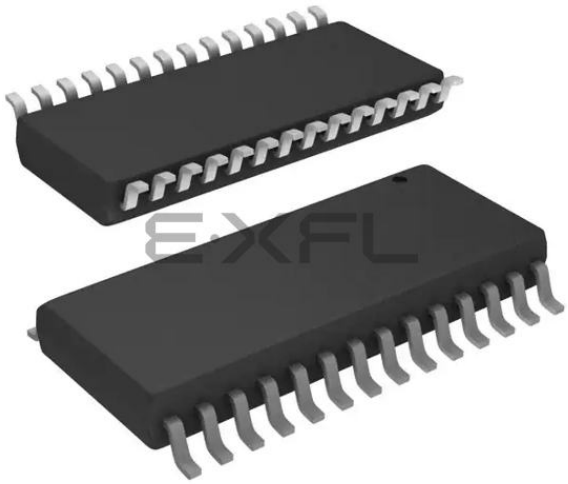


Welcome to E-XFL.COM

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, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f26k42-e-so

TABLE 3: 48-PIN ALLOCATION TABLE FOR PIC18(L)F5XK42 (CONTINUED)

I/O	48-Pin TQFP	48-Pin UQFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^2C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC2	40	40	ANC2	—	-	—	—	—	—	—	—	T5CKI(1)	CCP1(1)	—	—	—	—	IOCC2	—
RC3	41	41	ANC3	—	-	—	—	SCL1(3,4)	SCK1(1)	—	—	T2IN(1)	-	—	—	—	—	IOCC3	—
RC4	46	46	ANC4	—	—	—	—	SDA1(3,4)	SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	47	47	ANC5	—	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	48	48	ANC6	—	—	—	—	—	—	CTS1(1)	—	—	—	—	—	—	—	IOCC6	—
RC7	1	1	ANC7	—	—	—	—	—	—	RX1(1)	—	—	—	—	—	—	—	IOCC7	—
RD0	42	42	AND0	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD1	43	43	AND1	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD2	44	44	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD3	45	45	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD4	2	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD5	3	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD6	4	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD7	5	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE0	27	27	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE1	28	28	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE2	29	29	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE3	20	20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	$\overline{MCLR}$ V_{PP}
RF0	36	36	ANF0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF1	37	37	ANF1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF2	38	38	ANF2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF3	39	39	ANF3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF4	12	12	ANF4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF5	13	13	ANF5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF6	14	14	ANF6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF7	15	15	ANF7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.
2: All output signals shown in this row are PPS remappable.
3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
4: These pins can be configured for I^2C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4/RD0/RD1 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

REGISTER 5-9: CONFIGURATION WORD 5L (30 0008h)

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/W-1
—	—	—	—	—	—	—	$\overline{CP}$
bit 7							bit 0

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

bit 7-1 **Unimplemented:** Read as '1'

bit 0 **$\overline{CP}$:** User Program Flash Memory and Data EEPROM Code Protection bit
1 = User Program Flash Memory and Data EEPROM code protection is disabled
0 = User Program Flash Memory and Data EEPROM code protection is enabled

REGISTER 5-10: CONFIGURATION WORD 5H (30 0009h)

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 7-0 **Unimplemented:** Read as '1'

