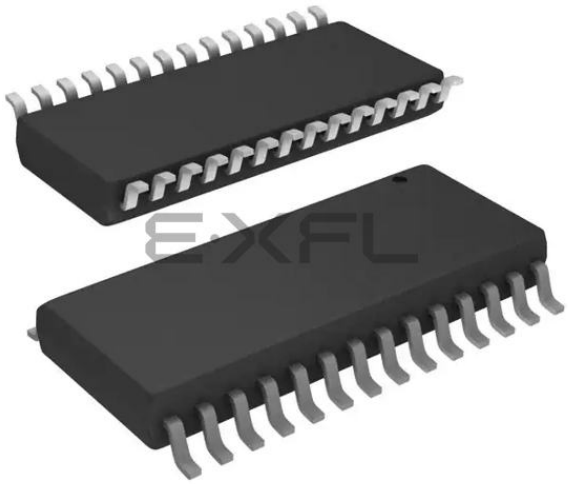


Welcome to E-XFL.COM

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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k83-i-so

3.2.3 ISR PRIORITY > PERIPHERAL PRIORITY > MAIN PRIORITY

In this case, interrupt routines and peripheral operation (DMAx, Scanner) will stall the CPU. Interrupt will preempt peripheral operation. This results in lowest interrupt latency and highest throughput for the peripheral to access the memory.

3.2.4 PERIPHERAL 1 PRIORITY > ISR PRIORITY > MAIN PRIORITY > PERIPHERAL 2 PRIORITY

In this case, the Peripheral 1 will stall the execution of the CPU. However, Peripheral 2 can access the memory in cycles unused by Peripheral 1.

The operation of the System Arbiter is controlled through the following registers:

REGISTER 3-1: ISRPR: INTERRUPT SERVICE ROUTINE PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	ISRPR<2:0>		
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	HS = Hardware set

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ISRPR<2:0>**: Interrupt Service Routine Priority Selection bits

REGISTER 3-2: MAINPR: MAIN ROUTINE PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-1/1
—	—	—	—	—	MAINPR<2:0>		
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	HS = Hardware set

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **MAINPR<2:0>**: Main Routine Priority Selection bits

REGISTER 3-3: DMA1PR: DMA1 PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0
—	—	—	—	—	DMA1PR<2:0>		
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	HS = Hardware set

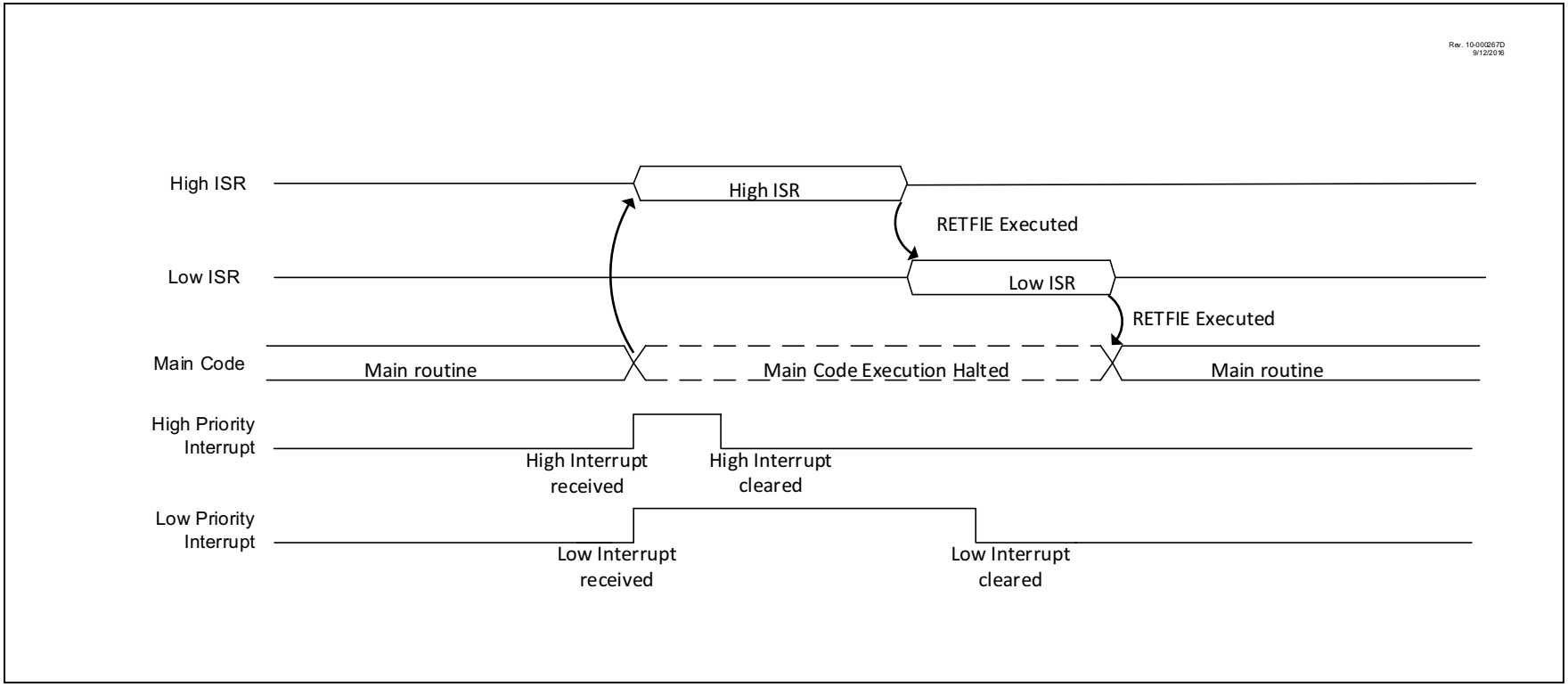
bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **DMA1PR<2:0>**: DMA1 Priority Selection bits

