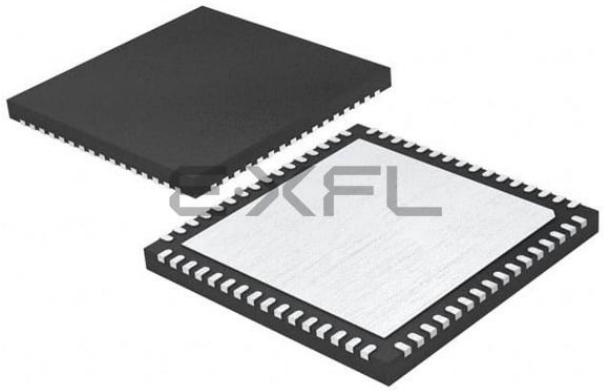


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	54
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 30x10b
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/pic16lf1527-i-mr

3.5 Special Function Register

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

3.5.1 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh).

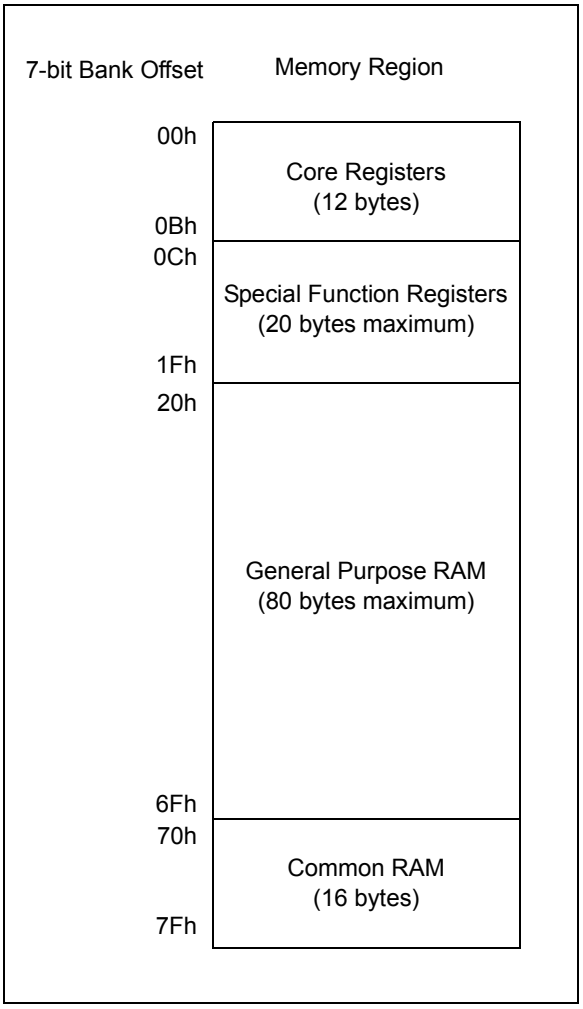
3.5.1.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 3.8.2 "Linear Data Memory"** for more information.

3.5.2 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

FIGURE 3-3: BANKED MEMORY PARTITIONING



3.5.3 DEVICE MEMORY MAPS

The memory maps for PIC16(L)F1526/7 are shown in Table 3-3.

