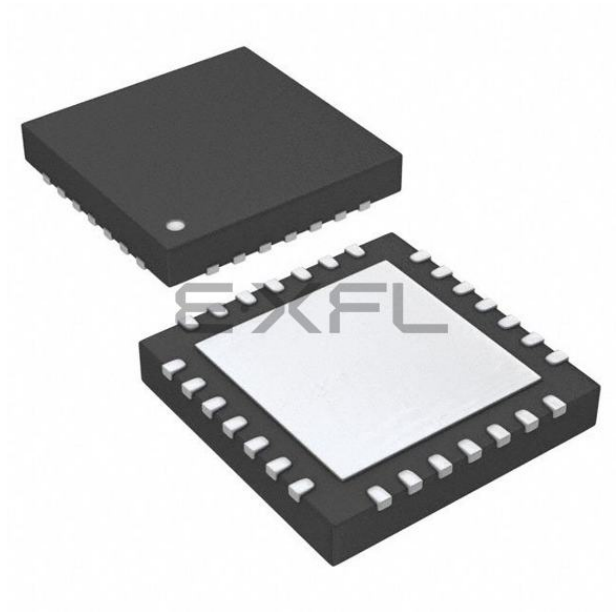


Welcome to E-XFL.COM

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	CANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k83-i-ml

REGISTER 5-2: CONFIGURATION WORD 1H (30 0001h)

U-1	U-1	R/W-1	U-1	R/W-1	U-1	R/W-1	R/W-1
—	—	FCMEN	—	CSWEN	—	PR1WAY	$\overline{\text{CLKOUTEN}}$
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 7-6 **Unimplemented:** Read as '1'

bit 5 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = FSCM timer is enabled
0 = FSCM timer is disabled

bit 4 **Unimplemented:** Read as '1'

bit 3 **CSWEN:** Clock Switch Enable bit
1 = Writing to NOSC and NDIV is allowed
0 = The NOSC and NDIV bits cannot be changed by user software

bit 2 **Unimplemented:** Read as '1'

bit 1 **PR1WAY:** PRLOCKED One-Way Set Enable bit
1 = PRLOCKED bit can be cleared and set only once; Priority registers remain locked after one clear/set cycle
0 = PRLOCKED bit can be set and cleared repeatedly (subject to the unlock sequence)

bit 0 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
If FEXTOSC<2:0> = EC (high, mid or low) or Not Enabled:
1 = CLKOUT function is disabled; I/O or oscillator function on OSC2
0 = CLKOUT function is enabled; FOSC/4 clock appears at OSC2
Otherwise:
This bit is ignored.