9.4.4 SIMULTANEOUS LOW AND HIGH PRIORITY INTERRUPTS

When both high and low interrupts are active in the same instruction cycle (i.e., simultaneous interrupt events), both the high and the low priority requests are generated. The high priority ISR is serviced first before servicing the low priority interrupt see Figure 9-5.

FIGURE 9-5: INTERRUPT EXECUTION: SIMULTANEOUS LOW AND HIGH PRIORITY INTERRUPTS



13.4 Register Definitions: Nonvolatile Memory

REGISTER 13-1: NVMCON1: NONVOLATILE MEMORY CONTROL 1 REGISTER

R/W-0/0	R/W-0/0	U-0	R/S/HC-0/0	R/W/HS-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
REG<1:0>		—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	HC = Bit is cleared by hardware
x = Bit is unknown	-n = Value at POR	S = Bit can be set by software, but not cleared
'0' = Bit is cleared	'1' = Bit is set	U = Unimplemented bit, read as '0'

bit 7-6 **REG<1:0>:** NVM Region Selection bit
10 =Access PFM Locations
x1 = Access User IDs, Configuration Bits, DIA, DCI, Rev ID and Device ID
00 = Access Data EEPROM Memory Locations

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

bit 4 **FREE:** Program Flash Memory Erase Enable bit(1)
1 = Performs an erase operation on the next WR command
0 = The next WR command performs a write operation

bit 3 **WRERR:** Write-Reset Error Flag bit(2,3,4)
1 = A write operation was interrupted by a Reset (hardware set),
or WR was written to 1'b1 when an invalid address is accessed (Table 4-1, Table 13-1)
or WR was written to 1'b1 when REG<1:0> and address do not point to the same region
or WR was written to 1'b1 when a write-protected address is accessed (Table 4-2).
0 = All write operations have completed normally

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase and refresh cycles
0 = Inhibits programming/erasing and user refresh of NVM

bit 1 **WR:** Write Control bit(5,6,7)
When REG points to a Data EEPROM Memory location:
1 = Initiates an erase/program cycle at the corresponding Data EEPROM Memory location
When REG points to a PFM location:
1 = Initiates the PFM write operation with data from the holding registers
0 = NVM program/erase operation is complete and inactive

bit 0 **RD:** Read Control bit(8)
1 = Initiates a read at address pointed by REG and NVMADR, and loads data into NVMDAT
0 = NVM read operation is complete and inactive