TABLE 3-3: PIC16(L)F1526/7 MEMORY MAP (CONTINUED)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	Core Registers (Table 3-2)	480h	Core Registers (Table 3-2)	500h	Core Registers (Table 3-2)	580h	Core Registers (Table 3-2)	600h	Core Registers (Table 3-2)	680h	Core Registers (Table 3-2)	700h	Core Registers (Table 3-2)	780h	Core Registers (Table 3-2)
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh		78Bh	
40Ch	ANSELF	48Ch	—	50Ch	Unimplemented Read as '0'	58Ch	—	60Ch	—	68Ch	Unimplemented Read as '0'	70Ch	Unimplemented Read as '0'	78Ch	Unimplemented Read as '0'
40Dh	ANSELG	48Dh	WPUG			58Dh	—	60Dh	—						
40Eh	—	48Eh	—			58Eh	—	60Eh	—						
40Fh	—	48Fh	—			58Fh	—	60Fh	—						
410h	—	490h	—			590h	—	610h	—						
411h	TMR3L	491h	RC2REG			591h	—	611h	CCPR6L						
412h	TMR3H	492h	TX2REG			592h	—	612h	CCPR6H						
413h	T3CON	493h	SP2BRG			593h	—	613h	CCP6CON						
414h	T3GCON	494h	SP2BRGH			594h	—	614h	CCPR7L						
415h	TMR4	495h	RC2STA			595h	TMR8	615h	CCPR7H						
416h	PR4	496h	TX2STA			596h	PR8	616h	CCP7CON						
417h	T4CON	497h	BAUD2CON			597h	T8CON	617h	CCPR8L						
418h	TMR5H	498h	—			598h	—	618h	CCPR8H						
419h	TMR5H	499h	—			599h	—	619h	CCP8CON						
41Ah	T5CON	49Ah	—			59Ah	—	61Ah	CCPR9L						
41Bh	T5GCON	49Bh	—			59Bh	—	61Bh	CCPR9H						
41Ch	TMR6	49Ch	—			59Ch	TMR10	61Ch	CCP9CON						
41Dh	PR6	49Dh	—			59Dh	PR10	61Dh	CCPR10L						
41Eh	T6CON	49Eh	—			59Eh	T10CON	61Eh	CCPR10H						
41Fh	—	49Fh	—	51Fh		59Fh	—	61Fh	CCP10CON	69Fh		71Fh		79Fh	
420h	General Purpose Register 80 Bytes	4A0h	General Purpose Register 32 Bytes	520h	General Purpose Register 80 Bytes(1)	5A0h	General Purpose Register 80 Bytes(1)	620h	General Purpose Register 80 Bytes(1)	6A0h	General Purpose Register 80 Bytes(1)	720h	General Purpose Register 80 Bytes(1)	7A0h	General Purpose Register 80 Bytes(1)
46Fh		4BFh													
		4C0h	General Purpose Register 48 Bytes(1)												
		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh		7EFh	
470h	Common RAM (Accesses 70h – 7Fh)	4F0h	Common RAM (Accesses 70h – 7Fh)	570h	Common RAM (Accesses 70h – 7Fh)	5F0h	Common RAM (Accesses 70h – 7Fh)	670h	Common RAM (Accesses 70h – 7Fh)	6F0h	Common RAM (Accesses 70h – 7Fh)	770h	Common RAM (Accesses 70h – 7Fh)	7F0h	Common RAM (Accesses 70h – 7Fh)
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh	

Legend: = Unimplemented data memory locations, read as '0'.

Note 1: PIC16(L)F1527 only.

TABLE 3-3: PIC16(L)F1526 MEMORY MAP (CONTINUED)

	Bank 31
F80h F8Bh	Core Registers (Table 3-2)
F8Ch FE3h	Unimplemented Read as '0'
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH
FF0h FFFh	Common RAM (Accesses 70h – 7Fh)

Legend: = Unimplemented data memory locations, read as '0'.

5.4.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<3:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

5.4.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCSTAT register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or the internal oscillator.

