF4h	SPBRGH2	ECCh	CM2CON0	EA4h	SLRCOND
F43h	CRCSHFTL	F1Bh	SMT1WIN	EF3h	SPBRGL2	ECBh	CM3PCH	EA3h	INLVLD
F42h	CRCACCH	F1Ah	SMT1SIG	EF2h	TXREG2	ECAh	CM3NCH	EA2h	—
F41h	CRCACCL	F19h	SMT1CLK	EF1h	RCREG2	EC9h	CM3CON1	EA1h	WPUC
F40h	CRCDATAH	F18h	SMT1STAT	EF0h	BAUDCON3	EC8h	CM3CON0	EA0h	ODCONC
F3Fh	CRCDATAL	F17h	SMT1CON1	EEFh	TXSTA3	EC7h	DAC1CON1	E9Fh	SLRCONC
F3Eh	CWG1STR	F16h	SMT1CON0	EEEh	RCSTA3	EC6h	DAC1CON0	E9Eh	INLVLC
F3Dh	CWG1ASD1	F15h	SMT1PRU	EEDh	SPBRGH3	EC5h	ZCDCON	E9Dh	IOCCP
F3Ch	CWG1ASD0	F14h	SMT1PRH	EECh	SPBRGL3	EC4h	FVRCON	E9Ch	IOCCN
F3Bh	CWG1CON1	F13h	SMT1PRL	EEBh	TXREG3	EC3h	LVDCON1	E9Bh	IOCCF
F3Ah	CWG1CON0	F12h	SMT1CPWU	EEAh	RCREG3	EC2h	LVDCON0	E9Ah	ANSELB
F39h	CWG1DBF	F11h	SMT1CPWH	EE9h	BAUDCON4	EC1h	—	E99h	WPUB
F38h	CWG1DBR	F10h	SMT1CPWL	EE8h	TXSTA4	EC0h	WPUH	E98h	ODCONB

TABLE 10-5: REGISTER FILE SUMMARY FOR PIC18(L)F6XK40 DEVICES (CONTINUED)

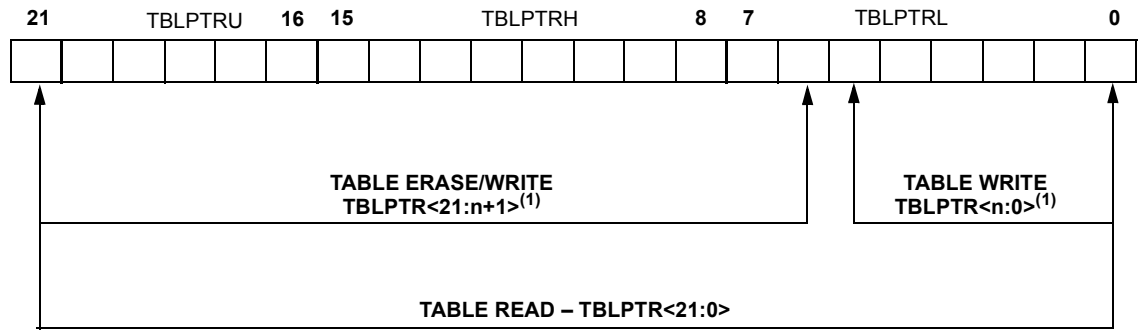
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
DFAh	T5GPPS	—	—	T5GPPS<5:0>						--110100
DF9h	T5CKIPPS	—	—	T5CKIPPS<5:0>						--011001
DF8h	T3GPPS	—	—	T3GPPS<5:0>						--000101
DF7h	T3CKIPPS	—	—	T3CKIPPS<5:0>						--001101
DF6h	T1GPPS	—	—	T1GPPS<5:0>						--001101
DF5h	T1CKIPPS	—	—	T1CKIPPS<5:0>						--010000
DF4h	T0CKIPPS	—	—	T0CKIPPS<5:0>						--000100
DF3h	INT3PPS	—	—	INT3PPS<5:0>						--000100
DF2h	INT2PPS	—	—	INT2PPS<5:0>						--001010
DF1h	INT1PPS	—	—	INT1PPS<5:0>						--001001
DF0h	INT0PPS	—	—	INT0PPS<5:0>						--001000
DE0h	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	-------0
DD0h to E7Eh	—	Unimplemented								—

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not available on LF devices.

