

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	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
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)	2.3V ~ 5.5V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic16f18855-i-ss

PIC16(L)F18855/75

TABLE 1-3: PIC16F18875 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC0/ANC0/T1CKI ⁽¹⁾ /T3CKI ⁽¹⁾ /T3G ⁽¹⁾ /SMTWIN1 ⁽¹⁾ /IOCC0/SOSCO	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	ANC0	AN	—	ADC Channel C0 input.
	T1CKI ⁽¹⁾	TTL/ST	—	Timer1 external digital clock input.
	T3CKI ⁽¹⁾	TTL/ST	—	Timer3 external digital clock input.
	T3G ⁽¹⁾	TTL/ST	—	Timer3 gate input.
	SMTWIN1 ⁽¹⁾	TTL/ST	—	Signal Measurement Timer1 (SMT1) input.
	IOCC0	TTL/ST	—	Interrupt-on-change input.
RC1/ANC1/SMTSIG1 ⁽¹⁾ /CCP2 ⁽¹⁾ /IOCC1/SOSCI	RC1	TTL/ST	CMOS/OD	General purpose I/O.
	ANC1	AN	—	ADC Channel C1 input.
	SMTSIG1 ⁽¹⁾	TTL/ST	—	Signal Measurement Timer1 (SMT1) signal input.
	CCP2 ⁽¹⁾	TTL/ST	CMOS/OD	Capture/compare/PWM2 (default input location for capture function).
	IOCC1	TTL/ST	—	Interrupt-on-change input.
	SOSCI	AN	—	32.768 kHz secondary oscillator crystal driver input.
RC2/ANC2/T5CKI ⁽¹⁾ /CCP1 ⁽¹⁾ /IOCC2	RC2	TTL/ST	CMOS/OD	General purpose I/O.
	ANC2	AN	—	ADC Channel C2 input.
	T5CKI ⁽¹⁾	TTL/ST	—	Timer5 external digital clock input.
	CCP1 ⁽¹⁾	TTL/ST	CMOS/OD	Capture/compare/PWM1 (default input location for capture function).
	IOCC2	TTL/ST	—	Interrupt-on-change input.
RC3/ANC3/SCL1 ^(3,4) /SCK1 ⁽¹⁾ /T2IN ⁽¹⁾ /IOCC3	RC3	TTL/ST	CMOS/OD	General purpose I/O.
	ANC3	AN	—	ADC Channel C3 input.
	SCL1 ^(3,4)	I ² C/SMBus	OD	MSSP1 I ² C clock input/output.
	SCK1 ⁽¹⁾	TTL/ST	CMOS/OD	MSSP1 SPI clock input/output (default input location, SCK1 is a PPS remappable input and output).
	T2IN ⁽¹⁾	TTL/ST	—	Timer2 external input.
	IOCC3	TTL/ST	—	Interrupt-on-change input.
RC4/ANC4/SDA1 ^(3,4) /SDI1 ⁽¹⁾ /IOCC4	RC4	TTL/ST	CMOS/OD	General purpose I/O.
	ANC4	AN	—	ADC Channel C4 input.
	SDA1 ^(3,4)	I ² C/SMBus	OD	MSSP1 I ² C serial data input/output.
	SDI1 ⁽¹⁾	TTL/ST	—	MSSP1 SPI serial data input.
	IOCC4	TTL/ST	—	Interrupt-on-change input.
RC5/ANC5/T4IN ⁽¹⁾ /IOCC5	RC5	TTL/ST	CMOS/OD	General purpose I/O.
	ANC5	AN	—	ADC Channel C5 input.
	T4IN ⁽¹⁾	TTL/ST	—	Timer4 external input.
	IOCC5	TTL/ST	—	Interrupt-on-change input.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I²C = Schmitt Trigger input with I²CHV=
High Voltage XTAL= Crystal levels

- 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. Refer to Table 13-1 for details on which PORT pins may be used for this signal.
 - 2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 13-3.
 - 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 are configured for I²C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 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 I²C specific or SMBus input buffer thresholds.

TABLE 3-13: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18855	PIC16(L)F18875	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 29 (Continued)													
EA2h	CCP2PPS			—	—	—			CCP2PPS<4:0>			---1 0001	---u uuuu
EA3h	CCP3PPS			—	—	—			CCP3PPS<4:0>			---0 1101	---u uuuu
EA4h	CCP4PPS			—	—	—			CCP4PPS<4:0>			---0 1000	---u uuuu
EA5h	CCP5PPS			—	—				CCP5PPS<5:0>			--00 0100	--uu uuuu
EA6h	—	—							Unimplemented			—	—
EA7h	—	—							Unimplemented			—	—
EA8h	—	—							Unimplemented			—	—
EA9h	SMT1WINPPS			—	—	—			SMT1WINPPS<4:0>			---1 0000	---u uuuu
EAAh	SMT1SIGPPS			—	—	—			SMT1SIGPPS<4:0>			---1 0001	---u uuuu
EABh	SMT2WINPPS			—	—	—			SMT2WINPPS<4:0>			---1 0000	---u uuuu
EACH	SMT2SIGPPS			—	—	—			SMT2SIGPPS<4:0>			---1 0001	---u uuuu
EADh	—	—							Unimplemented			—	—
EAEh	—	—							Unimplemented			—	—
EAfh	—	—							Unimplemented			—	—
EB0h	—	—							Unimplemented			—	—
EB1h	CWG1PPS			—	—	—			CWG1PPS<4:0>			---0 1000	---u uuuu
EB2h	CWG2PPS			—	—	—			CWG2PPS<4:0>			---0 1001	---u uuuu
EB3h	CWG3PPS			—	—	—			CWG3PPS<4:0>			---0 1010	---u uuuu
EB4h	—	—							Unimplemented			—	—
EB5h	—	—							Unimplemented			—	—
EB6h	—	—							Unimplemented			—	—
EB7h	—	—							Unimplemented			—	—
EB8h	MDCARLPPS			—	—	—			MDCARLPPS<4:0>			---0 0011	---u uuuu
EB9h	MDCARHPPS			—	—	—			MDCARHPPS<4:0>			---0 0100	---u uuuu