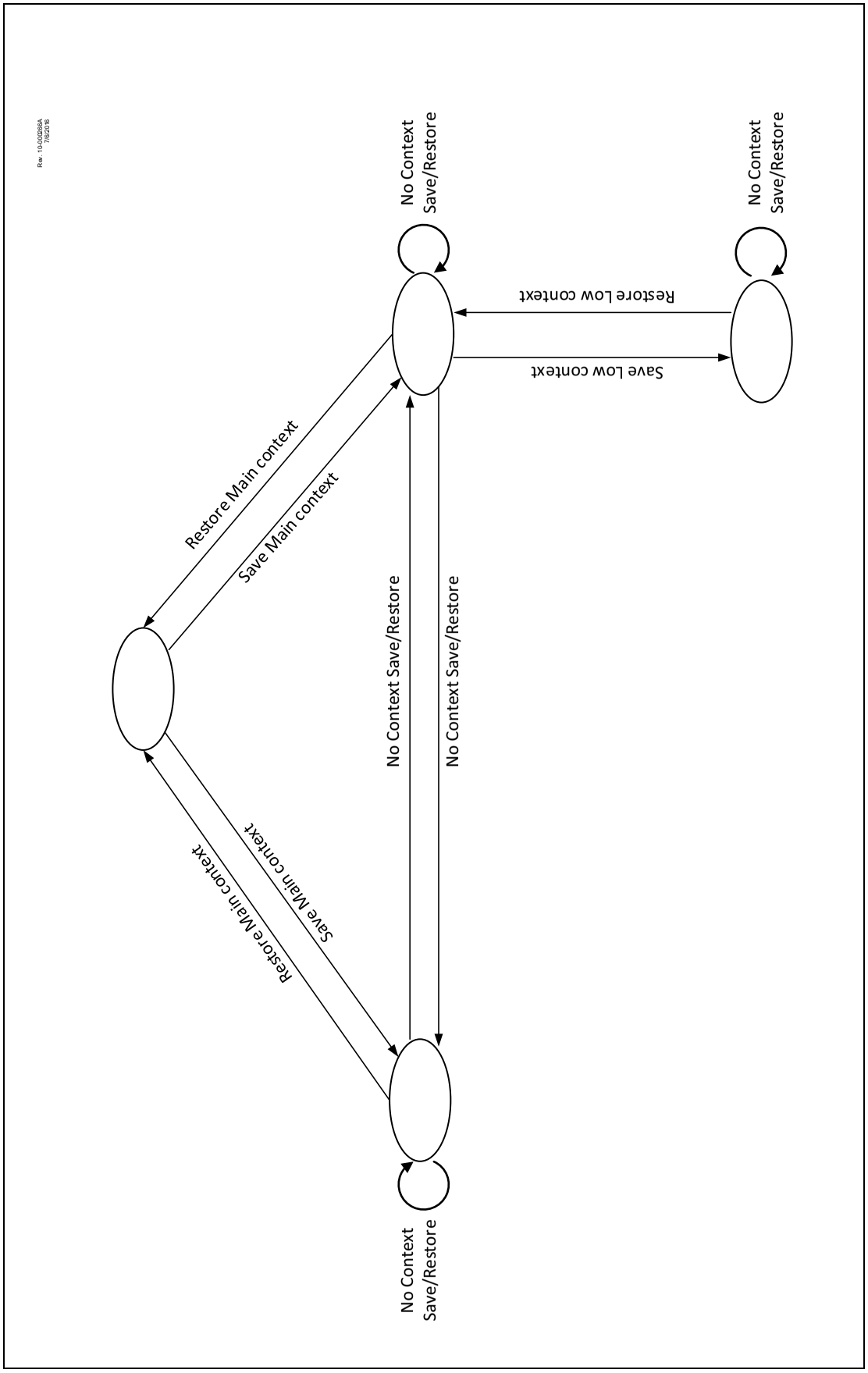


FIGURE 9-6: CONTEXT SAVE STATE MACHINE DIAGRAM

REGISTER 9-6: PIR3: PERIPHERAL INTERRUPT REGISTER 3(1)

R/W/HS-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
TMR0IF	U1IF(2)	U1EIF(3)	U1TXIF(4)	U1RXIF(4)	I2C1EIF(5)	I2C1IF(6)	I2C1TXIF(7)
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 = Bit is set in hardware

bit 7 **TMR0IF:** TMR0 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 6 **U1IF:** UART1 Interrupt Flag bit(2)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 5 **U1EIF:** UART1 Framing Error Interrupt Flag bit(3)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 4 **U1TXIF:** UART1 Transmit Interrupt Flag bit(4)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 3 **U1RXIF:** UART1 Receive Interrupt Flag bit(4)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

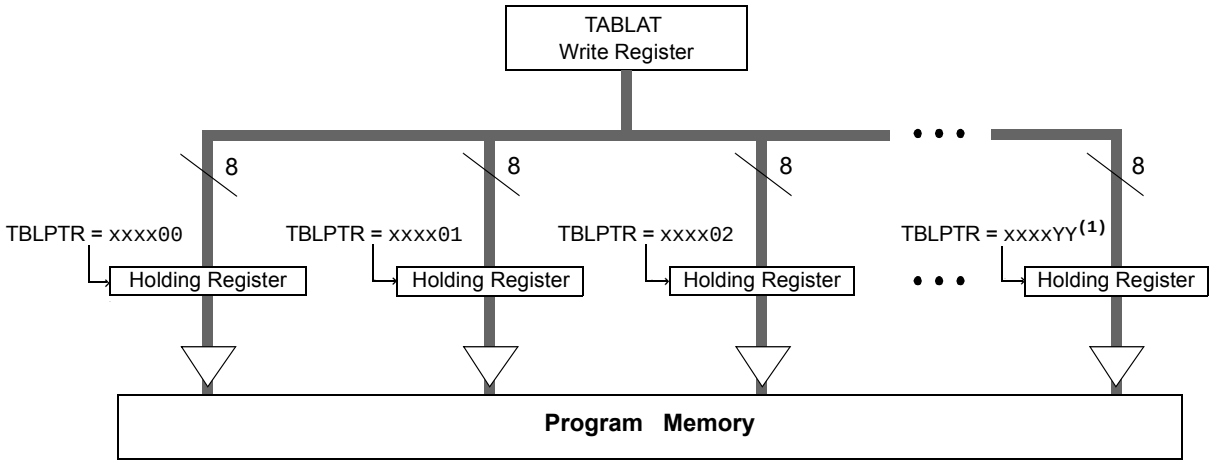
bit 2 **I2C1EIF:** I^2C1 Error Interrupt Flag bit(5)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 1 **I2C1IF:** I^2C1 Interrupt Flag bit(6)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 0 **I2C1TXIF:** I^2C1 Transmit Interrupt Flag bit(7)
1 = Interrupt has occurred
0 = Interrupt event has not occurred

Note 1: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.
2: UxIF is a read-only bit. To clear the interrupt condition, all bits in the UxUIR register must be cleared.
3: UxEIF is a read-only bit. To clear the interrupt condition, all bits in the UxERRIR register must be cleared.
4: UxTXIF and UxRXIF are read-only bits and cannot be set/cleared by the software.
5: I2CxEIF is a read-only bit. To clear the interrupt condition, all bits in the I2CxERR register must be cleared.
6: I2CxIF is a read-only bit. To clear the interrupt condition, all bits in the I2CxPIR register must be cleared.
7: I2CxTXIF and I2CxRXIF are read-only bits. To clear the interrupt condition, the CLRBF bit in I2CxSTAT1 register must be set.

FIGURE 13-8: TABLE WRITES TO PROGRAM FLASH MEMORY



Note 1: Refer to Table 5-4 for number of holding registers (e.g., YY = 3F for 64 holding registers).

13.1.6.1 Program Flash Memory Write Sequence

The sequence of events for programming an internal program memory location should be:

1. Read appropriate number of bytes into RAM. Refer to Table 13-2 for Write latch size.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the block erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the n-byte block into the holding registers with auto-increment. Refer to Table 13-2 for Write latch size.
7. Set REG<1:0> bits to point to program memory.
8. Clear FREE bit and set WREN bit in NVMCON1 register.
9. Disable interrupts.
10. Execute the unlock sequence (see **Section 13.1.4 "NVM Unlock Sequence"**).
11. WR bit is set in NVMCON1 register.
12. The CPU will stall for the duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update each write block of memory. An example of the required code is given in Example 13-4.