TABLE 11-3: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 11-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



Note 1: Refer to Table 11-3 for the row size values.

REGISTER 11-2: NVMCON2: NONVOLATILE MEMORY CONTROL 2 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMCON2<7:0>							
bit 7							bit 0

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

bit 7-0 **NVMCON2<7:0>:**
Refer to **Section 11.1.4 "NVM Unlock Sequence"**.

Note 1: This register always reads zeros, regardless of data written.

Register 11-3: NVMADRL: Data EEPROM Memory Address Low

R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0
NVMADR<7:0>							
bit 7							bit 0

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

bit 7-0 **NVMADR<7:0>:** EEPROM Read Address bits

REGISTER 11-4: NVMADRH: DATA EEPROM MEMORY ADDRESS HIGH

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u
—	—	—	—	—	—	NVMADR<9:8>	
bit 7							bit 0

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

bit 7-2 **Unimplemented:** Read as '0'
bit 1-0 **NVMADR<9:8>:** EEPROM Read Address bits

REGISTER 13-16: SCANHADRH: SCAN HIGH ADDRESS HIGH 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
HADR<15:8>(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 7-0 **HADR<15:8>**: Scan End Address bits(1, 2)
Most Significant bits of the address at the end of the designated scan

Note 1: Registers SCANHADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 13-17: SCANHADRL: SCAN HIGH 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
HADR<7:0>(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 7-0 **HADR<7:0>**: Scan End Address bits(1, 2)
Least Significant bits of the address at the end of the designated scan

Note 1: Registers SCANHADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 13-18: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<3:0>			
bit 7							bit 0

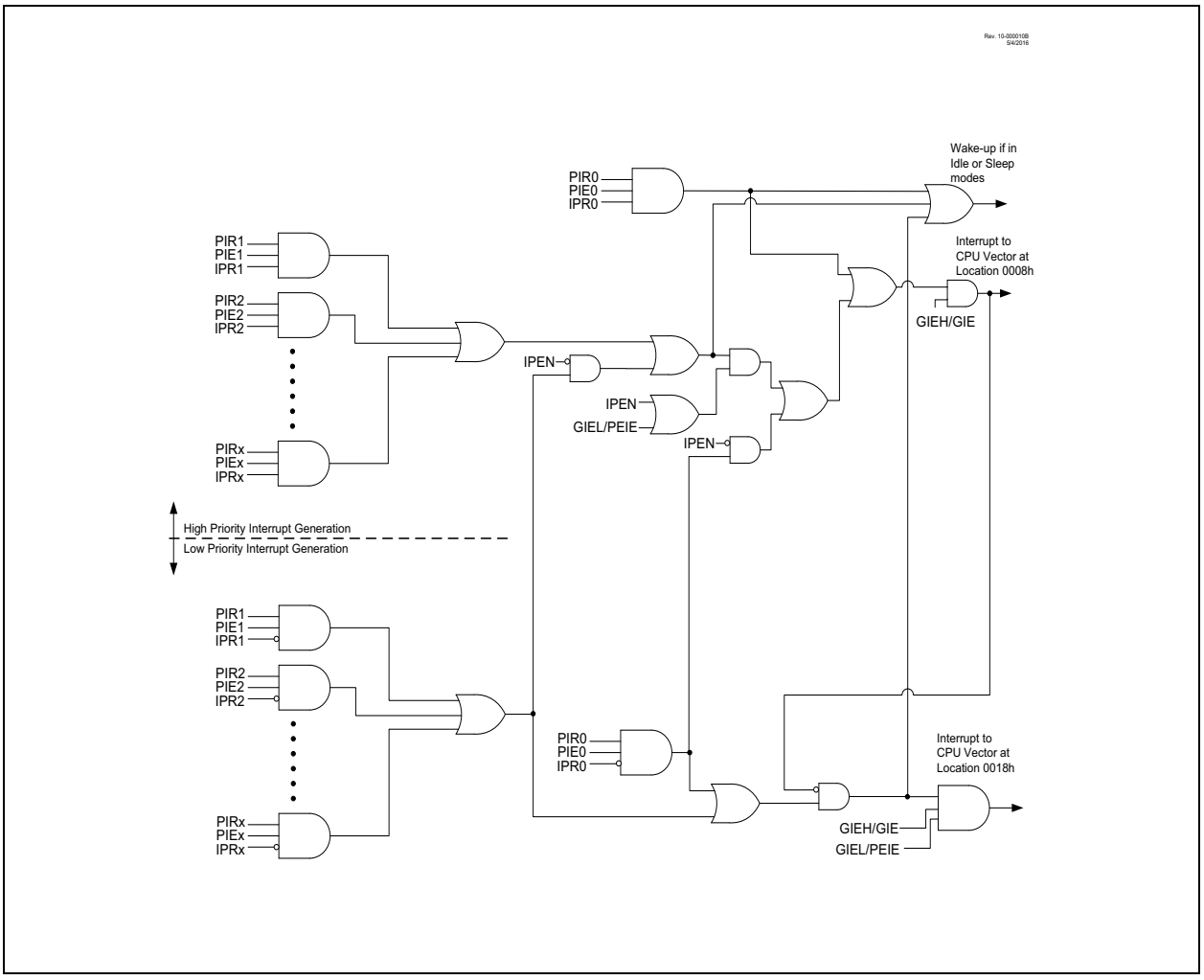
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **TSEL<3:0>**: Scanner Data Trigger Input Selection bits