TABLE 5-2: SUMMARY OF CONFIGURATION WORDS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
30 0000h	CONFIG1L	—	RSTOSC<2:0>			—	FEXTOSC<2:0>			1111 1111
30 0001h	CONFIG1H	—	—	FCMEN	—	CSWEN	—	PR1WAY	$\overline{CLKOUTEN}$	1111 1111
30 0002h	CONFIG2L	BOREN<1:0>		$\overline{LPBOREN}$	IVT1WAY	MVECEN	PWRTS<1:0>		MCLRE	1111 1111
30 0003h	CONFIG2H	$\overline{XINST}$	—	$\overline{DEBUG}$	STVREN	PPS1WAY	$\overline{ZCD}$	BORV<1:0>		1111 1111
30 0004h	CONFIG3L	—	WDTE<1:0>		WDTCPS<4:0>					1111 1111
30 0005h	CONFIG3H	—	—	WDTCCS<2:0>			WDTCWS<2:0>			1111 1111
30 0006h	CONFIG4L	$\overline{WRTAPP}$	—	—	$\overline{SAFEN}$	$\overline{BBEN}$	BBSIZE<2:0>			1111 1111
30 0007h	CONFIG4H	—	—	LVP	—	$\overline{WRTSAF}$	$\overline{WRTD}$	$\overline{WRTC}$	$\overline{WRTB}$	1111 1111
30 0008h	CONFIG5L	—	—	—	—	—	—	—	$\overline{CP}$	1111 1111
30 0009h	CONFIG5H	—	—	—	—	—	—	—	—	1111 1111

EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY (CONTINUED)

```
WRITE_BYTE_TO_HREGS
              MOVF      POSTINC0, W              ; get low byte of buffer data
              MOVWF     TABLAT                   ; present data to table latch
              TBLWT+*                            ; write data, perform a short write
                                                 ; to internal TBLWT holding register.
              DECFSZ    COUNTER                  ; loop until holding registers are full
              BRA       WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
              BCF       NVMCON1, REG0            ; point to Program Flash Memory
              BSF       NVMCON1, REG1            ; point to Program Flash Memory
              BSF       NVMCON1, WREN            ; enable write to memory
              BCF       NVMCON1, FREE            ; enable write to memory
              BCF       INTCON0, GIE             ; disable interrupts
              MOVLW     55h
Required      MOVWF     NVMCON2                  ; write 55h
Sequence      MOVLW     0AAh
              MOVWF     NVMCON2                  ; write 0AAh
              BSF       NVMCON1, WR              ; start program (CPU stall)
              DCFSZ     COUNTER2                 ; repeat for remaining write blocks
              BRA       WRITE_BYTE_TO_HREGS
              BSF       INTCON0, GIE             ; re-enable interrupts
              BCF       NVMCON1, WREN            ; disable write to memory
```

21.3 Timer1/3/5 Prescaler

Timer1/3/5 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The CKPS bits of the TxCON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMRxH or TMRxL.

21.4 Timer1/3/5 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{SYNC}}$ of the TxCON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake up the processor. However, special precautions in software are needed to read/write the timer (see **Section 21.4.1 "Reading and Writing Timer1/3/5 in Asynchronous Counter Mode"**).

Note:	When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

21.4.1 READING AND WRITING TIMER1/3/5 IN ASYNCHRONOUS COUNTER MODE

Reading TMRxH or TMRxL while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads. For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMRxH:TMRxL register pair.

21.5 Timer1/3/5 16-Bit Read/Write Mode

Timer1/3/5 can be configured to read and write all 16 bits of data, to and from, the 8-bit TMRxL and TMRxH registers, simultaneously. The 16-bit read and write operations are enabled by setting the RD16 bit of the TxCON register.

To accomplish this function, the TMRxH register value is mapped to a buffer register called the TMRxH buffer register. While in 16-Bit mode, the TMRxH register is not directly readable or writable and all read and write operations take place through the use of this TMRxH buffer register.

When a read from the TMRxL register is requested, the value of the TMRxH register is simultaneously loaded into the TMRxH buffer register. When a read from the TMRxH register is requested, the value is provided from the TMRxH buffer register instead. This provides the user with the ability to accurately read all 16 bits of the Timer1/3/5 value from a single instance in time. Reference the block diagram in Figure 21-2 for more details.

In contrast, when not in 16-Bit mode, the user must read each register separately and determine if the values have become invalid due to a rollover that may have occurred between the read operations.

When a write request of the TMRxL register is requested, the TMRxH buffer register is simultaneously updated with the contents of the TMRxH register. The value of TMRxH must be preloaded into the TMRxH buffer register prior to the write request for the TMRxL register. This provides the user with the ability to write all 16 bits to the TMRxL:TMRxH register pair at the same time.