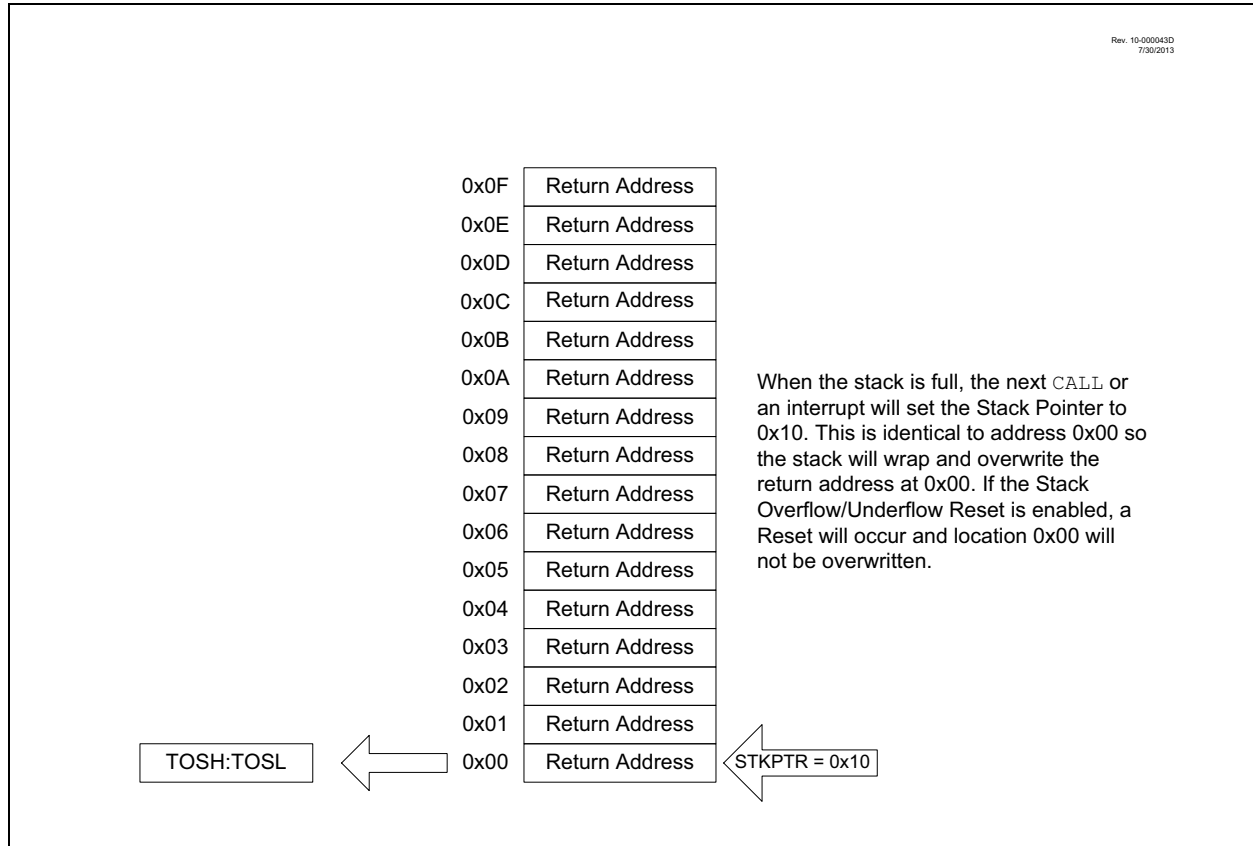
Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Register present on PIC16F18855/75 devices only.

Note 2: Unimplemented, read as '1'.

PIC16(L)F18855/75

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



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.