1111-1101 = Reserved
1100 = SMT2OUT
1011 = SMT1OUT
1010 = TMR8_postscaled
1001 = TMR7_output
1000 = TMR6_postscaled
0111 = TMR5_output
0110 = TMR4_postscaled
0101 = TMR3_output
0100 = TMR2_postscaled
0011 = TMR1_output
0010 = TMR0_output
0001 = CLKREF_output
0000 = LFINTOSC

FIGURE 14-1: PIC18 INTERRUPT LOGIC



REGISTER 14-21: PIE9: PERIPHERAL INTERRUPT ENABLE REGISTER 9

U-0	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
—	—	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE
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 **SMT2PWAIE:** SMT2 Pulse Width Acquisition Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **SMT2PRAIE:** SMT2 Period Acquisition Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **SMT2IE:** SMT2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **SMT1PWAIE:** SMT1 Pulse Width Acquisition Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **SMT1PRAIE:** SMT1 Period Acquisition Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **SMT1IE:** SMT1 Interrupt Enable bit
1 = Enabled
0 = Disabled

REGISTER 14-29: IPR7: PERIPHERAL INTERRUPT PRIORITY REGISTER 7

U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	CCP5IP	CCP4IP	CCP3IP	CCP2IP	CCP1IP
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **CCP5IP:** ECCP5 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CCP4IP:** ECCP4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP3IP:** ECCP3 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP2IP:** ECCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **CCP1IP:** ECCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 15-3: LATx: LATx REGISTER(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
LATx7	LATx6	LATx5	LATx4	LATx3	LATx2	LATx1	LATx0
bit 7							bit 0

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

bit 7-0 **LATx<7:0>:** Rx7:Rx0 Output Latch Value bits

Note 1: Writes to LATx are equivalent with writes to the corresponding PORTx register. Reads from LATx register return register values, not I/O pin values.

TABLE 15-3: LAT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0
LATG	LATG7	LATG6	—	LATG4	LATG3	LATG2	LATG1	LATG0
LATH	—	—	—	—	LATH3	LATH2	LATH1	LATH0

19.9 Timer1/3/5/7 Interrupt

The Timer1/3/5/7 register pair (TMRxH:TMRxL) increments to FFFFh and rolls over to 0000h. When Timer1/3/5/7 rolls over, the Timer1/3/5/7 interrupt flag bit of the PIR5 register is set. To enable the interrupt-on-rollover, you must set these bits:

- TMRxON bit of the TxCON register
- TMRxIE bits of the PIE5 register
- PEIE/GIEL bit of the INTCON register
- GIE/GIEH bit of the INTCON register

The interrupt is cleared by clearing the TMRxIF bit in the Interrupt Service Routine.

For more information on selecting high or low priority status for the Timer1/3/5/7 Overflow Interrupt, see **Section 14.0 "Interrupts"**.

Note:	The TMRxH:TMRxL register pair and the TMRxIF bit should be cleared before enabling interrupts.