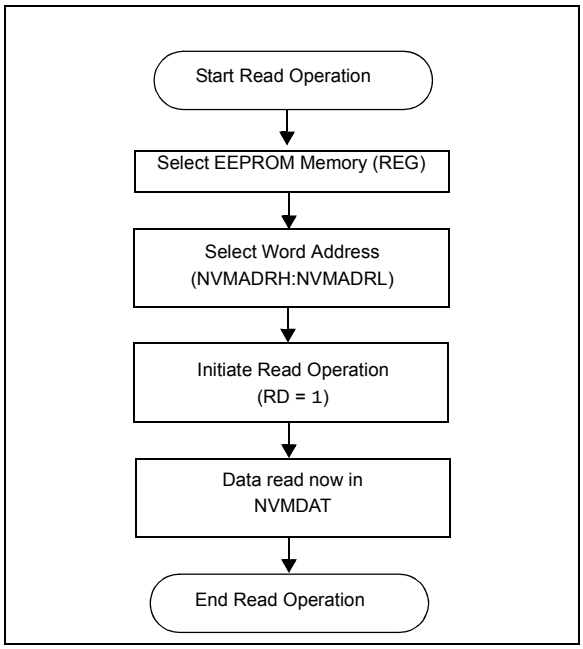
Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the bytes in the holding registers.

13.3.3 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the NVMADRL and NVMADRH register pair, clear REG<1:0> control bit in NVMCON1 register to access Data EEPROM locations and then set control bit, RD. The data is available on the very next instruction cycle; therefore, the NVMDAT register can be read by the next instruction. NVMDAT will hold this value until another read operation, or until it is written to by the user (during a write operation).

The basic process is shown in Example 13-5.

FIGURE 13-11: DATA EEPROM READ FLOWCHART



13.3.4 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the address must first be written to the NVMADRL and NVMADRH register pair and the data written to the NVMDAT register. The sequence in Example 13-6 must be followed to initiate the write cycle.

The write will not begin if NVM Unlock sequence, described in **Section 13.1.4 "NVM Unlock Sequence"**, is not exactly followed for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in NVMCON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

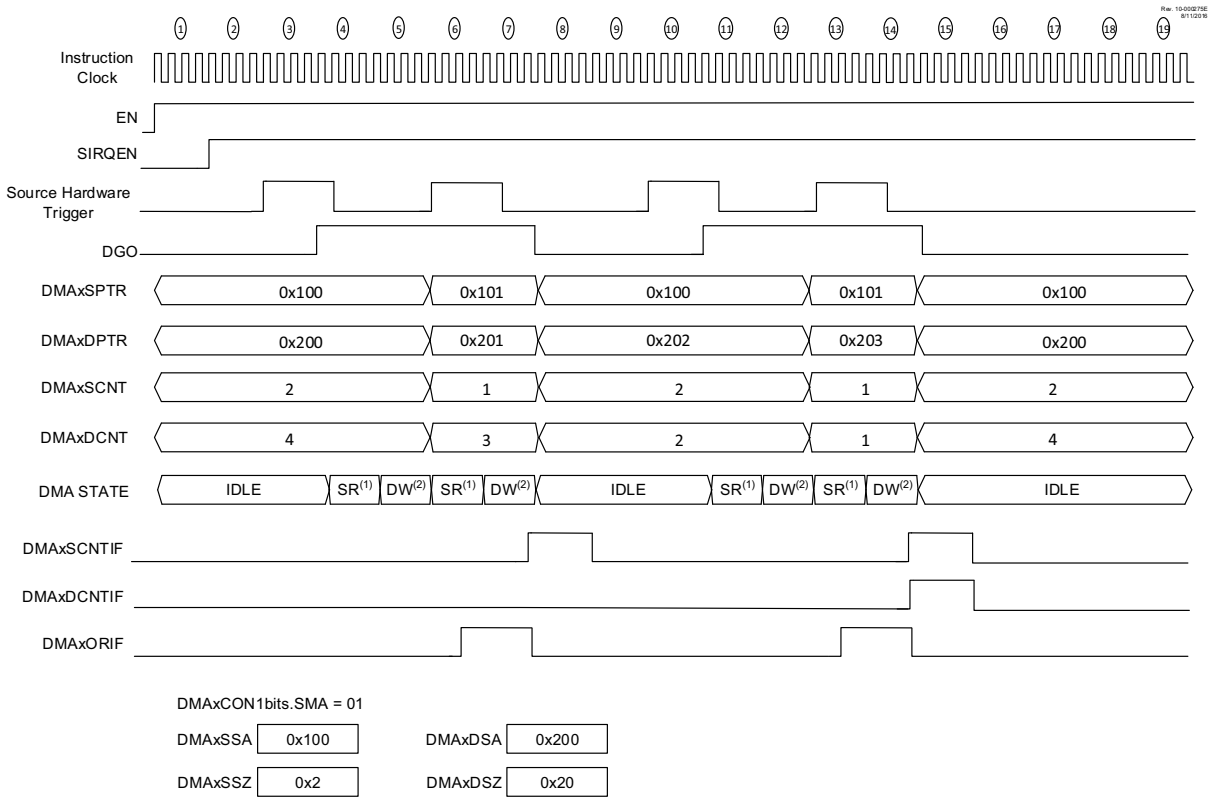
After a write sequence has been initiated, NVMCON1, NVMADRL, NVMADRH and NVMDAT cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. Both WR and WREN cannot be set with the same instruction.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. A single Data EEPROM word is written and the operation includes an implicit erase cycle for that word (it is not necessary to set FREE). CPU execution continues in parallel and at the completion of the write cycle, the WR bit is cleared in hardware and the NVM Interrupt Flag bit (NVMIF) is set. The user can either enable this interrupt or poll this bit. NVMIF must be cleared by software.