FIGURE 5-8: TWO-SPEED START-UP

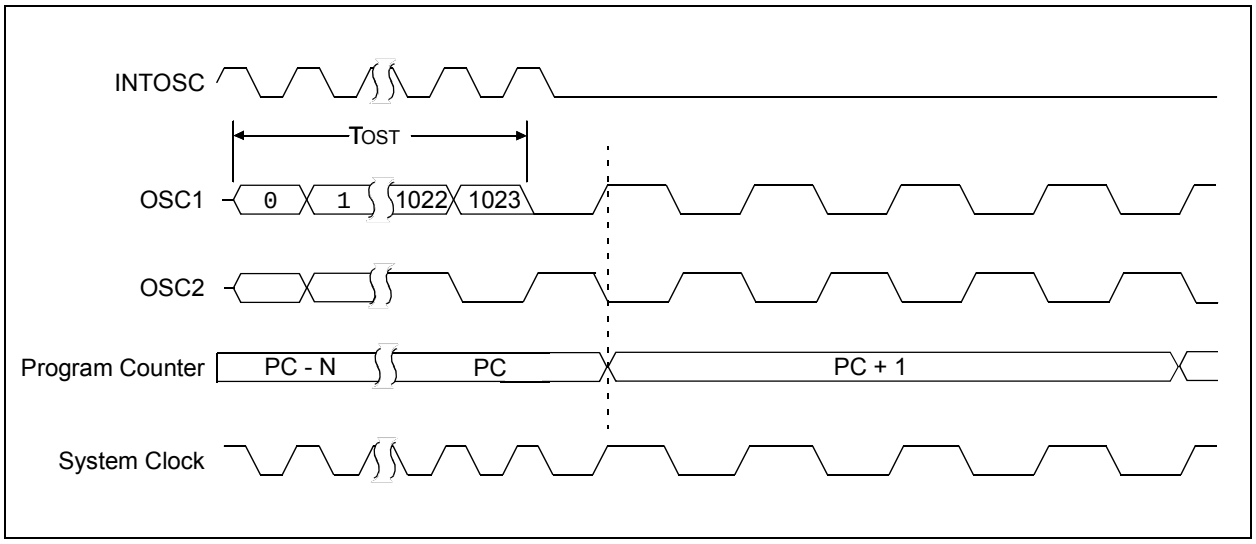
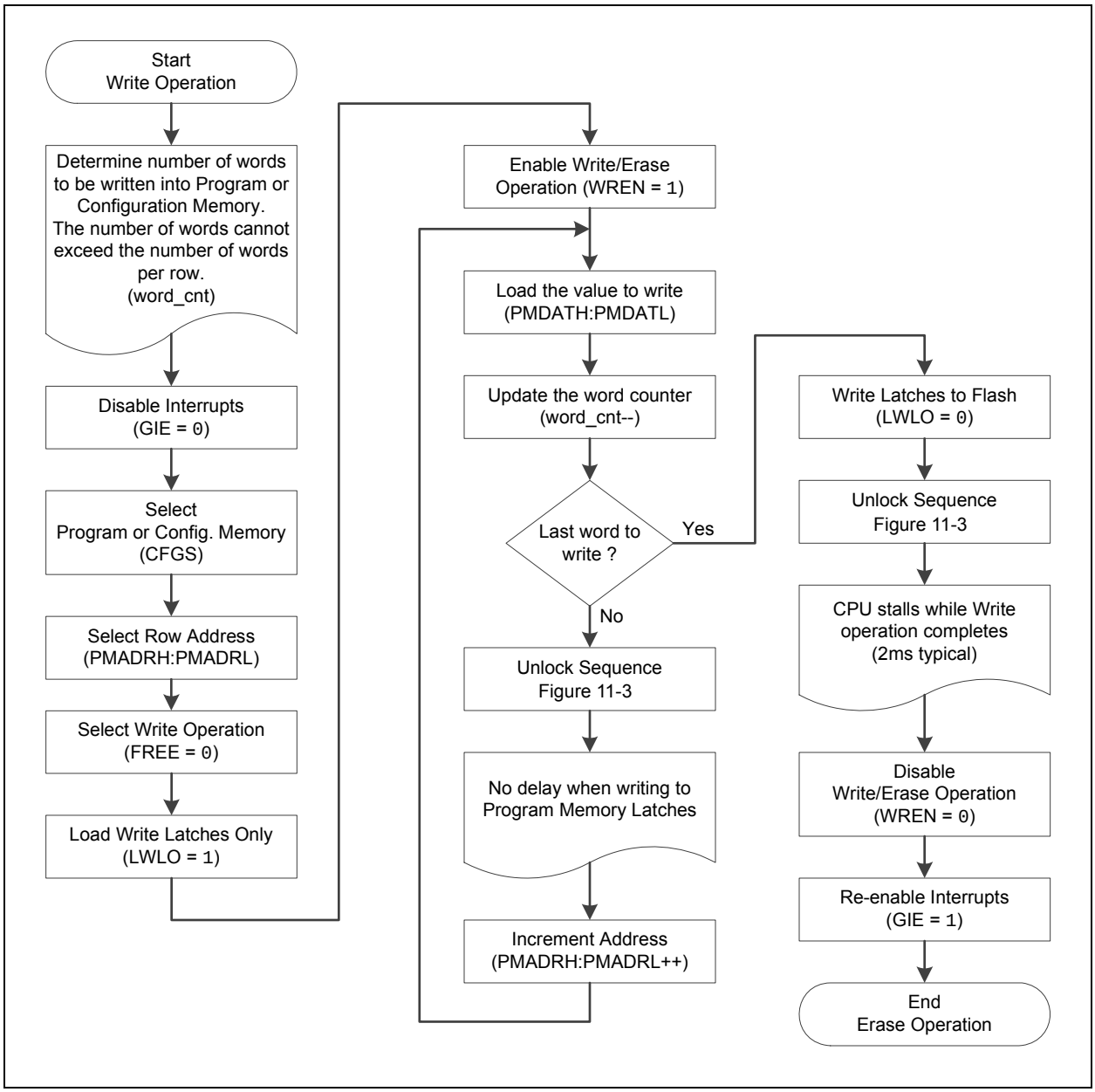


FIGURE 11-6: FLASH PROGRAM MEMORY WRITE FLOWCHART



REGISTER 11-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q[2]	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—[1]	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **CFGS:** Configuration Select bit
- 1 = Access Configuration, User ID and Device ID Registers
- 0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit[3]
- 1 = Only the addressed program memory write latch is loaded/updated on the next WR command
- 0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
- 1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
- 0 = Performs an write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write ‘1’) of the WR bit).
- 0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash program/erase operation.
  The operation is self-timed and the bit is cleared by hardware once operation is complete.
  The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash is complete and inactive.

bit 0 **RD:** Read Control bit
- 1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash read.