Note 1: This can only be used with PFM.
2: This bit is set when WR = 1 and clears when the internal programming timer expires or the write is completed successfully.
3: Bit must be cleared by the user; hardware will not clear this bit.
4: Bit may be written to '1' by the user in order to implement test sequences.
5: This bit can only be set by following the unlock sequence of **Section 13.1.4 "NVM Unlock Sequence"**.
6: Operations are self-timed and the WR bit is cleared by hardware when complete.
7: Once a write operation is initiated, setting this bit to zero will have no effect.
8: The bit can only be set in software. The bit is cleared by hardware when the operation is complete.

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CRCACCH	ACC<15:8>								209
CRCACCL	ACC<7:0>								210
CRCCON0	EN	GO	BUSY	ACCM	—	—	SHIFTM	FULL	208
CRCCON1	DLEN<3:0>				PLEN<3:0>				208
CRCDATH	DATA<15:8>								209
CRCDATL	DATA<7:0>								209
CRCSHIFTH	SHIFT<15:8>								210
CRCSHIFTL	SHIFT<7:0>								210
CRCXORH	X<15:8>								211
CRCXORL	X<7:1>							—	211
SCANCON0	EN	TRIGEN	SGO	—	—	MREG	BURSTMD	BUSY	212
SCANHADRU	—	—	HADR<21:16>						214
SCANHADRH	HADR<15:8>								215
SCANHADRL	HADR<7:0>								215
SCANLADRU	—	—	LADR<21:16>						213
SCANLADRH	LADR<15:8>								213
SCANLADRL	LADR<7:0>								214
SCANTRIG	—	—	—	—	TSEL<3:0>				216

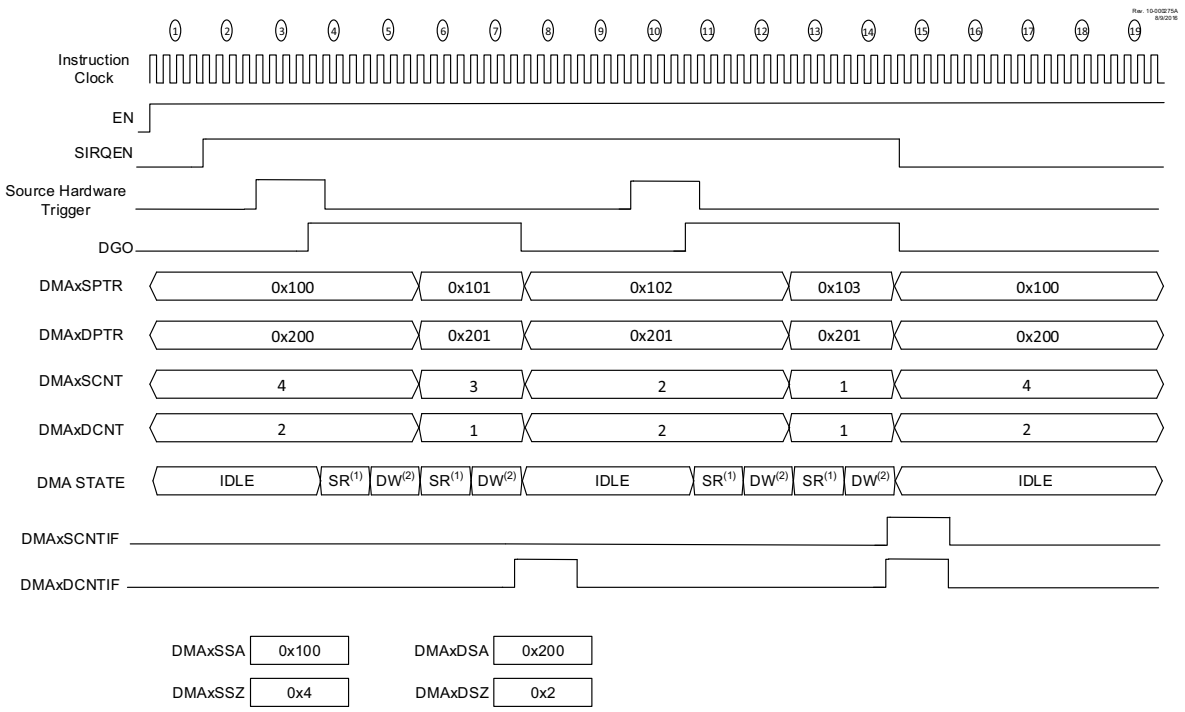
Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the CRC module.