19.10 Timer1/3/5/7 Operation During Sleep

Timer1/3/5/7 can only operate during Sleep when set up in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMRxON bit of the TxCON register must be set
- TMRxIE bit of the PIE5 register must be set
- PEIE/GIEL bit of the INTCON register must be set
- $\overline{\text{TxSYNC}}$ bit of the TxCON register must be set
- Configure the TMRxCLK register for using secondary oscillator as the clock source
- Enable the SOSCEN bit of the OSCEN register (Register 4-7)

The device will wake-up on an overflow and execute the next instruction. If the GIE/GIEH bit of the INTCON register is set, the device will call the Interrupt Service Routine.

The secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{TxSYNC}}$ bit setting.

19.11 CCP Capture/Compare Time Base

The CCP modules use the TMRxH:TMRxL register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMRxH:TMRxL register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value in the CCPRxH:CCPRxL register pair matches the value in the TMRxH:TMRxL register pair. This event can be a Special Event Trigger.

For more information, see **Section 21.0 "Capture/Compare/PWM Module"**.

19.12 CCP Special Event Trigger

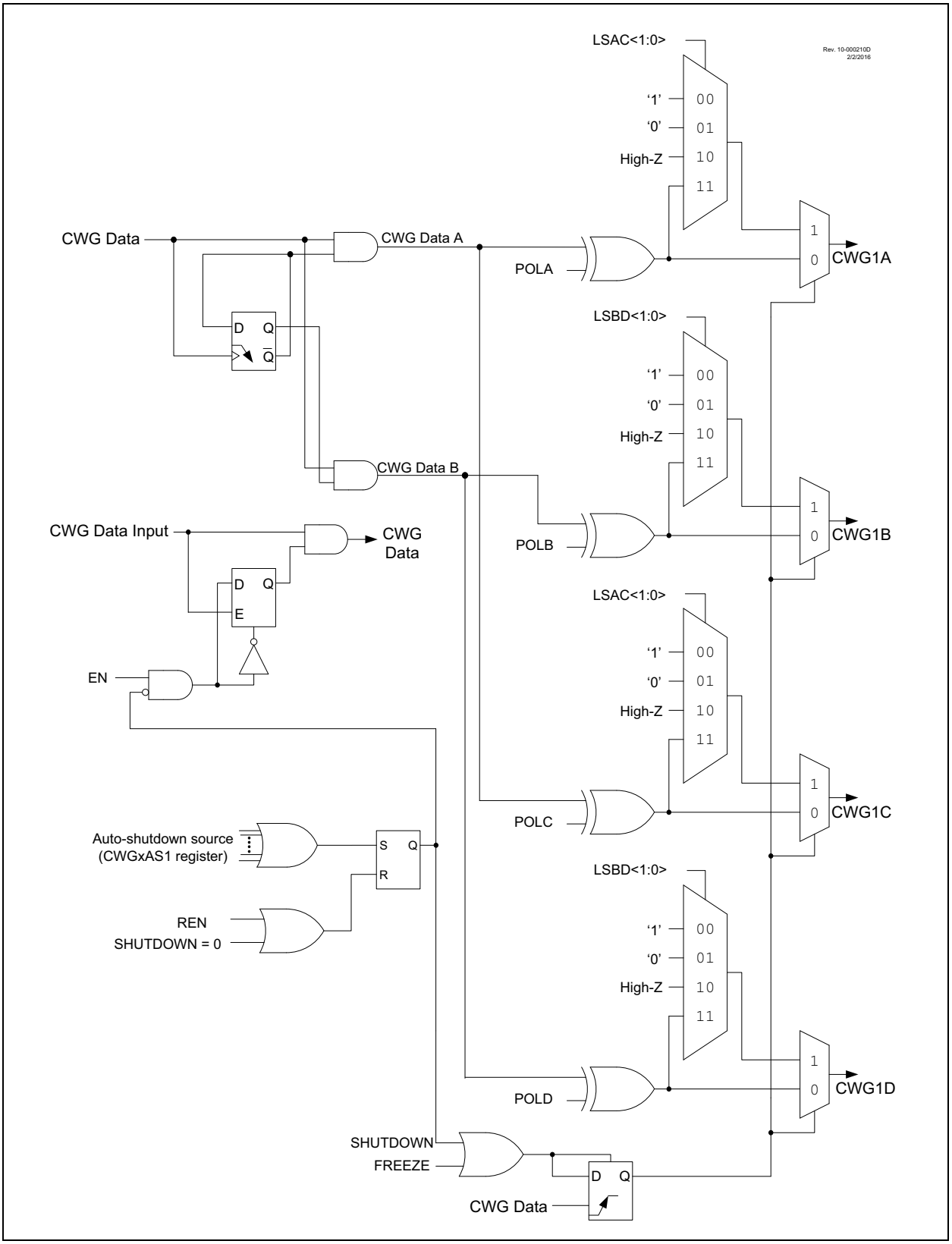
When any of the CCP's are configured to trigger a special event, the trigger will clear the TMRxH:TMRxL register pair. This special event does not cause a Timer1/3/5/7 interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPRxH:CCPRxL register pair becomes the period register for Timer1/3/5/7.