Any requests to write to the TMRxH directly does not clear the Timer1/3/5 prescaler value. The prescaler value is only cleared through write requests to the TMRxL register.

FIGURE 25-1: SMT BLOCK DIAGRAM

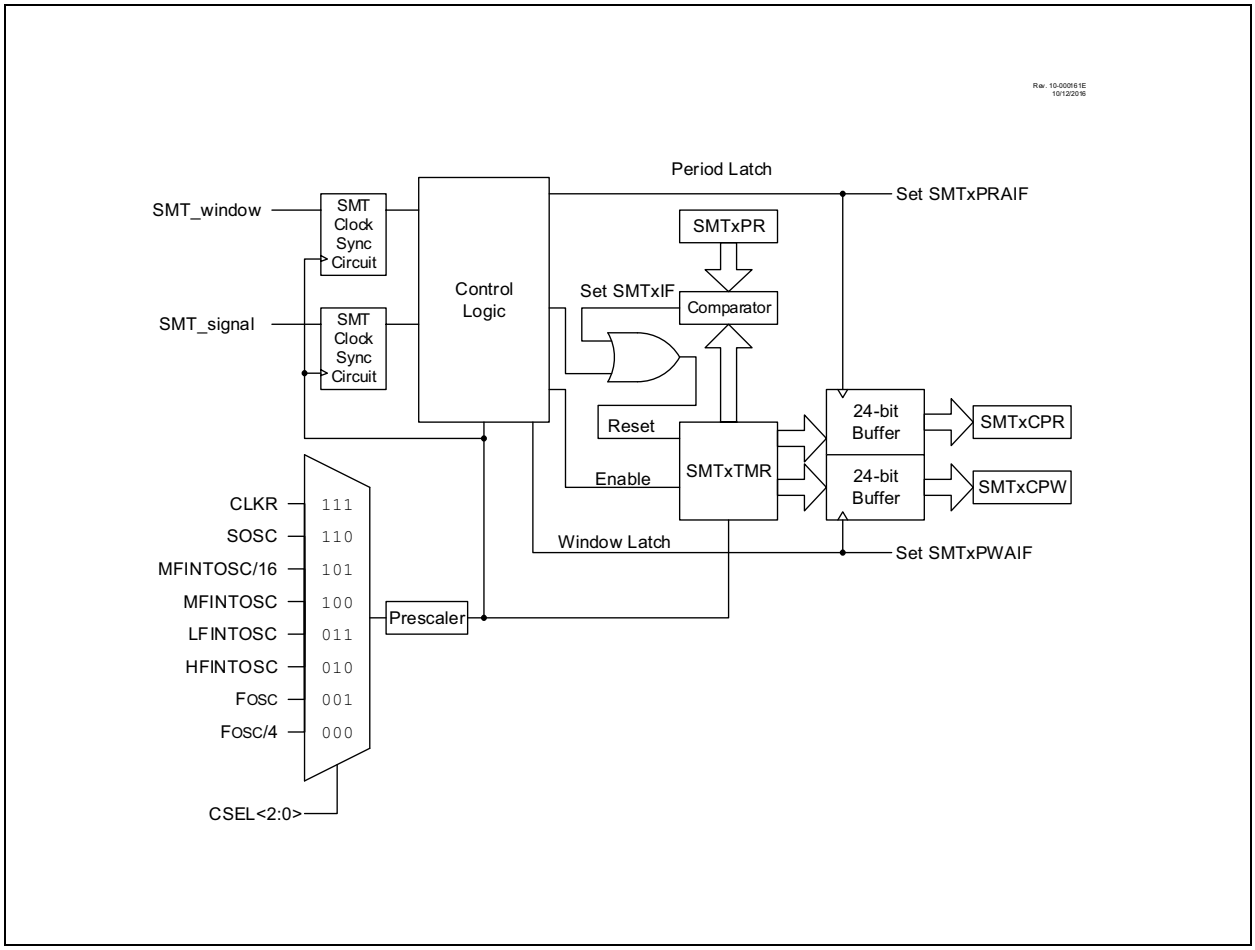
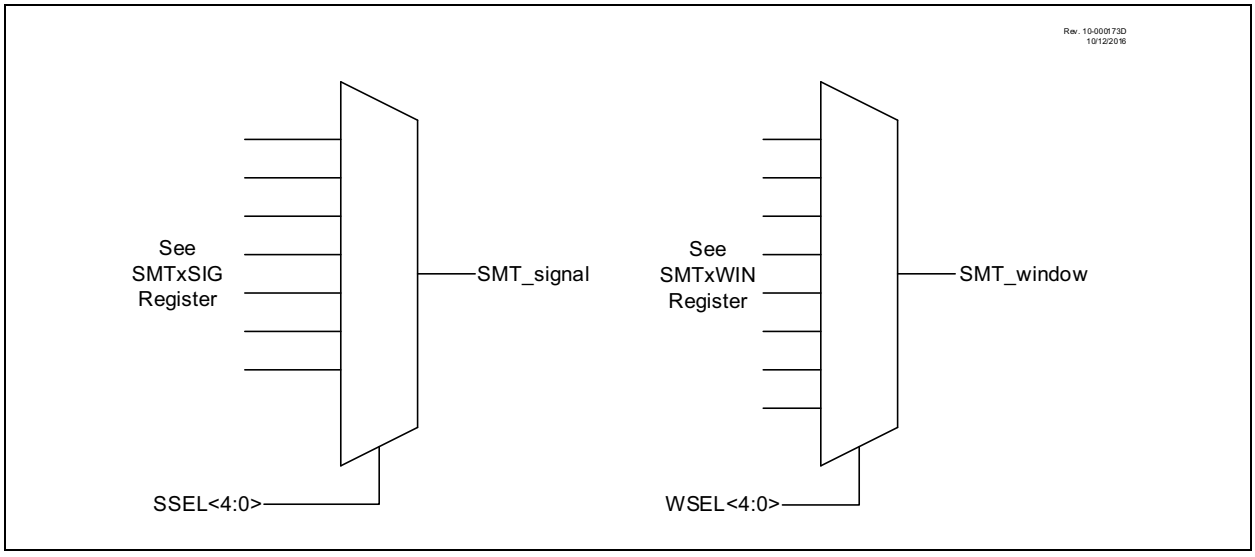