The following sections describe with visual reference the sequence of events for different configurations of the DMA module

15.9.1 SOURCE STOP

When the Source Stop bit is set (SSTP = 1) and the DMAxSCNT register reloads, the DMA clears the SIRQEN bit to stop receiving new start interrupt request signals and sets the DMAxSCNTIF flag.

FIGURE 15-5: GPR-GPR TRANSACTIONS WITH HARDWARE TRIGGERS, SSTP = 1



Note 1: SR – Source Read

2: DW – Destination Write

REGISTER 16-3: LATx: LATx REGISTER(1)

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATx7	LATx6	LATx5	LATx4	LATx3	LATx2	LATx1	LATx0
bit 7							bit 0

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

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

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

TABLE 16-4: LAT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0

25.6.7 TIME OF FLIGHT MEASURE MODE

This mode measures the time interval between a rising edge on the SMTWINx input and a rising edge on the SMTx_signal input, beginning to increment the timer upon observing a rising edge on the SMTWINx input, while updating the SMTxCPR register and resetting the timer upon observing a rising edge on the SMTx_signal input. In the event of two SMTWINx rising edges without an SMTx_signal rising edge, it will update the SMTxCPW register with the current value of the timer and reset the timer value. See Figure 25-14 and Figure 25-15.

FIGURE 25-15: TIME OF FLIGHT MODE SINGLE ACQUISITION TIMING DIAGRAM

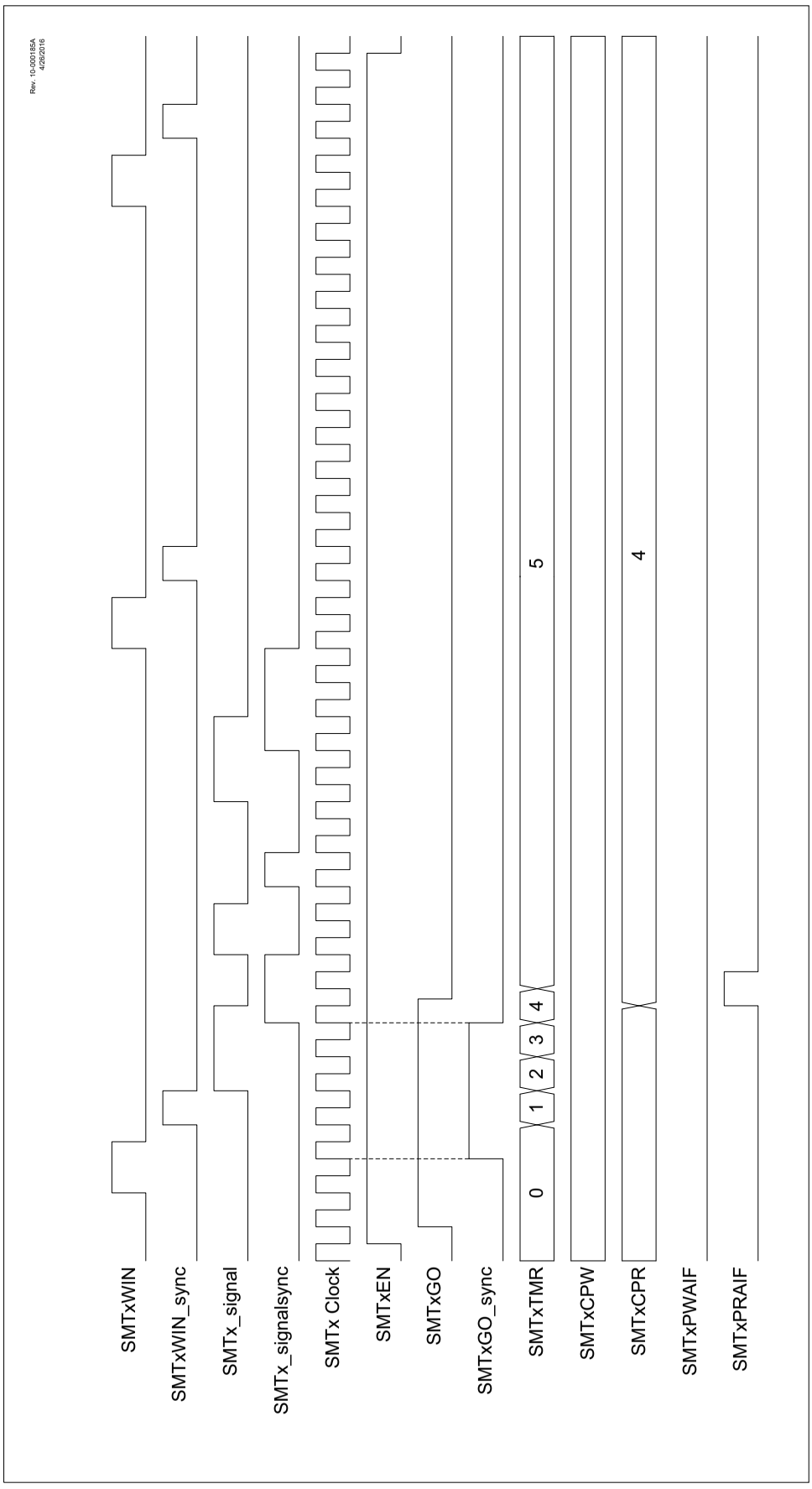


FIGURE 26-16: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (REN = 1, LSAC = 01, LSBD = 01)