15.9.5 OVERRUN INTERRUPT

The Overrun Interrupt flag is set if the DMA receives a trigger to start a new message before the current message is completed.

FIGURE 15-9: OVERRUN INTERRUPT



Note 1: SR – Source Read

2: DW – Destination Write

27.2 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR5 register will be set when either edge detector is triggered and its associated enable bit is set. The INTP enables rising edge interrupts and the INTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- CLCxIE bit of the respective PIE register
- INTP bit of the CLCxCON register (for a rising edge detection)
- INTN bit of the CLCxCON register (for a falling edge detection)
- GIE bits of the INTCON0 register

The CLCxIF bit of the respective PIR register, must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

27.3 Output Mirror Copies

Mirror copies of all CON output bits are contained in the CLCxDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the OUT bits in the individual CLCxCON registers.

27.4 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

27.5 Operation During Sleep

The CLC module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

The HFINTOSC remains active during Sleep when the CLC module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLC input source, when the CLC is enabled, the CPU will go Idle during Sleep, but the CLC will continue to operate and the HFINTOSC will remain active.

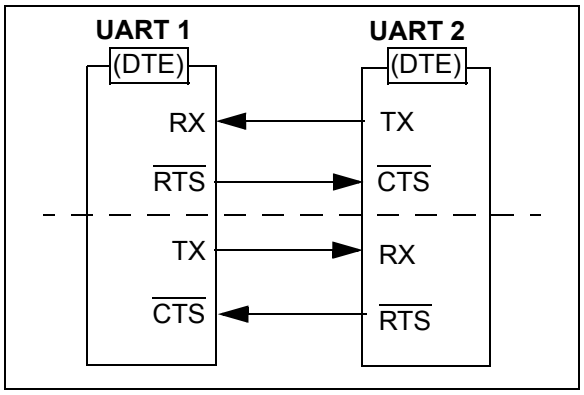
This will have a direct effect on the Sleep mode current.

27.6 CLCx Setup Steps

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the EN bit.
- Select desired inputs using CLCxSEL0 through CLCxSEL3 registers (See Table 27-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the GyPOL bits of the CLCxPOL register.
- Select the desired logic function with the MODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the POL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device pin, set the desired pin PPS control register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the INTP bit in the CLCxCON register for rising event.
  - Set the INTN bit in the CLCxCON register for falling event.
  - Set the CLCxIE bit of the respective PIE register.
  - Set the GIE bits of the INTCON0 register.
- Enable the CLCx by setting the EN bit of the CLCxCON register.

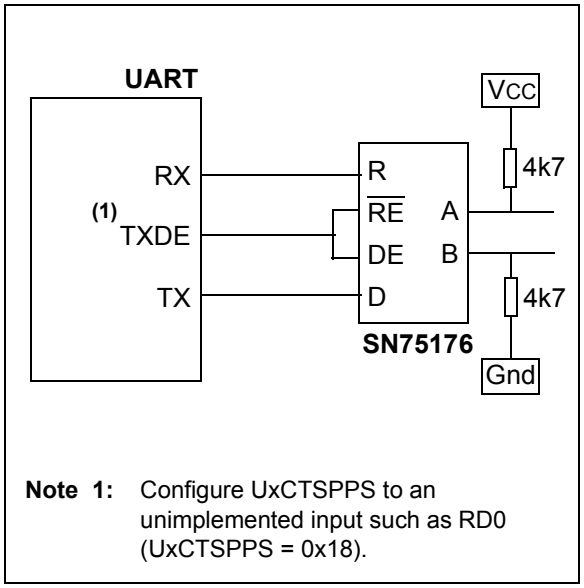
FIGURE 31-10: FLOW CONTROL



31.12.2 RS-485 TRANSCEIVER CONTROL

Hardware flow control can be used to control the direction of an RS-485 transceiver as shown in Figure 31-11. Configure the CTS input to be always enabled by setting the UxCTSPPS selection to an unimplemented port pin such as RD0. When the signal and control lines are configured as shown in Figure 31-11, then the UART will not receive its own transmissions. To verify that there are no collisions on the RS-485 lines then the transceiver RE control can be disconnected from TXDE and tied low thereby enabling loop-back reception of all transmissions. See **Section 31.14 "Collision Detection"** for more information.

FIGURE 31-11: RS-485 CONFIGURATION



Note 1: Configure UxCTSPPS to an unimplemented input such as RD0 (UxCTSPPS = 0x18).

31.12.3 XON/XOFF FLOW CONTROL

XON/XOFF flow control is selected by setting the FLO<1:0> bits to '01'.

XON/XOFF is a data based flow control method. The signals to suspend and resume transmission are special characters sent by the receiver to the transmitter The advantage is that additional hardware lines are not needed.

XON/XOFF flow control requires full-duplex operation because the transmitter must be able to receive the signal to suspend transmitting while the transmission is in progress. Although XON and XOFF are not defined in the ASCII code, the generally accepted values are 13h for XOFF and 11h for XON. The UART uses those codes.