Timer1/3/5/7 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Special Event Trigger. Asynchronous operation of Timer1/3/5/7 can cause a Special Event Trigger to be missed.

In the event that a write to TMRxH or TMRxL coincides with a Special Event Trigger from the CCP, the write will take precedence.

FIGURE 24-3: SIMPLIFIED CWG BLOCK DIAGRAM (PUSH-PULL MODE, MODE<2:0> = 101)



25.0 SIGNAL MEASUREMENT TIMER (SMT)

The SMT is a 24-bit counter with advanced clock and gating logic, which can be configured for measuring a variety of digital signal parameters such as pulse width, frequency and duty cycle, and the time difference between edges on two signals.

Features of the SMT include:

- 24-bit timer/counter
  - Three 8-bit registers (SMTxTMRL/H/U)
  - Readable and writable
  - Optional 16-bit operating mode
- Two 24-bit measurement capture registers
- One 24-bit period match register
- Multi-mode operation, including relative timing measurement
- Interrupt on period match
- Multiple clock, gate and signal sources
- Interrupt on acquisition complete

Ability to read current input values

25.1 Register Definitions: SMT Control

Long bit name prefixes for the SMT peripherals are shown in Table 25-1. Refer to **Section 1.4.2.2 "Long Bit Names"** for more information.

TABLE 25-1:

Peripheral	Bit Name Prefix
SMT1	SMT1
SMT2	SMT2

REGISTER 25-1: SMTxCON0: SMT CONTROL REGISTER 0

R/W-0/0	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
EN(1)	—	STP	WPOL	SPOL	CPOL	SMTxPS<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 **EN:** SMT Enable bit(1)
1 = SMT is enabled
0 = SMT is disabled; internal states are reset, clock requests are disabled

bit 6 **Unimplemented**: Read as '0'

bit 5 **STP:** SMT Counter Halt Enable bit
When SMTxTMR = SMTxPR:
1 = Counter remains SMTxPR; period match interrupt occurs when clocked
0 = Counter resets to 24'h000000; period match interrupt occurs when clocked

bit 4 **WPOL:** SMTxWIN Input Polarity Control bit
1 = SMTxWIN signal is active-low/falling edge enabled
0 = SMTxWIN signal is active-high/rising edge enabled

bit 3 **SPOL:** SMTxSIG Input Polarity Control bit
1 = SMTx_signal is active-low/falling edge enabled
0 = SMTx_signal is active-high/rising edge enabled

bit 2 **CPOL:** SMT Clock Input Polarity Control bit
1 = SMTxTMR increments on the falling edge of the selected clock signal
0 = SMTxTMR increments on the rising edge of the selected clock signal

bit 1-0 **SMTxPS<1:0>:** SMT Prescale Select bits
11 = Prescaler = 1:8
10 = Prescaler = 1:4
01 = Prescaler = 1:2
00 = Prescaler = 1:1

Note 1: Setting EN to '0' does not affect the register contents.