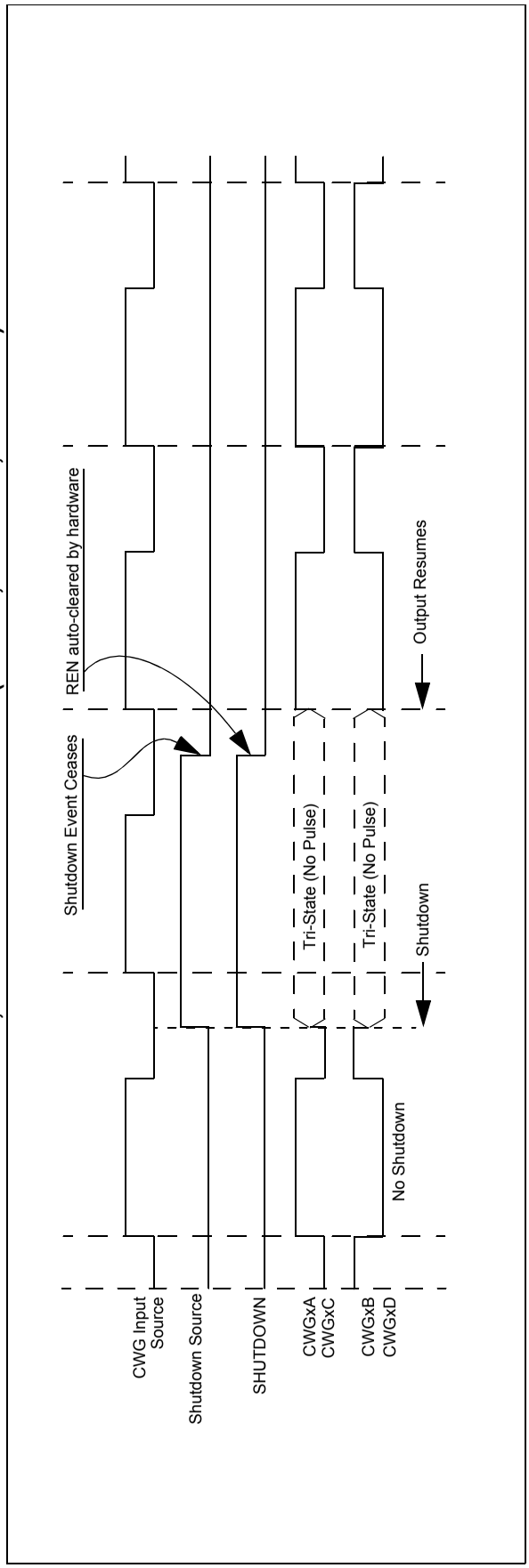


TABLE 27-1: CLCx DATA INPUT SELECTION

DyS<5:0> Value	CLCx Input Source
111111 [63]	Reserved
•	
•	
•	
110110 [55]	
110110 [54]	CAN_tx1
110101 [53]	CAN_tx0
110100 [52]	CWG3B_out
110011 [51]	CWG3A_out
110010 [50]	CWG2B_out
110001 [49]	CWG2A_out
110000 [48]	CWG1B_out
101111 [47]	CWG1A_out
101110 [46]	SS1
101101 [45]	SCK1
101100 [44]	SDO1
101011 [43]	Reserved
101010 [42]	UART2_tx_out
101001 [41]	UART1_tx_out
101000 [40]	CLC4_out
100111 [39]	CLC3_out
100110 [38]	CLC2_out
100101 [37]	CLC1_out
100100 [36]	DSM1_out
100011 [35]	IOC_flag
100010 [34]	ZCD_out
100001 [33]	CMP2_out
100000 [32]	CMP1_out
011111 [31]	NCO1_out
011110 [30]	Reserved
011101 [29]	Reserved
011100 [28]	PWM8_out
011011 [27]	PWM7_out
011010 [26]	PWM6_out
011001 [25]	PWM5_out
011000 [24]	CCP4_out
010111 [23]	CCP3_out
010110 [22]	CCP2_out
010101 [21]	CCP1 _out
010100 [20]	SMT2_out
010011 [19]	SMT1_out
010010 [18]	TMR6_out

TABLE 27-1: CLCx DATA INPUT SELECTION (CONTINUED)

DyS<5:0> Value	CLCx Input Source
010001 [17]	TMR5 _overflow
010000 [16]	TMR4 _out
001111 [15]	TMR3 _overflow
001110 [14]	TMR2 _out
001101 [13]	TMR1 _overflow
001100 [12]	TMR0 _overflow
001011 [11]	CLKR _out
001010 [10]	ADCRC
001001 [9]	SOSC
001000 [8]	MFINTOSC (32 kHz)
000111 [7]	MFINTOSC (500 kHz)
000110 [6]	LFINTOSC
000101 [5]	HFINTOSC
000100 [4]	FOSC
000011 [3]	CLCIN3PPS