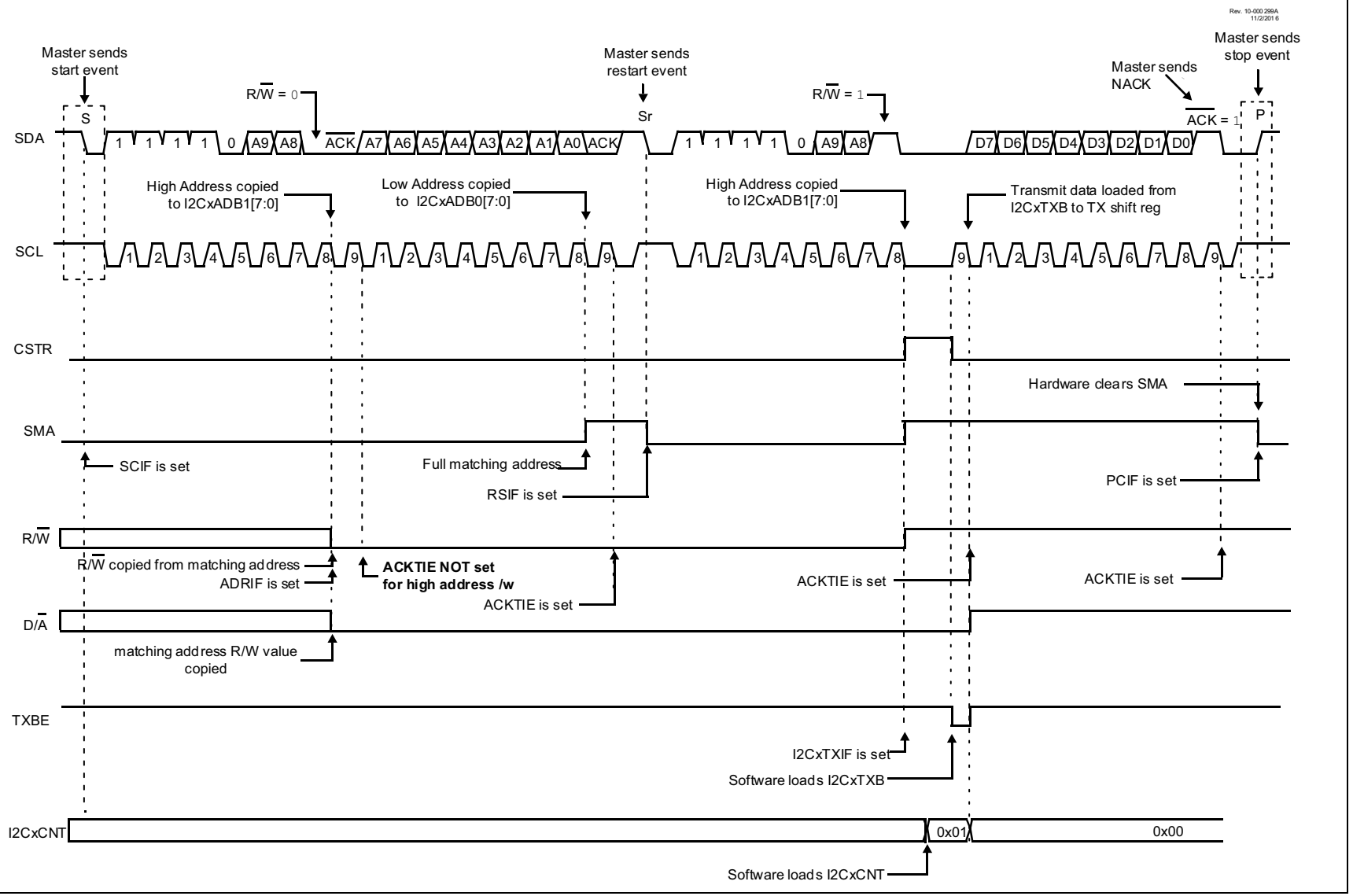
The transmitter defaults to XON, or transmitter enabled. This state is also indicated by the read-only XON bit in the UxFIFO register.

When an XOFF character is received, the transmitter stops transmitting after completing the character actively being transmitted. The transmitter remains disabled until an XON character is received.

XON will be forced on when software toggles the TXEN bit.

When the RUNOVF bit in the UxCON2 register is set then XON and XOFF characters continue to be received and processed without the need to clear the input FIFO by reading the UxRXB. However, if the RUNOVF bit is clear then the UxRXB must be read to avoid a receive overflow which will suspend flow control when the receive buffer overflows.

FIGURE 33-12: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION



messages: SDFLC

34.3.3 MODE 2 – ENHANCED FIFO MODE

In Mode 2, two or more receive buffers are used to form the receive FIFO (first in, first out) buffer. There is no one-to-one relationship between the receive buffer and acceptance filter registers. Any filter that is enabled and linked to any FIFO receive buffer can generate acceptance and cause FIFO to be updated.

FIFO length is user-programmable, from 2-8 buffers deep. FIFO length is determined by the very first programmable buffer that is configured as a transmit buffer. For example, if Buffer 2 (B2) is programmed as a transmit buffer, FIFO consists of RXB0, RXB1, B0 and B1, creating a FIFO length of four. If all programmable buffers are configured as receive buffers, FIFO will have the maximum length of eight.

The following is the list of resources available in Mode 2:

- Three transmit buffers: TXB0, TXB1 and TXB2
- Two receive buffers: RXB0 and RXB1
- Six buffers programmable as TX or RX; receive buffers form FIFO: B0-B5
- Automatic RTR handling on B0-B5
- Sixteen acceptance filters: RXF0-RXF15
- Two dedicated acceptance mask registers; RXF15 programmable as third mask: RXM0-RXM1, RXF15
- Programmable data filter on standard identifier messages: SDFLC, useful for DeviceNet protocol

34.4 CAN Message Buffers

34.4.1 DEDICATED TRANSMIT BUFFERS

The CAN module implements three dedicated transmit buffers – TXB0, TXB1 and TXB2. Each of these buffers occupies 14 bytes of SRAM and are mapped into the SFR memory map. These are the only transmit buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each transmit buffer contains one Control register (TXBnCON), four Identifier registers (TXBnSIDL, TXBnSIDH, TXBnEIDL, TXBnEIDH), one Data Length Count register (TXBnDLC) and eight Data Byte registers (TXBnDm).