FIGURE 25-2: SMT SIGNAL AND WINDOW BLOCK DIAGRAM



25.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

25.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMT1CPW and SMT1CPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMT1CPW interrupt is controlled by SMT1PWAIF and SMT1PWAIE bits in the respective PIR and PIE registers. The SMT1CPR interrupt is controlled by the SMT1PRAIF and SMT1PRAIE bits, also located in the respective PIR and PIE registers.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMT1CLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

25.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 25.1.2 "Period Match interrupt"**, the SMT will also interrupt upon SMT1TMR, matching SMT1PR with its period match limit functionality described in **Section 25.3 "Halt Operation"**. The period match interrupt is controlled by SMT1IF and SMT1IE, located in the respective PIR and PIE registers.

REGISTER 25-3: SMT1STAT: SMT STATUS REGISTER

R/W/HC-0/0	R/W/HC-0/0	R/W/HC-0/0	U-0	U-0	R-0/0	R-0/0	R-0/0
CPRUP	CPWUP	RST	—	—	TS	WS	AS
bit 7							bit 0

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

bit 7 **CPRUP:** SMT Manual Period Buffer Update bit
1 = Request update to SMT1PRx registers
0 = SMT1PRx registers update is complete

bit 6 **CPWUP:** SMT Manual Pulse Width Buffer Update bit
1 = Request update to SMT1CPW registers
0 = SMT1CPW registers update is complete

bit 5 **RST:** SMT Manual Timer Reset bit
1 = Request Reset to SMT1TMR registers
0 = SMT1TMR registers update is complete

bit 4-3 **Unimplemented**: Read as '0'

bit 2 **TS:** GO Value Status bit
1 = SMT timer is incrementing
0 = SMT timer is not incrementing

bit 1 **WS:** SMT1WIN Value Status bit
1 = SMT window is open
0 = SMT window is closed

bit 0 **AS:** SMT_signal Value Status bit
1 = SMT acquisition is in progress
0 = SMT acquisition is not in progress

28.1 NCO Operation

The NCO operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCO output (NCO_overflow). This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See Equation 28-1.