3.5 Indirect Addressing

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

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

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

PIC16(L)F18855/75

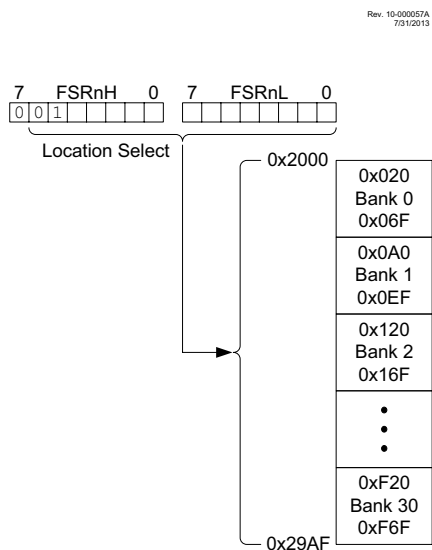
3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

FIGURE 3-10: TRADITIONAL DATA MEMORY MAP



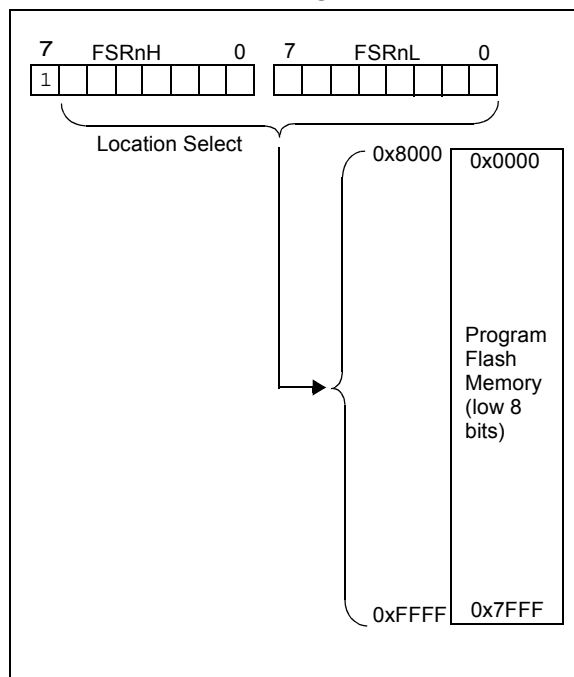
3.5.3 DATA EEPROM MEMORY

The EEPROM memory can be read or written through the NVMCONx/NVMADRx/NVMDATx register interface (see section **Section 10.2 “Data EEPROM Memory”**). However, to make access to the EEPROM memory easier, read-only access to the EEPROM contents are also available through indirect addressing by an FSR. When the MSB of the FSR (ex: FSRxH) is set to 0x70, the lower 8-bit address value (in FSRxL) determines the EEPROM location that may be read from (through the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000-0x70FF. Writing to the EEPROM cannot be accomplished via the FSR/INDF interface. Reads from the EEPROM through the FSR/INDF interface will require one additional instruction cycle to complete.

3.5.4 PROGRAM FLASH MEMORY

To make constant data access easier, the entire Program Flash Memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the Program Flash Memory cannot be accomplished via the FSR/INDF interface. All instructions that access Program Flash Memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-11: PROGRAM FLASH MEMORY MAP



PIC16(L)F18855/75

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PEx registers)

The PIR1, PIR2, PIR3 and PIR4 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See “**Section 7.5 “Automatic Context Saving”**”)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1: Individual interrupt flag bits are set, regardless of the state of any other enable bits.

2: All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

PIC16(L)F18855/75

REGISTER 7-20: PIR8: PERIPHERAL INTERRUPT REQUEST REGISTER 8

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

HS = Hardware set

bit 7-6 **Unimplemented:** Read as '0'.

bit 5 **SMT2PWAIF:** SMT2 Pulse Width Acquisition Interrupt Flag bit

1 = Interrupt is pending

0 = Interrupt is not pending

bit 4 **SMT2PRAIF:** SMT2 Period Acquisition Interrupt Flag bit

1 = Interrupt is pending

0 = Interrupt is not pending

bit 3 **SMT2IF:** SMT2 Overflow Interrupt Flag bit

1 = An SMT overflow event has occurred (must be cleared in software)

0 = No overflow event detected

bit 2 **SMT1PWAIF:** SMT1 Pulse Width Acquisition Interrupt Flag bit

1 = Interrupt is pending

0 = Interrupt is not pending

bit 1 **SMT1PRAIF:** SMT1 Period Acquisition Interrupt Flag bit

1 = Interrupt is pending

0 = Interrupt is not pending

bit 0 **SMT1IF:** SMT1 Overflow Interrupt Flag bit

1 = An SMT overflow event has occurred (must be cleared in software)

0 = No overflow event detected

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

PIC16(L)F18855/75

10.4.3 NVMREG WRITE TO EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address + F000h) into the NVMADRH:NVMADRL register pair (Table 10-2).
3. Perform the unlock sequence as described in **Section 10.4.2 “NVM Unlock Sequence”**.

A single EEPROM word is written with NVMDATA. The operation includes an implicit erase cycle for that word (it is not necessary to set the FREE bit), and requires many instruction cycles to finish. CPU execution continues in parallel and, when complete, WR is cleared by hardware, NVMIF is set, and an interrupt will occur if NVMIE is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged.

Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will continue to run to completion.

10.4.4 NVMREG ERASE OF PFM

Before writing to PFM, the word(s) to be written must be erased or previously unwritten. PFM can only be erased one row at a time. No automatic erase occurs upon the initiation of the write to PFM.