Note 1: Unimplemented bit, read as ‘1’.
2: The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

12.16 Register Definitions: PORTG

REGISTER 12-28: PORTG: PORTG REGISTER

U-0	U-0	R-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	RG5	RG4	RG3	RG2	RG1	RG0
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 5-0 **RG<5:0>**: PORTG I/O Pin bits(1)
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

Note 1: Writes to PORTG are actually written to corresponding LATG register. Reads from PORTG register is return of actual I/O pin values.

REGISTER 12-29: TRISG: PORTG TRI-STATE REGISTER

U-0	U-0	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—(1)	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0
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 **Unimplemented**: Read as '1'

bit 4-0 **TRISG<4:0>:** RG<4:0> Tri-State Control bits(1)
1 = PORTG pin configured as an input (tri-stated)
0 = PORTG pin configured as an output

Note 1: Unimplemented, read as '1'.

REGISTER 12-30: LATG: PORTG DATA LATCH REGISTER

U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0
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 **Unimplemented**: Read as '0'

bit 4-0 **LATG<4:0>**: PORTG Output Latch Value bits(1)

Note 1: Writes to PORTG are actually written to corresponding LATG register. Reads from PORTG register is return of actual I/O pin values.

REGISTER 12-31: ANSELG: PORTG ANALOG SELECT REGISTER

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	U-0
—	—	—	ANSG4	ANSG3	ANSG2	ANSG1	—
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	

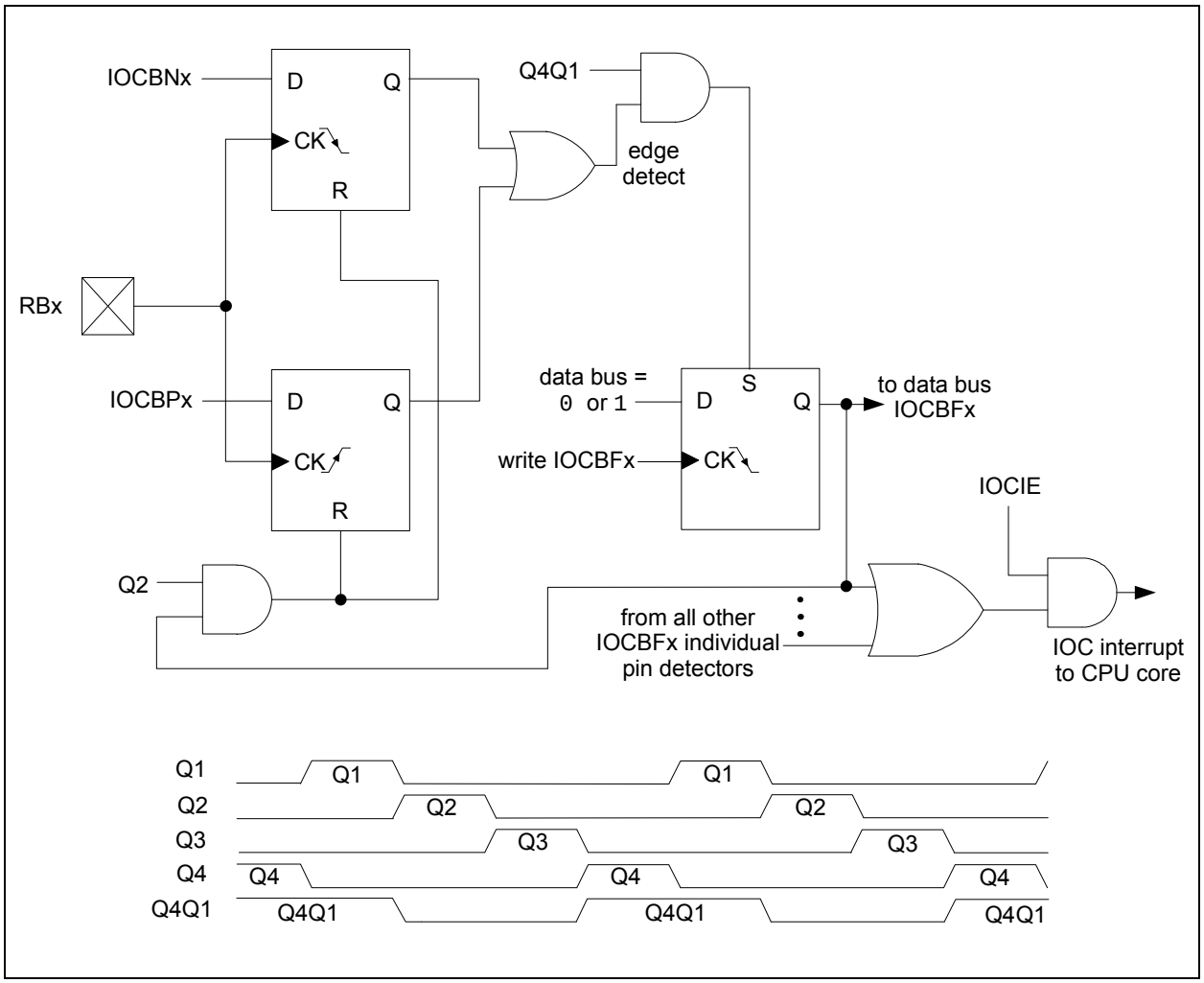
bit 7-5 **Unimplemented:** Read as '0'