The NCO output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCO output is then distributed internally to other peripherals and can be optionally output to a pin. The accumulator overflow also generates an interrupt (NCO_overflow).

The NCO period changes in discrete steps to create an average frequency. This output depends on the ability of the receiving circuit (i.e., CWG or external resonant converter circuitry) to average the NCO output to reduce uncertainty.

EQUATION 28-1: NCO OVERFLOW FREQUENCY

$$F_{OVERFLOW} = \frac{NCO\ Clock\ Frequency \times Increment\ Value}{2^{20}}$$

28.1.1 NCO CLOCK SOURCES

Clock sources available to the NCO include:

- FOSC
- HFINTOSC
- LFINTOSC
- MFINTOSC/4 (32 kHz)
- MFINTOSC (500 kHz)
- CLC1/2/3/4_out
- CLKREF
- SOSC

The NCO clock source is selected by configuring the N1CKS<2:0> bits in the NCO1CLK register.

28.1.2 ACCUMULATOR

The accumulator is a 20-bit register. Read and write access to the accumulator is available through three registers:

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

28.1.3 ADDER

The NCO Adder is a full adder, which operates independently from the source clock. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

28.1.4 INCREMENT REGISTERS

The increment value is stored in three registers making up a 20-bit incrementer. In order of LSB to MSB they are:

- NCO1INCL
- NCO1INCH
- NCO1INCU

When the NCO module is enabled, the NCO1INCU and NCO1INCH registers should be written first, then the NCO1INCL register. Writing to the NCO1INCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

The registers are readable and writable. The increment registers are double-buffered to allow value changes to be made without first disabling the NCO module.

When the NCO module is disabled, the increment buffers are loaded immediately after a write to the increment registers.

Note: The increment buffer registers are not user-accessible.

REGISTER 28-2: NCO1CLK: NCO1 INPUT CLOCK CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PWS<2:0>(1,2)			—	CKS<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	

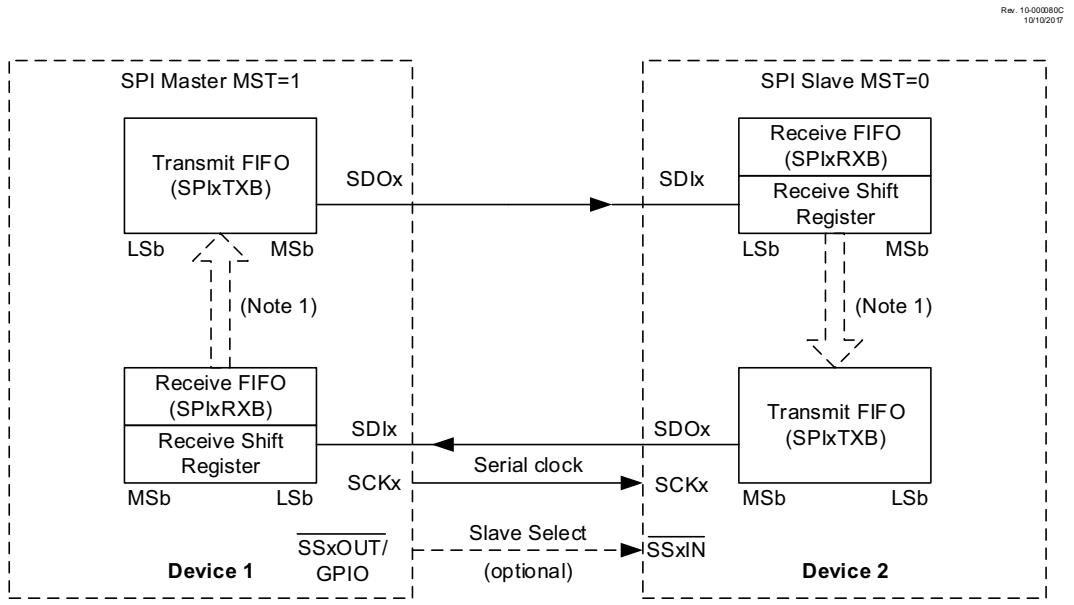
bit 7-5 **PWS<2:0>:** NCO1 Output Pulse Width Select bits(1,2)
- 111 = NCO1 output is active for 128 input clock periods
- 110 = NCO1 output is active for 64 input clock periods
- 101 = NCO1 output is active for 32 input clock periods
- 100 = NCO1 output is active for 16 input clock periods
- 011 = NCO1 output is active for 8 input clock periods
- 010 = NCO1 output is active for 4 input clock periods
- 001 = NCO1 output is active for 2 input clock periods
- 000 = NCO1 output is active for 1 input clock period