To erase a PFM row:

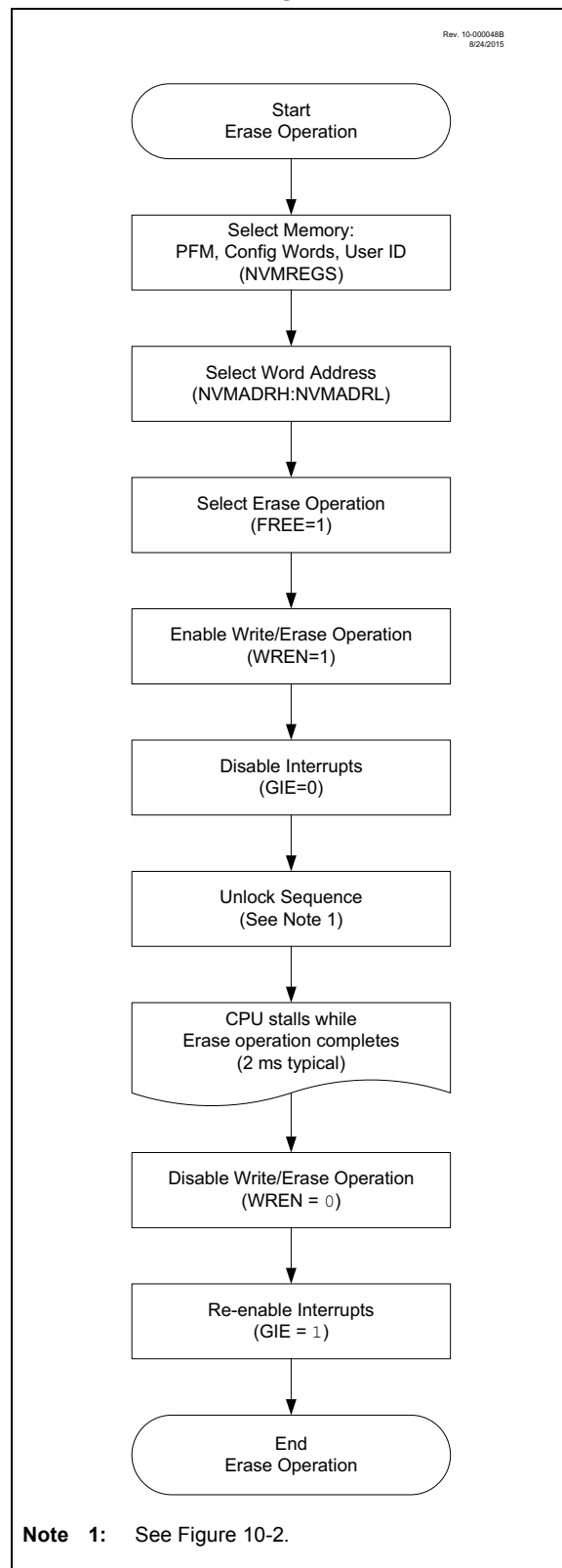
1. Clear the NVMREGS bit of the NVMCON1 register to erase PFM locations, or set the NVMREGS bit to erase User ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 10-2).
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in **Section 10.4.2 “NVM Unlock Sequence”**.

If the PFM address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing PFM, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

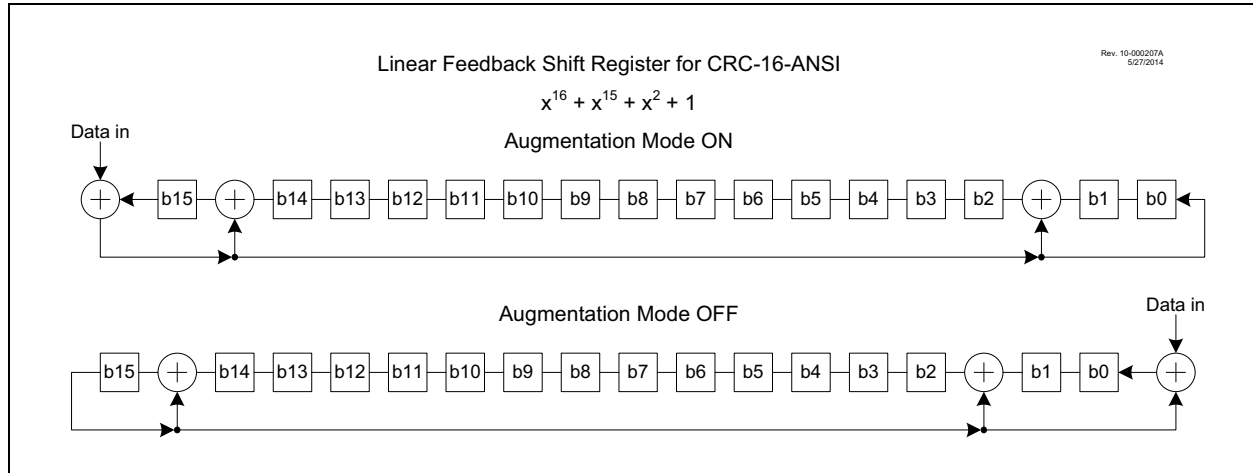
Write latch data is not affected by erase operations, and WREN will remain unchanged.