000010 [2]	CLCIN2PPS
000001 [1]	CLCIN1PPS
000000 [0]	CLCIN0PPS

REGISTER 28-5: NCO1ACCU: NCO1 ACCUMULATOR REGISTER – UPPER BYTE(1)

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	ACC<19:16>			
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 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **ACC<19:16>:** NCO1 Accumulator, Upper Byte

Note 1: The accumulator spans registers NCO1ACCU:NCO1ACCH: NCO1ACCL. The 24 bits are reserved but not all are used.This register updates in real time, asynchronously to the CPU; there is no provision to guarantee atomic access to this 24-bit space using an 8-bit bus. Writing to this register while the module is operating will produce undefined results.

REGISTER 28-6: NCO1INCL: NCO1 INCREMENT REGISTER – LOW BYTE(1,2)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
INC<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 **INC<7:0>:** NCO1 Increment, Low Byte

Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

2: NCO1INC is double-buffered as INCBUF; INCBUF is updated on the next falling edge of NCOCLK after writing to NCO1INCL; NCO1INCU and NCO1INCH should be written prior to writing NCO1INCL.

REGISTER 28-7: NCO1INCH: NCO1 INCREMENT REGISTER – HIGH BYTE(1)

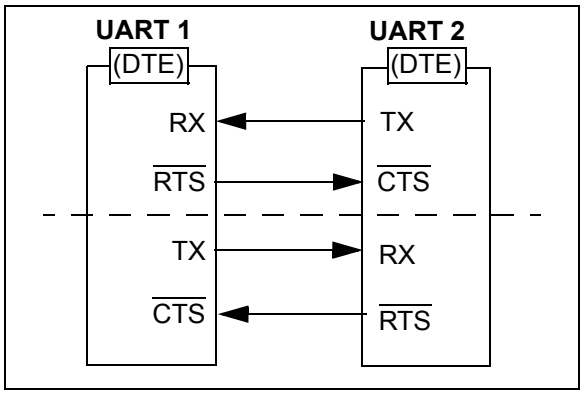
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
INC<15:8>							
bit 7							bit 0