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

bit 3-0 **CKS<3:0>:** NCO1 Clock Source Select bits
- 1111 = Reserved
- •
- •
- •
- 1011 = Reserved
- 1010 = CLC4_out
- 1001 = CLC3_out
- 1000 = CLC2_out
- 0111 = CLC1_out
- 0110 = CLKREF_out
- 0101 = SOSC
- 0100 = MFINTOSC/4 (32 kHz)
- 0011 = MFINTOSC (500 kHz)
- 0010 = LFINTOSC
- 0001 = HFINTOSC
- 0000 = FOSC

Note 1: N1PWS applies only when operating in Pulse Frequency mode.
2: If NCO1 pulse width is greater than NCO1 overflow period, operation is undefined.

FIGURE 32-2: SPI MASTER/SLAVE CONNECTION WITH FIFOs



Note 1: In some modes, if the Transmit FIFO is empty, the most recently received byte of data will be transmitted.

2: This diagram assumes that the LSBF bit is cleared (communications are MSb-first). If LSBF is set, the communications will be LSb-first.

32.9 Register definitions: SPI

REGISTER 32-1: SPIxINTF: SPI INTERRUPT FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	R/W/HS-0/0	R/W/HS-0/0	U-0
SRMTIF	TCZIF	SOSIF	EOSIF	—	RXOIF	TXUIF	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0' HS = Bit can be set by hardware

bit 7 **SRMTIF**: Shift Register Empty Interrupt Flag bit

Slave mode:

This bit is ignored

Master mode:

1 = The data transfer is complete

0 = Either no data transfers have occurred or a data transfer is in progress

bit 6 **TCZIF**: Transfer Counter is Zero Interrupt Flag bit

1 = The transfer counter (as defined by BMODE in Register 32-7, TCNTH/L, and TWIDTH) has decremented to zero

0= No interrupt pending

bit 5 **SOSIF**: Start of Slave Select Interrupt Flag bit

1 = SS(in) transitioned from false to true

0 = No interrupt pending

bit 4 **EOSIF**: End of Slave Select Interrupt Flag bit

1 = SS(in) transitioned from true to false

0 = No interrupt pending

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

bit 2 **RXOIF**: Receiver Overflow Interrupt Flag bit

1 = Data transfer completed when RXBF = 1 (edge triggered) and RXR = 1

0 = No interrupt pending

bit 1 **TXUIF**: Transmitter Underflow Interrupt Flag bit

1 = Slave Data transfer started when TXBE = 1 and TXR = 1

0 = No interrupt pending

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

REGISTER 36-15: ADCNT: ADC REPEAT COUNTER 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
CNT<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 **CNT<7:0>**: ADC Repeat Count bits
Counts the number of times that the ADC has been triggered and is used along with CNT to determine when the error threshold is checked when the computation is Low-pass Filter, Burst Average, or Average modes. See Table Table 36-2 for more details.