34.4.2 DEDICATED RECEIVE BUFFERS

The CAN module implements two dedicated receive buffers: RXB0 and RXB1. Each of these buffers occupies 14 bytes of SRAM and are mapped into SFR memory map. These are the only receive buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each receive buffer contains one Control register (RXBnCON), four Identifier registers (RXBnSIDL, RXBnSIDH, RXBnEIDL, RXBnEIDH), one Data Length Count register (RXBnDLC) and eight Data Byte registers (RXBnDm).

There is also a separate Message Assembly Buffer (MAB) which acts as an additional receive buffer. MAB is always committed to receiving the next message from the bus and is not directly accessible to user firmware. The MAB assembles all incoming messages one by one. A message is transferred to appropriate receive buffers only if the corresponding acceptance filter criteria is met.

34.4.3 PROGRAMMABLE TRANSMIT/ RECEIVE BUFFERS

The CAN module implements six non-dedicated buffers: B0-B5. These buffers are individually programmable as either transmit or receive buffers. These buffers are available only in Mode 1 and 2. As with dedicated transmit and receive buffers, each of these programmable buffers occupies 14 bytes of SRAM and are mapped into SFR memory map.

Each buffer contains one Control register (BnCON), four Identifier registers (BnSIDL, BnSIDH, BnEIDL, BnEIDH), one Data Length Count register (BnDLC) and eight Data Byte registers (BnDm). Each of these registers contains two sets of control bits. Depending on whether the buffer is configured as transmit or receive, one would use the corresponding control bit set. By default, all buffers are configured as receive buffers. Each buffer can be individually configured as a transmit or receive buffer by setting the corresponding TXENn bit in the BSEL0 register.

When configured as transmit buffers, user firmware may access transmit buffers in any order similar to accessing dedicated transmit buffers. In receive configuration with Mode 1 enabled, user firmware may also access receive buffers in any order required. But in Mode 2, all receive buffers are combined to form a single FIFO. Actual FIFO length is programmable by user firmware. Access to FIFO must be done through the FIFO Pointer bits (FP<4:0>) in the CANCON register. It must be noted that there is no hardware protection against out of order FIFO reads.

34.4.4 PROGRAMMABLE AUTO-RTR BUFFERS

In Mode 1 and 2, any of six programmable transmit/ receive buffers may be programmed to automatically respond to predefined RTR messages without user firmware intervention. Automatic RTR handling is enabled by setting the TX2EN bit in the BSEL0 register and the RTREN bit in the BnCON register. After this setup, when an RTR request is received, the TXREQ bit is automatically set and the current buffer content is automatically queued for transmission as a RTR

response. As with all transmit buffers, once the TXREQ bit is set, buffer registers become read-only and any writes to them will be ignored.

REGISTER 34-23: BnCON: TX/RX BUFFER 'n' CONTROL REGISTERS IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1](1)

R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TXBIF(3)	TXABT(3)	TXLARB(3)	TXERR(3)	TXREQ(2,4)	RTREN	TXPRI1(5)	TXPRI0(5)
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 **TXBIF:** Transmit Buffer Interrupt Flag bit(3)
1 = A message was successfully transmitted
0 = No message was transmitted

bit 6 **TXABT:** Transmission Aborted Status bit(3)
1 = Message was aborted
0 = Message was not aborted

bit 5 **TXLARB:** Transmission Lost Arbitration Status bit(3)
1 = Message lost arbitration while being sent
0 = Message did not lose arbitration while being sent

bit 4 **TXERR:** Transmission Error Detected Status bit(3)
1 = A bus error occurred while the message was being sent
0 = A bus error did not occur while the message was being sent

bit 3 **TXREQ:** Transmit Request Status bit(2,4)
1 = Requests sending a message; clears the TXABT, TXLARB and TXERR bits
0 = Automatically cleared when the message is successfully sent

bit 2 **RTREN:** Automatic Remote Transmission Request Enable bit
1 = When a remote transmission request is received, TXREQ will be automatically set
0 = When a remote transmission request is received, TXREQ will be unaffected

bit 1-0 **TXPRI<1:0>:** Transmit Priority bits(5)
11 = Priority Level 3 (highest priority)
10 = Priority Level 2
01 = Priority Level 1
00 = Priority Level 0 (lowest priority)

Note 1: These registers are available in Mode 1 and 2 only.
2: Clearing this bit in software while the bit is set will request a message abort.
3: This bit is automatically cleared when TXREQ is set.
4: While TXREQ is set or a transmission is in progress, Transmit Buffer registers remain read-only.
5: These bits set the order in which the Transmit Buffer register will be transferred. They do not alter the CAN message identifier.

REGISTER 34-39: RXFnEIDH: RECEIVE ACCEPTANCE FILTER 'n' EXTENDED IDENTIFIER REGISTERS, HIGH BYTE [0 ≤ n ≤ 15](1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
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 7-0 **EID<15:8>:** Extended Identifier Filter bits

Note 1: Registers, RXF6EIDH:RXF15EIDH, are available in Mode 1 and 2 only.