FIGURE 10-3: NVM ERASE FLOWCHART



PIC16(L)F18855/75

FIGURE 11-1: CRC LFSR EXAMPLE



11.4 CRC Data Sources

Data can be input to the CRC module in two ways:

- User data using the CRCDAT registers
- Flash using the Program Memory Scanner

To set the number of bits of data, up to 16 bits, the DLEN bits of CRC CON1 must be set accordingly. Only data bits in CRCDATA registers up to DLEN will be used, other data bits in CRCDATA registers will be ignored.

Data is moved into the CRC SHIFT as an intermediate to calculate the check value located in the CRC ACC registers.

The SHIFTM bit is used to determine the bit order of the data being shifted into the accumulator. If SHIFTM is not set, the data will be shifted in MSb first. The value of DLEN will determine the MSb. If SHIFTM bit is set, the data will be shifted into the accumulator in reversed order, LSb first.

The CRC module can be seeded with an initial value by setting the CRC ACC<15:0> registers to the appropriate value before beginning the CRC.

11.4.1 CRC FROM USER DATA

To use the CRC module on data input from the user, the user must write the data to the CRCDAT registers. The data from the CRCDAT registers will be latched into the shift registers on any write to the CRCDATL register.

11.4.2 CRC FROM FLASH

To use the CRC module on data located in Flash memory, the user can initialize the Program Memory Scanner as defined in **Section 11.8, Program Memory Scan Configuration**.

11.5 CRC Check Value

The CRC check value will be located in the CRC ACC registers after the CRC calculation has finished. The check value will depend on two mode settings of the CRC CON: ACCM and SHIFTM.

If the ACCM bit is set, the CRC module will augment the data with a number of zeros equal to the length of the polynomial to find the final check value. If the ACCM bit is not set, the CRC will stop at the end of the data. A number of zeros equal to the length of the polynomial can then be entered to find the same check value as augmented mode, alternatively the expected check value can be entered at this point to make the final result equal to 0.

A final XOR value may be needed with the check value to find the desired CRC result

11.6 CRC Interrupt

The CRC will generate an interrupt when the BUSY bit transitions from '1' to '0'. The CRCIF interrupt flag bit of the PIR6 register is set every time the BUSY bit transitions, regardless of whether or not the CRC interrupt is enabled. The CRCIF bit can only be cleared in software. The CRC interrupt enable is the CRCIE bit of the PIE6 register.

PIC16(L)F18855/75

TABLE 13-3: PPS OUTPUT SIGNAL ROUTING OPTIONS

Output Signal Name	RxyPPS Register Value	Remappable to Pins of PORTx							
		PIC16F18855			PIC16F18875				
		PORTA	PORTB	PORTC	PORTA	PORTB	PORTC	PORTD	PORTE
ADGRDG	0x25	•		•	•		•		
ADGRDA	0x24	•		•	•		•		
CWG3D	0x23	•		•	•			•	
CWG3C	0x22	•		•	•			•	
CWG3B	0x21	•		•	•				•
CWG3A	0x20		•	•		•	•		
CWG2D	0x1F		•	•		•		•	
CWG2C	0x1E		•	•		•		•	
CWG2B	0x1D		•	•		•		•	
CWG2A	0x1C		•	•		•	•		
DSM	0x1B	•		•	•			•	
CLKR	0x1A		•	•		•	•		
NCO	0x19	•		•	•			•	
TMR0	0x18		•	•		•	•		
SDO2/SDA2	0x17		•	•		•		•	
SCK2/SCL2	0x16		•	•		•		•	
SD01/SDA1	0x15		•	•		•	•		
SCK1/SCL1	0x14		•	•		•	•		
C2OUT	0x13	•		•	•				•
C1OUT	0x12	•		•	•			•	
DT	0x11		•	•		•	•		
TX/CK	0x10		•	•		•	•		
PWM7OUT	0x0F	•		•	•		•		
PWM6OUT	0x0E	•		•	•			•	
CCP5	0x0D	•		•	•				•
CCP4	0x0C		•	•		•		•	
CCP3	0x0B		•	•		•		•	
CCP2	0x0A		•	•		•	•		
CCP1	0x09		•	•		•	•		
CWG1D	0x08		•	•		•		•	
CWG1C	0x07		•	•		•		•	
CWG1B	0x06		•	•		•		•	
CWG1A	0x05		•	•		•	•		
CLC4OUT	0x04		•	•		•		•	
CLC3OUT	0x03		•	•		•		•	
CLC2OUT	0x02	•		•	•		•		
CLC1OUT	0x01	•		•	•		•		

Note: When RxyPPS = 0x00, port pin Rxy output value is controlled by the respective LATxy bit.

18.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See Comparator Specifications in Table 37-14 for more information.

18.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 28.7 “Timer Gate”** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

18.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 18-2) and the Timer1 Block Diagram (Figure 28-1) for more information.

18.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Note: Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

18.6 Comparator Positive Input Selection

Configuring the CxPCH<2:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN0+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- Vss (Ground)

See **Section 16.0 “Fixed Voltage Reference (FVR)”** for more information on the Fixed Voltage Reference module.