REGISTER 36-16: ADFLTRH: ADC FILTER HIGH BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FLTR<15:8>							
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 7-0 **FLTR<15:8>**: ADC Filter Output Most Significant bits
In Accumulate, Average, and Burst Average mode, this is equal to ACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

REGISTER 36-17: ADFLTRL: ADC FILTER LOW BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FLTR<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 **FLTR<7:0>**: ADC Filter Output Least Significant bits
In Accumulate, Average, and Burst Average mode, this is equal to ACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

REGISTER 38-4: CMxPCH: COMPARATOR x NON-INVERTING CHANNEL SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PCH<2:0>		
bit 7							bit 0

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

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PCH<2:0>:** Comparator Non-Inverting Input Channel Select bits

111 = Vss
110 = FVR_Buffer2
101 = DAC_Output
100 = PCH not connected
011 = PCH not connected
010 = PCH not connected
001 = CxIN1+
000 = CxIN0+

REGISTER 38-5: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	C2OUT	C1OUT
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 7-2 **Unimplemented:** Read as '0'

bit 1 **C2OUT:** Mirror copy of C2OUT bit

bit 0 **C1OUT:** Mirror copy of C1OUT bit

TABLE 38-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CMxCON0	EN	OUT	—	POL	—	—	HYS	SYNC	648
CMxCON1	—	—	—	—	—	—	INTP	INTN	649
CMxNCH	—	—	—	—	—	NCH<2:0>			649
CMxPCH	—	—	—	—	—	PCH<2:0>			650
CMOUT	—	—	—	—	—	—	C2OUT	C1OUT	650

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

REGISTER 39-2: HLVDCON1: LOW-VOLTAGE DETECT CONTROL REGISTER 1

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

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **SEL<3:0>:** High/Low Voltage Detection Limit Selection bits
Refer to Table 44-14 for voltage detection limits.

TABLE 39-2: SUMMARY OF REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
HLVDCON0	EN	—	OUT	RDY	—	—	INTH	INTL	657
HLVDCON1	—	—	—	—	SEL<3:0>				658

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the HLVD module.

RRNCF **Rotate Right f (No Carry)**

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

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

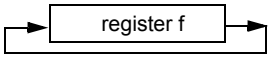
Operation: (f<n>) → dest<n – 1>,
(f<0>) → dest<7>

Status Affected: N, Z

Encoding:

0100	00da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
If 'a' is '0', the Access Bank will be selected (default), overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "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 1: RRNCF REG, 1, 0

Before Instruction
REG = 1101 0111
After Instruction
REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction
W = ?
REG = 1101 0111
After Instruction
W = 1110 1011
REG = 1101 0111

SETF **Set f**

Syntax: SETF f {,a}

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

Operation: FFh → f

Status Affected: None

Encoding:

0110	100a	ffff	ffff

Description: The contents of the specified register are set to FFh.
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 ≤ 95 (5Fh). See **Section 41.2.3 "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: SETF REG, 1