REGISTER 34-40: RXFnEIDL: RECEIVE ACCEPTANCE FILTER 'n' EXTENDED IDENTIFIER REGISTERS, LOW BYTE [0 ≤ n ≤ 15](1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
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 7-0 **EID<7:0>:** Extended Identifier Filter bits

Note 1: Registers, RXF6EIDL:RXF15EIDL, are available in Mode 1 and 2 only.

REGISTER 34-41: RXMnSIDH: RECEIVE ACCEPTANCE MASK 'n' STANDARD IDENTIFIER MASK REGISTERS, HIGH BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
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 7-0 **SID<10:3>:** Standard Identifier Mask bits or Extended Identifier Mask bits (EID<28:21>)

REGISTER 34-44: RXMnEIDL: RECEIVE ACCEPTANCE MASK 'n' EXTENDED IDENTIFIER MASK REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
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 7-0 **EID<7:0>:** Extended Identifier Mask bits

REGISTER 34-45: RXFCONn: RECEIVE FILTER CONTROL REGISTER 'n' [0 ≤ n ≤ 1](1)

RXFCON0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN

RXFCON1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN
	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 7-0 **RXF<7:0>EN:** Receive Filter n Enable bits
0 = Filter is disabled
1 = Filter is enabled

Note 1: This register is available in Mode 1 and 2 only.

Note: Register 34-46 through Register 34-51 are writable in Configuration mode only.

REGISTER 34-47: RXFBCONn: RECEIVE FILTER BUFFER CONTROL REGISTER 'n'(1)

RXFBCON0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F1BP_3	F1BP_2	F1BP_1	F1BP_0	F0BP_3	F0BP_2	F0BP_1	F0BP_0
RXFBCON1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
	F3BP_3	F3BP_2	F3BP_1	F3BP_0	F2BP_3	F2BP_2	F2BP_1	F2BP_0
RXFBCON2	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
	F5BP_3	F5BP_2	F5BP_1	F5BP_0	F4BP_3	F4BP_2	F4BP_1	F4BP_0
RXFBCON3	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F7BP_3	F7BP_2	F7BP_1	F7BP_0	F6BP_3	F6BP_2	F6BP_1	F6BP_0
RXFBCON4	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F9BP_3	F9BP_2	F9BP_1	F9BP_0	F8BP_3	F8BP_2	F8BP_1	F8BP_0
RXFBCON5	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F11BP_3	F11BP_2	F11BP_1	F11BP_0	F10BP_3	F10BP_2	F10BP_1	F10BP_0
RXFBCON6	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F13BP_3	F13BP_2	F13BP_1	F13BP_0	F12BP_3	F12BP_2	F12BP_1	F12BP_0
RXFBCON7	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	F15BP_3	F15BP_2	F15BP_1	F15BP_0	F14BP_3	F14BP_2	F14BP_1	F14BP_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 7-0 **F<15:2>BP_<3:0>:** Filter n Buffer Pointer Nibble bits

0000 = Filter n is associated with RXB0
0001 = Filter n is associated with RXB1
0010 = Filter n is associated with B0
0011 = Filter n is associated with B1
...
0111 = Filter n is associated with B5
1111-1000 = Reserved

Note 1: This register is available in Mode 1 and 2 only.

39.7 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 45-15 and Table 45-17 for more details.

39.8 Analog Input Connection Considerations

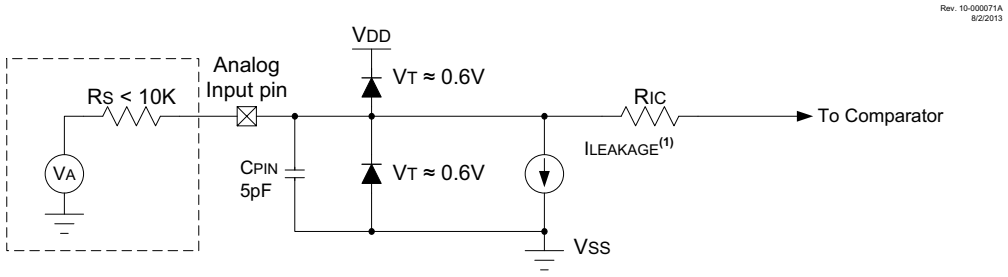
A simplified circuit for an analog input is shown in Figure 39-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 39-3: ANALOG INPUT MODEL



Legend: C_{PIN} = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_S = Source Impedance
V_A = Analog Voltage
V_T = Threshold Voltage

Note 1: See **Section 45.0 "Electrical Specifications"**.

RLNCF **Rotate Left f (No Carry)**

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

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

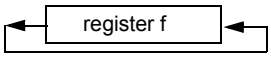
Operation: (f<n>) → dest<n + 1>,
(f<7>) → dest<0>

Status Affected: N, Z

Encoding:

0100	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 42.2.3 "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: RLNCF REG, 1, 0

Before Instruction
REG = 1010 1011
After Instruction
REG = 0101 0111

RRCF **Rotate Right f through Carry**

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

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

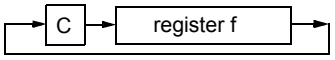
Operation: (f<n>) → dest<n – 1>,
(f<0>) → C,
(C) → dest<7>

Status Affected: C, N, Z

Encoding:

0011	00da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the CARRY flag. 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 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 42.2.3 "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: RRCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 0111 0011
C = 0