REGISTER 27-4: SSPxBUF: MSSP DATA BUFFER REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
BUF<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 **BUF<7:0>:** MSSP Buffer bits

REGISTER 27-5: SSPxADD: MSSP ADDRESS REGISTER (SPI MODE)

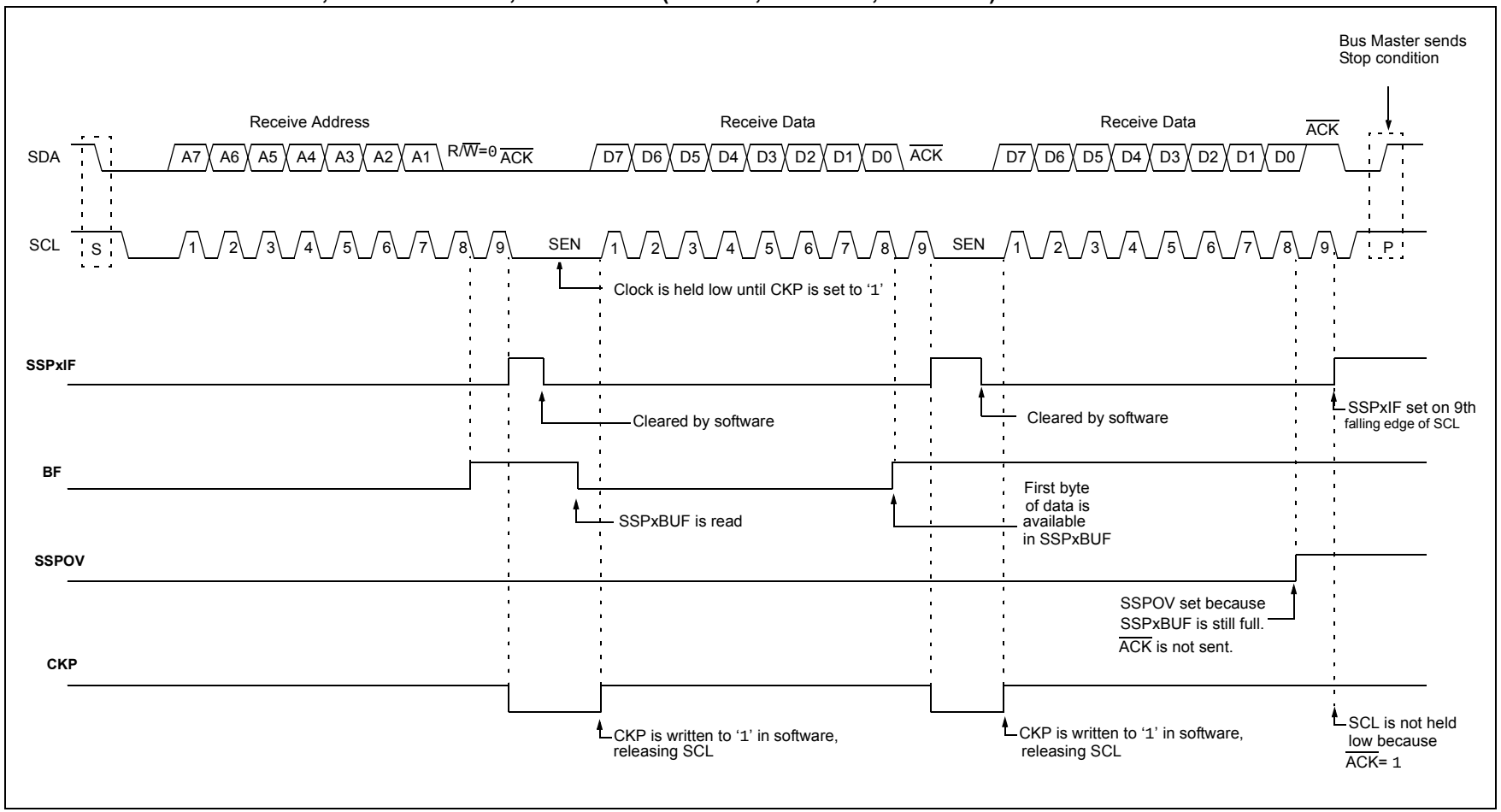
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD<7:0>							
bit 7							bit 0

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	

Master mode: SPI mode

bit 7-0 Baud Rate Clock Divider bits
SCK/SCL pin clock period = ((SSPxADD<7:0> + 1) *4)/FOSC

FIGURE 27-15: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)



27.11 Baud Rate Generator

The MSSP module has a Baud Rate Generator available for clock generation in both I^2C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 27-5). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