Before Instruction
REG = 5Ah
After Instruction
REG = FFh

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3F3Ah	NCO1ACCU	ACC								456
3F39h	NCO1ACCH	ACC								455
3F38h	NCO1ACCL	ACC								455
3F37h - 3F24h	—	Unimplemented								
3F23h	SMT1WIN	—	—	—	WSEL					398
3F22h	SMT1SIG	—	—	—	SSEL					399
3F21h	SMT1CLK	—	—	—	—	—	CSEL			397
3F20h	SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	396
3F1Fh	SMT1CON1	GO	REPEAT	—	—	MODE				395
3F1Eh	SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	PS		394
3F1Dh	SMT1PRU	PR								403
3F1Ch	SMT1PRH	PR								403
3F1Bh	SMT1PRL	PR								403
3F1Ah	SMT1CPWU	CPW								402
3F19h	SMT1CPWH	CPW								402
3F18h	SMT1CPWL	CPW								402
3F17h	SMT1CPRU	CPR								401
3F16h	SMT1CPRH	CPR								401
3F15h	SMT1CPRL	CPR								401
3F14h	SMT1TMRU	TMR								400
3F13h	SMT1TMRH	TMR								400
3F12h	SMT1TMRL	TMR								400
3F11h - 3F00h	—	Unimplemented								
3EFFh	ADCLK	—	—	CS						622
3EFEh	ADACT	—	—	—	ACT					635
3EFDh	ADREF	NREF				PREF				622
3EFCh	ADSTAT	OV	UTHR	LTHR	MATH	—	STAT			621
3EFBh	ADCON3	—	CALC			SOI	TMD			620
3EFAh	ADCON2	PSIS	CRS			ACLR	MODE			619
3EF9h	ADCON1	PPOL	IPEN	GPOL	—	—	—	—	DSEN	618
3EF8h	ADCON0	ON	CONT	—	CS	FM		—	GO	617
3EF7h	ADPREH	—	—	—	PRE					624
3EF6h	ADPREL	PRE								624
3EF5h	ADCAP	—	—	—	ADCAP					626
3EF4h	ADACQH	—	—	—	ACQ					625
3EF3h	ADACQL	ACQ								625
3EF2h	—	Unimplemented								
3EF1h	ADPCH	—	—	ADPCH						623
3EF0h	ADRESH	RES								628
3EEFh	ADRESL	RES								628
3EEEh	ADPREVH	PREV								630
3EEDh	ADPREVL	PREV								630
3EECh	ADRPT	RPT								626
3EEBh	ADCNT	CNT								627
3EEAh	ADACCU	ACC								631
3EE9h	ADACCH	ACC								631

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

43.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

43.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

43.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

43.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

TABLE 44-4: POWER-DOWN CURRENT (I_{PD})(1,2)

PIC18LF26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated)					
PIC18F26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated) VREGPM = 1					
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								V_{DD}	Note
D200	I_{PD}	I_{PD} Base	—	0.07	2	6	µA	3.0V	
D200 D200A	I_{PD}	I_{PD} Base	—	0.4	4	8	µA	3.0V	
			—	20	38	42	µA	3.0V	VREGPM = 0
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	0.9	3.2	7	µA	3.0V	
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	1.1	3.2	9	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	0.75	5	9	µA	3.0V	LP mode
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	1.0	6.5	10	µA	3.0V	LP mode
D203	I_{PD_FVR}	FVR	—	45	74	75	µA	3.0V	FVRCON = 0x81 or 0x84
D203	I_{PD_FVR}	FVR	—	40	70	76	µA	3.0V	FVRCON = 0x81 or 0x84
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	14	18	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	15	18	µA	3.0V	
D205	I_{PD_LPBOR}	Low-Power Brown-out Reset (LPBOR)	—	0.2	3	6	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.5	14.8	18	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.7	14.2	17	µA	3.0V	
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	2	6	µA	3.0V	ADC not converting (4)
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	4	8	µA	3.0V	ADC not converting (4)
D208	I_{PD_CMP}	Comparator	—	33	49	50	µA	3.0V	
D208	I_{PD_CMP}	Comparator	—	30	49	50	µA	3.0V	

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is FRC.

TABLE 44-27: TEMPERATURE INDICATOR REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)								
Param No.	**Symbol**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
TS01*	TACQMIN	Minimum ADC Acquisition Time Delay		—	25	—	µs	
TS02*	MV	Voltage Sensitivity	High Range	—	-3.684	—	mV/°C	TSRNG = 1
			Low Range	—	-2.456	—	mV/°C	TSRNG = 0

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

APPENDIX A: REVISION HISTORY

Revision A (6/2017)

Initial release of the document.

Revision B (12/2017)

Standard operating conditions updated in Section 44.0, Electrical Specifications. Other minor corrections.