bit 4-1 **ANSG<4:1>**: Analog Select between Analog or Digital Function on Pins RG<4:1>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

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

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

FIGURE 13-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM



16.2 ADC Operation

16.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 16.2.6 "ADC Conversion Procedure"**.

16.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

16.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

16.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC oscillator source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

16.2.5 SPECIAL EVENT TRIGGER

The Special Event Trigger of the CCPx module allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/DONE bit is set by hardware and the Timer1 counter resets to zero.

TABLE 16-2: SPECIAL EVENT TRIGGER

Device	CCP
PIC16(L)F1526/7	CCP10

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

Refer to **Section 20.0 "Capture/Compare/PWM Modules"** for more information.

TABLE 20-5: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/DONE	T5GVAL	T5GSS<1:0>		169
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								164*
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Register								164*
TMR6L	Holding Register for the Least Significant Byte of the 16-bit TMR6 Register								164*
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								164*
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								164*
TMR6H	Holding Register for the Most Significant Byte of the 16-bit TMR6 Register								164*
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	114

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by Compare mode.
* Page provides register information.

REGISTER 20-4: CCPTMRS2: CCP TIMER SELECTION CONTROL REGISTER 2

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	C10TSEL<1:0>		C9TSEL<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 7-4 **Unimplemented:** Read as '0'

bit 3-2 **C10TSEL<1:0>:** CCP10 Timer Selection bits

When in Capture/Compare mode:

x1 = CCP10 is based off Timer5 in Capture/Compare mode
x0 = CCP10 is based off Timer1 in Capture/Compare mode

When in PWM mode:

11 = Reserved
10 = CCP10 is based off Timer10 in PWM mode
01 = CCP10 is based off Timer8 in PWM mode
00 = CCP10 is based off Timer2 in PWM mode

bit 1-0 **C9TSEL<1:0>:** CCP9 Timer Selection bits

When in Capture/Compare mode:

x1 = CCP9 is based off Timer5 in Capture/Compare mode
x0 = CCP9 is based off Timer1 in Capture/Compare mode

When in PWM mode:

11 = Reserved
10 = CCP9 is based off Timer10 in PWM mode
01 = CCP9 is based off Timer8 in PWM mode
00 = CCP9 is based off Timer2 in PWM mode

21.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 21-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the 8th falling edge of the SCLx line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCLx.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note: SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the 9th SCLx pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCLx line to receive a Stop.

22.1.2.3 Receive Interrupts

The RCxIF interrupt flag bit of the PIR1/PIR4 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCxIF interrupts are enabled by setting the following bits:

- RCxIE interrupt enable bit of the PIE1/PIE4 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

22.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCxSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCxREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit.

22.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCxSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

22.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set, the EUSART will shift 9 bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCxREG.