See **Section 25.0 “5-Bit Digital-to-Analog Converter (DAC1) Module”** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

18.7 Comparator Negative Input Selection

The CxNCH<2:0> bits of the CMxCON1 register direct an analog input pin and internal reference voltage or analog ground to the inverting input of the comparator:

- CxIN- pin
- FVR (Fixed Voltage Reference)
- Analog Ground

Some inverting input selections share a pin with the operational amplifier output function. Enabling both functions at the same time will direct the operational amplifier output to the comparator inverting input.

Note: To use CxINy+ and CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

PIC16(L)F18855/75

18.12 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ON	OUT	—	POL	—	—	HYS	SYNC
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 **ON:** Comparator Enable bit
 1 = Comparator is enabled
 0 = Comparator is disabled and consumes no active power
- bit 6 **OUT:** Comparator Output bit
 If CxPOL = 1 (inverted polarity):
 1 = CxVP < CxVN
 0 = CxVP > CxVN
 If CxPOL = 0 (non-inverted polarity):
 1 = CxVP > CxVN
 0 = CxVP < CxVN
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **POL:** Comparator Output Polarity Select bit
 1 = Comparator output is inverted
 0 = Comparator output is not inverted
- bit 3-2 **Unimplemented:** Read as '0'
- bit 1 **HYS:** Comparator Hysteresis Enable bit
 1 = Comparator hysteresis enabled
 0 = Comparator hysteresis disabled
- bit 0 **SYNC:** Comparator Output Synchronous Mode bit
 1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source.
 Output updated on the falling edge of Timer1 clock source.
 0 = Comparator output to Timer1 and I/O pin is asynchronous

PIC16(L)F18855/75

REGISTER 18-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
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 **Unimplemented:** Read as '0'

bit 1 **INTP:** Comparator Interrupt on Positive-Going Edge Enable bits

- 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 bits

- 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

PIC16(L)F18855/75

22.1.2 DATA GATING

Outputs from the input multiplexers are directed to the desired logic function input through the data gating stage. Each data gate can direct any combination of the four selected inputs.

Note: Data gating is undefined at power-up.

The gate stage is more than just signal direction. The gate can be configured to direct each input signal as inverted or non-inverted data. Directed signals are ANDed together in each gate. The output of each gate can be inverted before going on to the logic function stage.

The gating is in essence a 1-to-4 input AND/NAND/OR/NOR gate. When every input is inverted and the output is inverted, the gate is an OR of all enabled data inputs. When the inputs and output are not inverted, the gate is an AND of all enabled inputs.

Table 22-3 summarizes the basic logic that can be obtained in gate 1 by using the gate logic select bits. The table shows the logic of four input variables, but each gate can be configured to use less than four. If no inputs are selected, the output will be zero or one, depending on the gate output polarity bit.

TABLE 22-3: DATA GATING LOGIC

CLCxGLSy	LCxGyPOL	Gate Logic
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	Logic 0
0x00	1	Logic 1

It is possible (but not recommended) to select both the true and negated values of an input. When this is done, the gate output is zero, regardless of the other inputs, but may emit logic glitches (transient-induced pulses). If the output of the channel must be zero or one, the recommended method is to set all gate bits to zero and use the gate polarity bit to set the desired level.

Data gating is configured with the logic gate select registers as follows:

- Gate 1: CLCxGLS0 (Register 22-7)
- Gate 2: CLCxGLS1 (Register 22-8)
- Gate 3: CLCxGLS2 (Register 22-9)
- Gate 4: CLCxGLS3 (Register 22-10)

Register number suffixes are different than the gate numbers because other variations of this module have multiple gate selections in the same register.

Data gating is indicated in the right side of Figure 22-2. Only one gate is shown in detail. The remaining three gates are configured identically with the exception that the data enables correspond to the enables for that gate.

22.1.3 LOGIC FUNCTION

There are eight available logic functions including:

- AND-OR
- OR-XOR
- AND
- S-R Latch
- D Flip-Flop with Set and Reset
- D Flip-Flop with Reset
- J-K Flip-Flop with Reset
- Transparent Latch with Set and Reset

Logic functions are shown in Figure 22-2. Each logic function has four inputs and one output. The four inputs are the four data gate outputs of the previous stage. The output is fed to the inversion stage and from there to other peripherals, an output pin, and back to the CLCx itself.

22.1.4 OUTPUT POLARITY

The last stage in the Configurable Logic Cell is the output polarity. Setting the LCxPOL bit of the CLCxPOL register inverts the output signal from the logic stage. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

PIC16(L)F18855/75

30.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

The Capture/Compare/PWM modules available are shown in Table 30-1.

TABLE 30-1: AVAILABLE CCP MODULES

Device	CCP1	CCP2	CCP3	CCP4	CCP5
PIC16(L)F18855/75	•	•	•	•	•

The Capture and Compare functions are identical for all CCP modules.