TABLE 43-1: REGISTER FILE SUMMARY FOR PIC18(L)F25/26K83 DEVICES (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3DF9h	U1ERRIR	TXMTIF	PERIF	ABDOVF	CERIF	FERIF	RXBKIF	RXFOIF	TXCIF	486
3DF8h	U1UIR	WUIF	ABDIF	—	—	—	ABDIE	—	—	488
3DF7h	U1FIFO	TXWRE	STPMD	TXBE	TXBF	RXIDL	XON	RXBE	RXBF	489
3DF6h	U1BRGH	BRGH								490
3DF5h	U1BRGL	BRGL								490
3DF4h	U1CON2	RUNOVF	RXPOL	STP		C0EN	TXPOL	FLO		485
3DF3h	U1CON1	ON	—	—	WUE	RXBIMD	—	BRKOVR	SENDB	484
3DF2h	U1CON0	BRGS	ABDEN	TXEN	RXEN	MODE				483
3DF1h	U1P3H	—	—	—	—	—	—	—	P3H	494
3DF0h	U1P3L	P3L								494
3DEFh	U1P2H	—	—	—	—	—	—	—	P2H	493
3DEEh	U1P2L	P2L								493
3DEDh	U1P1H	—	—	—	—	—	—	—	P1H	492
3DECh	U1P1L	P1L								492
3DEBh	U1TXCHK	TXCHK								495
3DEAh	U1TXB	TXB								491
3DE9h	U1RXCHK	RXCHK								495
3DE8h	U1RXB	RXB								491
3DE7h - 3DE3h	—	Unimplemented								—
3DE2h	U2ERRIE	TXMTIE	PERIE	ABDOVE	CERIE	FERIE	RXBKIE	RXFOIE	TXCIE	487
3DE1h	U2ERRIR	TXMTIF	PERIF	ABDOVF	CERIF	FERIF	RXBKIF	RXFOIF	TXCIF	486
3DE0h	U2UIR	WUIF	ABDIF	—	—	—	ABDIE	—	—	488
3DDFh	U2FIFO	TXWRE	STPMD	TXBE	TXBF	RXIDL	XON	RXBE	RXBF	489
3DDEh	U2BRGH	BRGH								490
3DDDh	U2BRGL	BRGL								490
3DDCh	U2CON2	RUNOVF	RXPOL	STP		—	TXPOL	FLO		485
3DDBh	U2CON1	ON	—	—	WUE	RXBIMD	—	BRKOVR	SENDB	484
3DDAh	U2CON0	BRGS	ABDEN	TXEN	RXEN	MODE				483
3DD9h	U2P3H	—	—	—	—	—	—	—	P3H	494
3DD8h	U2P3L	P3L								494
3DD7h	U2P2H	—	—	—	—	—	—	—	P2H	493
3DD6h	U2P2L	P2L								493
3DD5h	U2P1H	—	—	—	—	—	—	—	P1H	492
3DD4h	U2P1L	P1L								492
3DD3h	U2TXCHK	TXCHK								495
3DD2h	U2TXB	TXB								491
3DD1h	U2RXCHK	RXCHK								495
3DD0h	U2RXB	RXB								491
3DCFh - 3D7Dh	—	Unimplemented								—
3D7Ch	I2C1BTO	BTO								567
3D7Bh	I2C1CLK	CLK								566
3D7Ah	I2C1PIE	CNTIE	ACKTIE	—	WRIE	ADRIE	PCIE	RSCIE	SCIE	573
3D79h	I2C1PIR	CNTIF	ACKTIF	—	WRIF	ADRIF	PCIF	RSCIF	SCIF	572
3D78h	I2C1STAT1	TXWE	—	TXBE	—	RXRE	CLRBF	—	RXBF	569
3D77h	I2C1STAT0	BFRE	SMA	MMA	R	D	—	—	—	568
3D76h	I2C1ERR	—	BTOIF	BCLIF	NACKIF	—	BTOIE	BCLIE	NACKIE	570
3D75h	I2C1CON2	ACNT	GCEN	FME	ABD	SDAHT		BFRET		565
3D74h	I2C1CON1	ACKCNT	ACKDT	ACKSTAT	ACKT	—	RXO	TXU	CSD	564

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

TABLE 43-1: REGISTER FILE SUMMARY FOR PIC18(L)F25/26K83 DEVICES (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3716h	RXF9EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
3715h	RXF9SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
3714h	RXF9SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
3713h	RXF8EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
3712h	RXF8EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
3711h	RXF8SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
3710h	RXF8SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
370Fh	RXF7EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
370Eh	RXF7EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
370Dh	RXF7SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
370Ch	RXF7SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
370Bh	RXF6EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
370Ah	RXF6EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
3709h	RXF6SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
3708h	RXF6SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
3707h	RXFCON1	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN	633
3706h	RXFCON0	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN	633
3705h	BRGCON3	WAKDIS	WAKFIL	—	—	—	SEG2PH2	SEG2PH1	SEG2PH0	642
3704h	BRGCON2	SEG2PHTS	SAM	SEG1PH2	SEG1PH1	SEG1PH0	PRSEG2	PRSEG1	PRSEG0	641
3703h	BRGCON1	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	640
3702h	TXERRCNT	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	612
3701h	RXERRCNT	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	621
3700h	CIOCON	TX1SRC	—	—	—	—	—	—	CLKSEL	643

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

TABLE 45-7: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS (CONTINUED)

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS21	F_{CY}	Instruction Frequency	—	FOSC/4	—	MHz	
OS22	T_{CY}	Instruction Period	62.5	$1/F_{CY}$	—	ns	

* 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.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: The system clock frequency (FOSC) is selected by the "main clock switch controls" as described in **Section 10.0 "Power-Saving Operation Modes"**.

3: The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 45.2 "Standard Operating Conditions"**.

4: LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device. For clocking the device with the external square wave, one of the EC mode selections must be used.