22.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCxSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCxIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

FIGURE 22-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

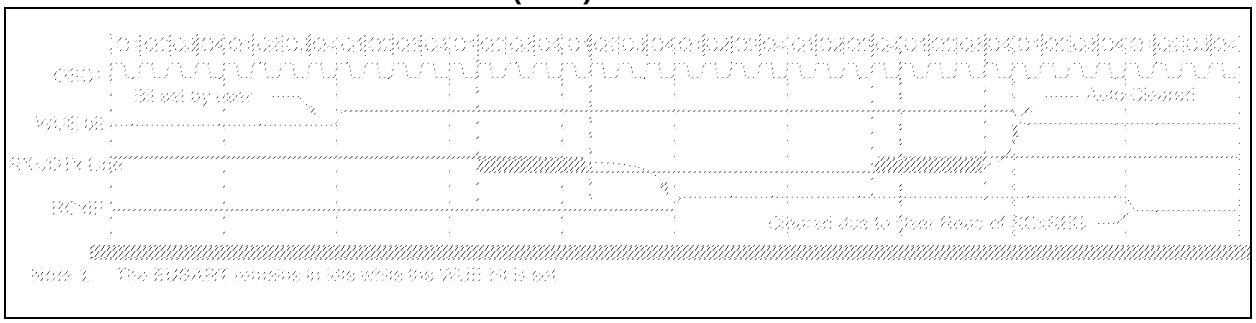
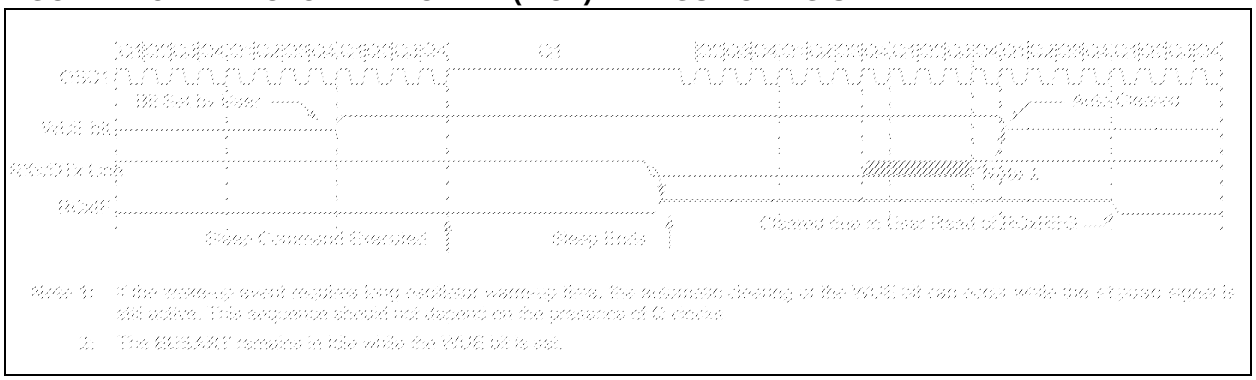


FIGURE 22-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



22.5.1.5 Synchronous Master Reception

Data is received at the RXx/DTx pin. The RXx/DTx pin output driver must be disabled by setting the corresponding TRIS bits when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RXx/DTx pin on the trailing edge of the TXx/CKx clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCxIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCxREG. The RCxIF bit remains set as long as there are un-read characters in the receive FIFO.

22.5.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TXx/CKx line. The TXx/CKx pin output driver must be disabled by setting the associated TRIS bit when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

22.5.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCxREG.

If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

22.5.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCxREG.

22.5.1.9 Synchronous Master Reception Set-up:

1. Initialize the SPxBRGH, SPxBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Set the RXx/DTx and TXx/CKx TRIS controls to '1'.
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Disable RXx/DTx and TXx/CKx output drivers by setting the corresponding TRIS bits.
4. Ensure bits CREN and SREN are clear.
5. If using interrupts, set the GIE and PEIE bits of the INTCON register and set RCxIE.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
9. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCxREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

TABLE 22-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	260
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	260
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	76
PIE1	TMR1GIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	77
PIE4	CCP10IE	CCP9IE	RC2IE	TX2IE	CCP8IE	CCP7IE	BCL2IE	SSP2IE	80
PIR1	TMR1GIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	81
PIR4	CCP10IF	CCP9IF	RC2IF	TX2IF	CCP8IF	CCP7IF	BCL2IF	SSP2IF	84
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	259
RC2STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	259
SP1BRGL	EUSART1 Baud Rate Generator, Low Byte								261*
SP1BRGH	EUSART1 Baud Rate Generator, High Byte								261*
SP2BRGL	EUSART2 Baud Rate Generator, Low Byte								261*
SP2BRGH	EUSART2 Baud Rate Generator, High Byte								261*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	120
TX1REG	EUSART1 Transmit Register								250*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	258
TX2REG	EUSART2 Transmit Register								250*
TX2STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	258

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous slave transmission.