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

PIC16(L)F18855/75

31.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULES

31.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

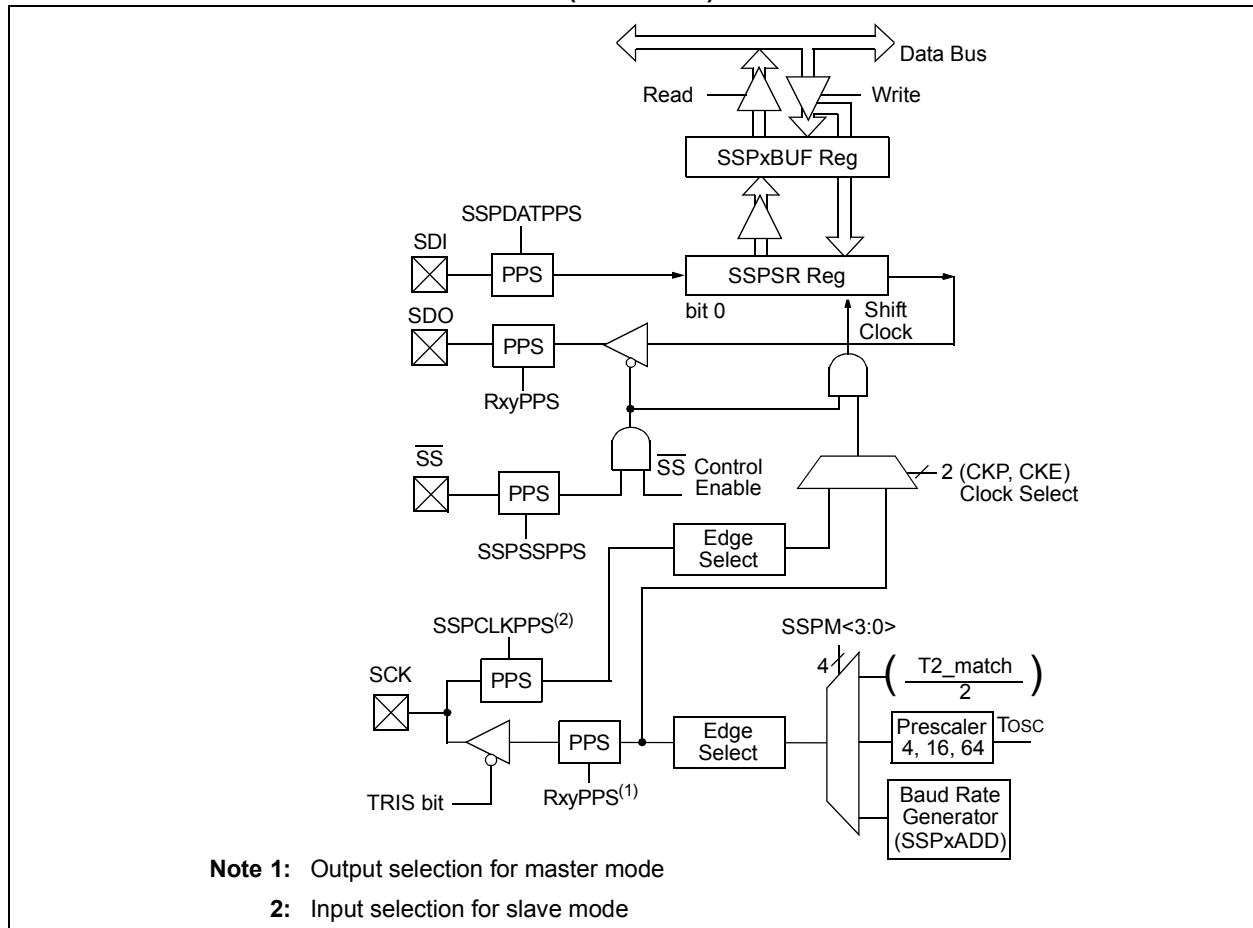
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 31-1 is a block diagram of the SPI interface module.

FIGURE 31-1: MSSP BLOCK DIAGRAM (SPI MODE)



31.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select (\overline{SS})

Figure 31-1 shows the block diagram of the MSSP module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 31-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 31-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