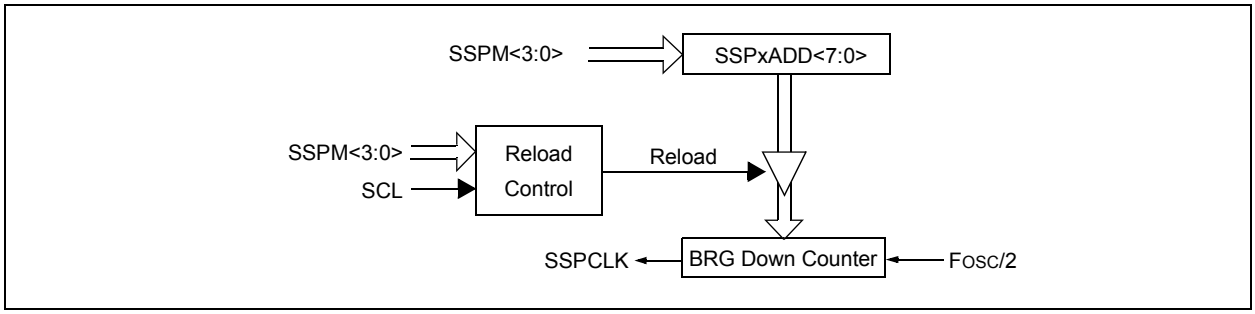
An internal signal “Reload” in Figure 27-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 27-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

EQUATION 27-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPADD + 1)(4)}$$

FIGURE 27-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I^2C. This is an implementation limitation.

TABLE 27-3: MSSP CLOCK RATE W/BRG

FOSC	FCY	BRG Value	FCLOCK (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: Refer to the I/O port electrical specifications in Table 37-8: Internal Oscillator Parameters, to ensure the system is designed to support IOL requirements.

ADDWFC **ADD W and CARRY bit to f**

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

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

Operation: (W) + (f) + (C) → dest

Status Affected: N,OV, C, DC, Z

Encoding:

0010	00da	ffff	ffff

Description: Add W, the CARRY flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 36.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ADDWFC REG, 0, 1`

Before Instruction
CARRY bit = 1
REG = 02h
W = 4Dh
After Instruction
CARRY bit = 0
REG = 02h
W = 50h

ANDLW **AND literal with W**

Syntax: ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. k → W

Status Affected: N, Z

Encoding:

0000	1011	kkkk	kkkk

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ANDLW 05Fh`

Before Instruction
W = A3h
After Instruction
W = 03h

NEGF **Negate f**

Syntax: NEGF f {,a}

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

Operation: $(\overline{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

0110	110a	ffff	ffff

Description: Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 36.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
NEGF    REG, 1
```
Before Instruction
REG = 0011 1010 [3Ah]
After Instruction
REG = 1100 0110 [C6h]

NOP **No Operation**

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000 1111	0000 xxxx	0000 xxxx	0000 xxxx

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

CALLW **Subroutine Call Using WREG**

Syntax: CALLW

Operands: None

Operation: (PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → PCU

Status Affected: None

Encoding:

0000	0000	0001	0100

Description: First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched. Unlike CALL, there is no option to update W, Status or BSR.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read WREG	PUSH PC to stack	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CALLW
```
Before Instruction
```
PC      =   address (HERE)
PCLATH  =   10h
PCLATU  =   00h
W       =   06h
```
After Instruction
```
PC      =   001006h
TOS     =   address (HERE + 2)
PCLATH  =   10h
PCLATU  =   00h
W       =   06h
```

MOVSF **Move Indexed to f**

Syntax: MOVSF [z_s], f_d

Operands: $0 \le z_s \le 127$
$0 \le f_d \le 4095$

Operation: ((FSR2) + z_s) → f_d

Status Affected: None

Encoding:

1st word (source)	1110	1011	0zzz	$zzzz_s$
2nd word (destin.)	1111	ffff	ffff	$ffff_d$

Description: The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).
The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.
If the resultant source address points to an indirect addressing register, the value returned will be 00h.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example:
```
MOVSF   [05h], REG2
```
Before Instruction
```
FSR2            =   80h
Contents
of 85h          =   33h
REG2            =   11h
```
After Instruction
```
FSR2            =   80h
Contents
of 85h          =   33h
REG2            =   33h
```