* Page provides register information.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

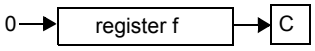
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0,destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

FIGURE 26-3: IDD TYPICAL, XT AND EXTRC OSCILLATOR, PIC16LF1526 ONLY

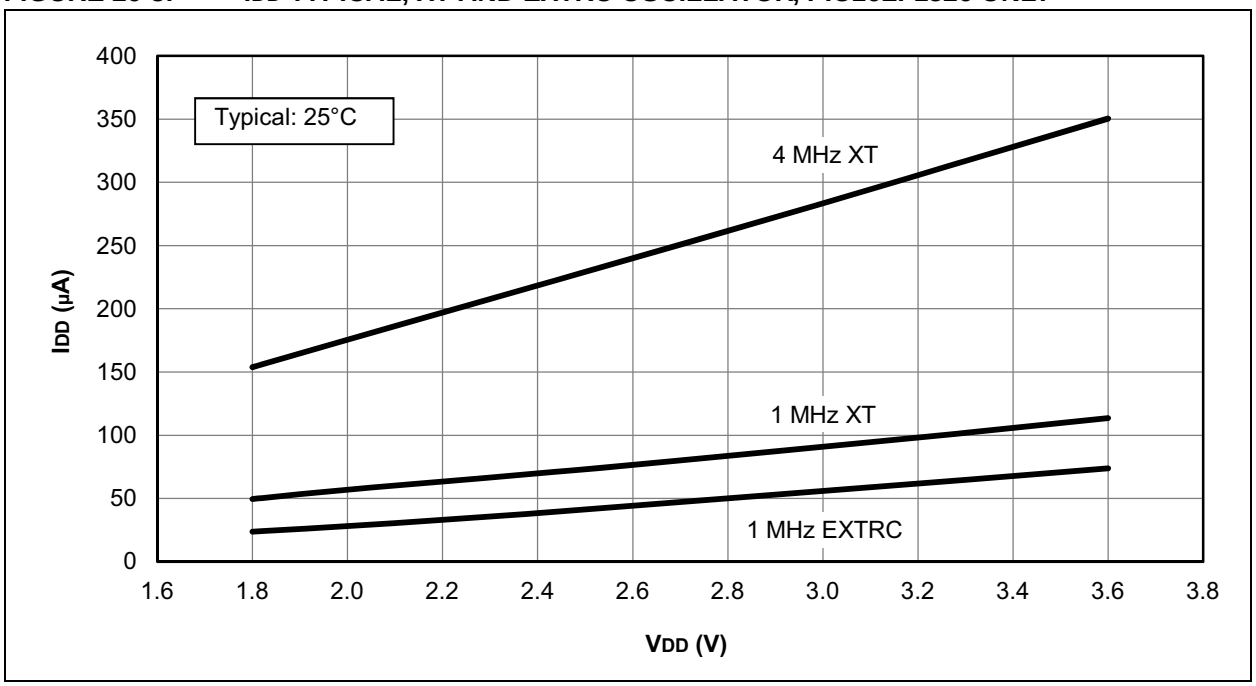


FIGURE 26-4: IDD MAXIMUM, XT AND EXTRC OSCILLATOR, PIC16LF1526 ONLY

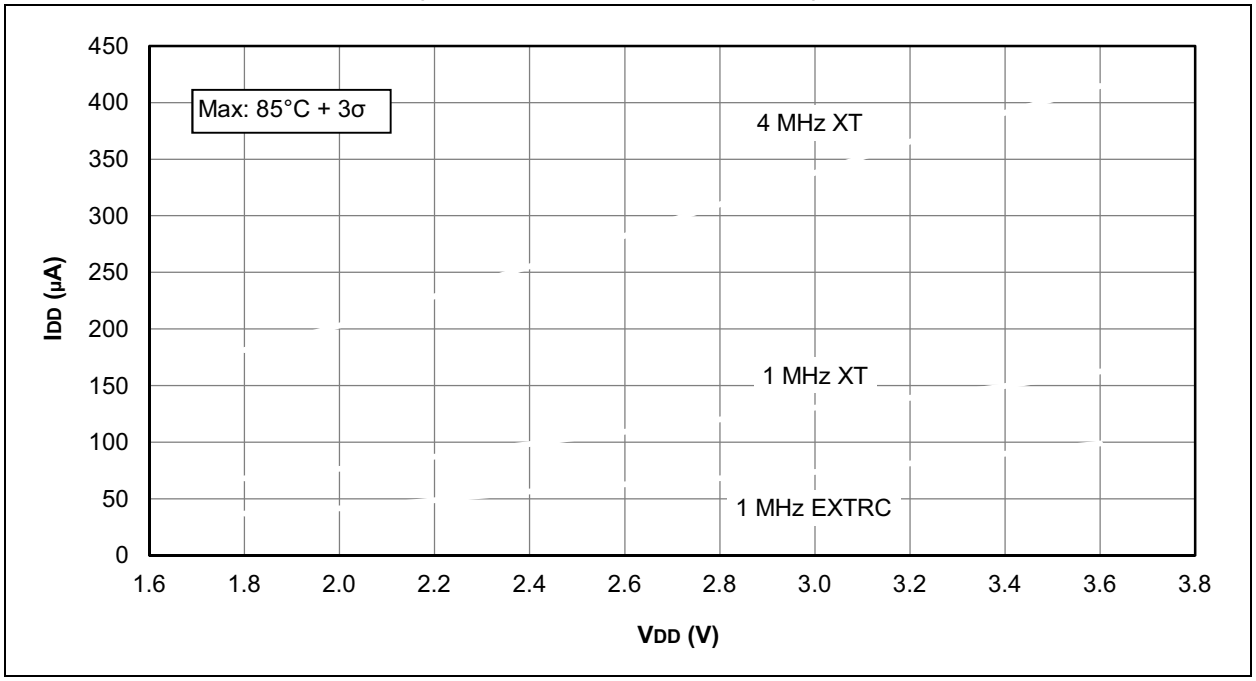


FIGURE 26-43: V_{OH} vs. I_{OH} OVER TEMPERATURE, V_{DD} = 3.0V

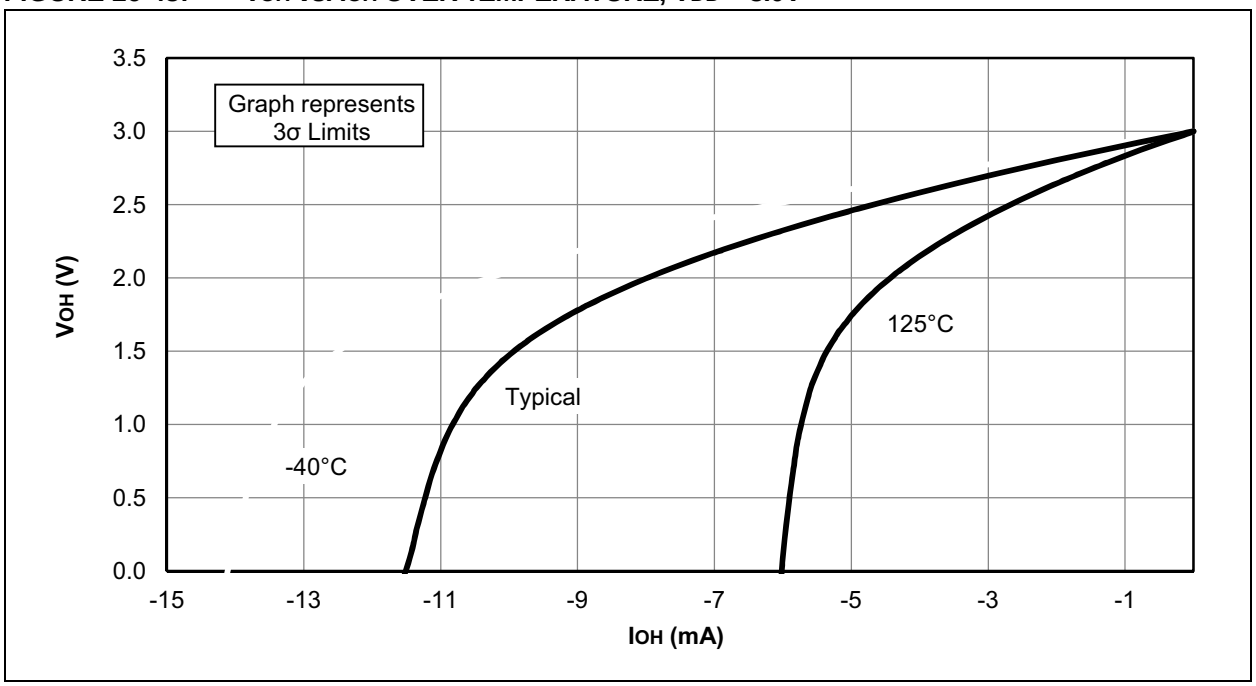


FIGURE 26-44: V_{OL} vs. I_{OL} OVER TEMPERATURE, V_{DD} = 3.0V