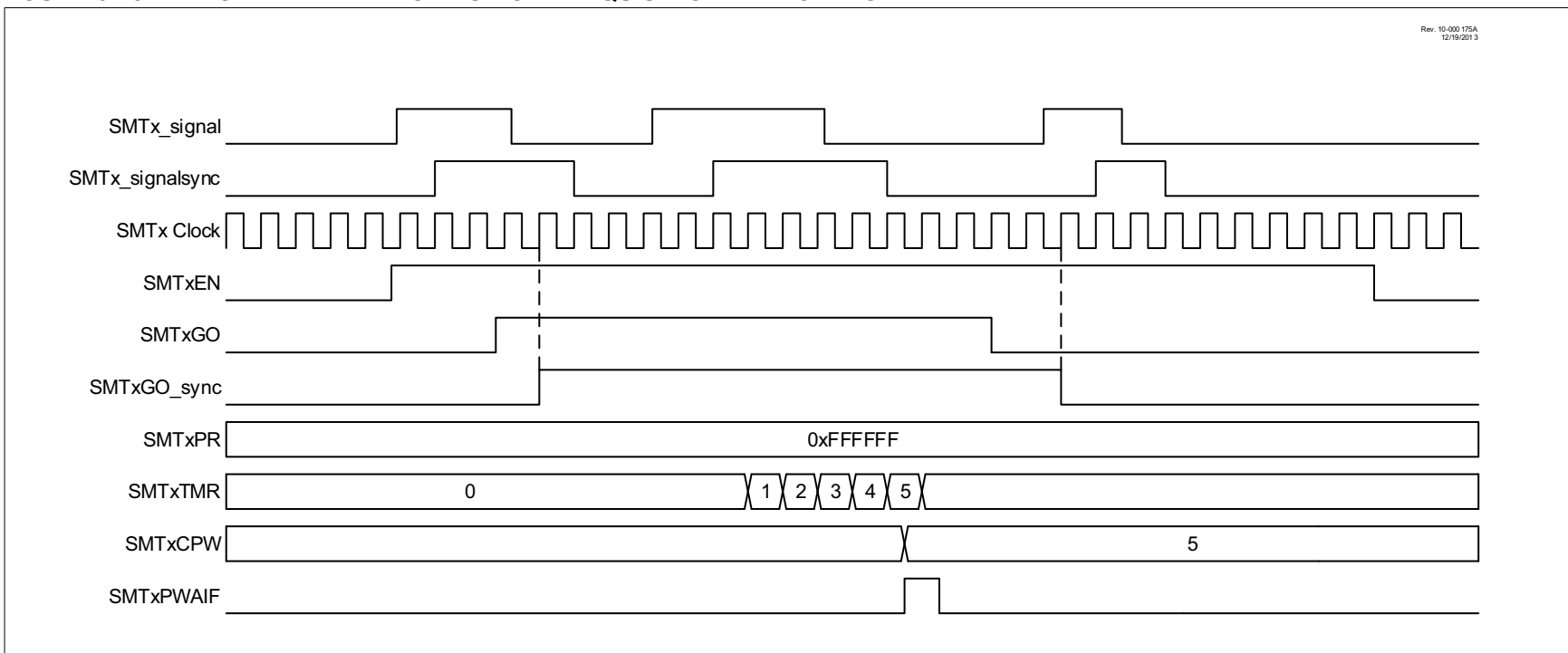
Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

FIGURE 32-5: GATED TIMER MODE SINGLE ACQUISITION TIMING DIAGRAM



PIC16(L)F18855/75

REGISTER 32-16: SMTxPRL: SMT PERIOD REGISTER – LOW BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<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 **SMTxPR<7:0>**: Significant bits of the SMT Timer Value for Period Match – Low Byte

REGISTER 32-17: SMTxPRH: SMT PERIOD REGISTER – HIGH BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<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 **SMTxPR<15:8>**: Significant bits of the SMT Timer Value for Period Match – High Byte

REGISTER 32-18: SMTxPRU: SMT PERIOD REGISTER – UPPER BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<23:16>							
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 **SMTxPR<23:16>**: Significant bits of the SMT Timer Value for Period Match – Upper Byte

33.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V_{OH} Mark state which represents a '1' data bit, and a V_{OL} Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 33-3 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

33.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 33-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

33.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TX1STA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TX1STA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RC1STA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note: The TXIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

33.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one T_{cy} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

33.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUD1CON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 33.4.1.2 "Clock Polarity"**.

33.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR3 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

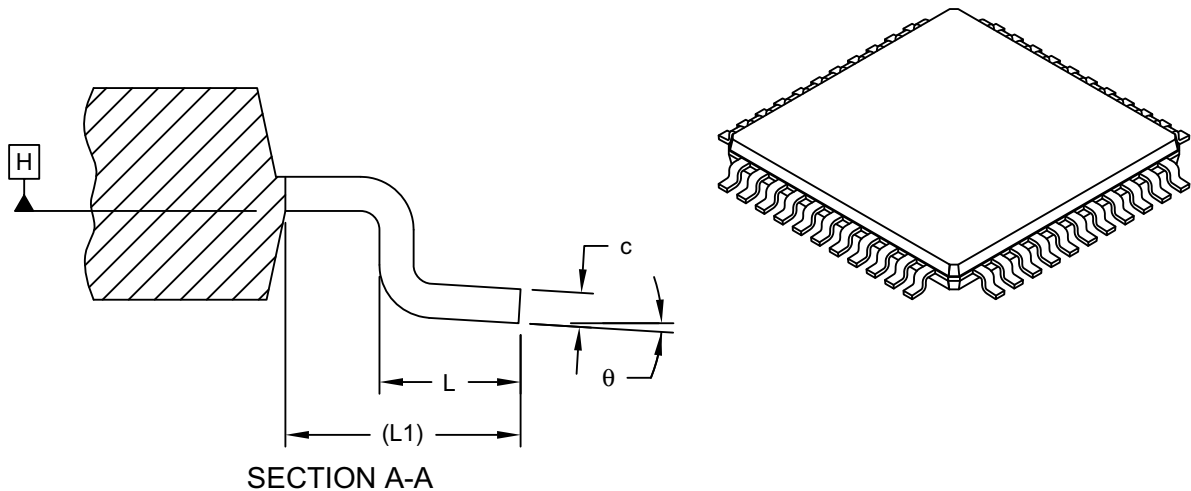
The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE3 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

PIC16(L)F18855/75

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1.0 mm Body [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	e	0.80 BSC		
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Width	E	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Width	b	0.30	0.37	0.45
Lead Thickness	c	0.09	-	0.20
Lead Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	θ	0°	3.5°	7°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Exact shape of each corner is optional.
- Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.