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 **INC<15:8>:** NCO1 Increment, High Byte

Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

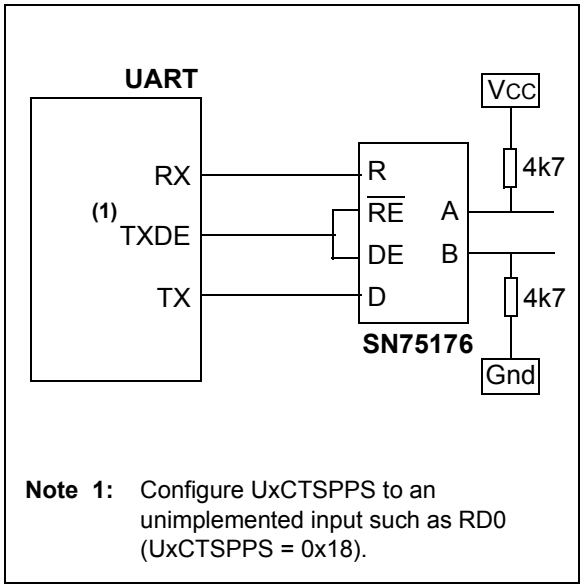
FIGURE 31-10: FLOW CONTROL



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.

REGISTER 31-7: UxFIFO: UART FIFO STATUS REGISTER

R/W/S-0/0	R/W-0/0	R/W/S/C-1/1	R/S/C-0/0	R/S/C-1/1	S/C-1/1	R/W/S/C-1/1	R/S/C-0/0
TXWRE	STPMD	TXBE	TXBF	RXIDL	XON	RXBE	RXBF
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	S = Hardware set	C = Hardware clear

bit 7 **TXWRE:** Transmit Write Error Status bit (Must be cleared by software)

LIN Master mode:

1 = UxP1L was written when a master process was active

LIN Slave mode:

1 = UxTXB was written when UxP2 = 0 or more than UxP2 bytes have been written to UxTXB since last Break

Address Detect mode:

1 = UxP1L was written before the previous data in UxP1L was transferred to TX shifter

All modes:

1 = A new byte was written to UxTXB when the output FIFO was full
0 = No error

bit 6 **STPMD**: Stop Bit Detection Mode bit

1 = Assert UxRXIF at end of last Stop bit or end of first Stop bit when STP = 11
0 = Assert UxRXIF in middle of first Stop bit

bit 5 **TXBE:** Transmit Buffer Empty Status bit

1 = Transmit buffer is empty. Setting this bit will clear the transmit buffer and output shift register.
0 = Transmit buffer is not empty. Software cannot clear this bit.

bit 4 **TXBF**: Transmit Buffer Full Status bit

1 = Transmit buffer is full
0 = Transmit buffer is not full

bit 3 **RXIDL:** Receive Pin Idle Status bit

1 = Receive pin is in Idle state
0 = UART is receiving Start, Stop, Data, Auto-baud, or Break

bit 2 **XON:** Software Flow Control Transmit Enable Status bit

1 = Transmitter is enabled
0 = Transmitter is disabled

bit 1 **RXBE:** Receive Buffer Empty Status bit

1 = Receive buffer is empty. Setting this bit will clear the RX buffer(1)
0 = Receive buffer is not empty. Software cannot clear this bit.

bit 0 **RXBF:** Receive Buffer Full Status bit

1 = Receive buffer is full
0 = Receive buffer is not full

Note 1: The BSF instruction should not be used to set RXBE because doing so will clear a byte pending in the transmit shift register when the UxTXB register is empty. Instead, use the MOVWF instruction with a '0' in the TXBE bit location.

33.0 I²C MODULE

The device has two dedicated, independent I²C modules. Figure 33-1 is a block diagram of the I²C interface module. The figure shows both the Master and Slave modes together.

FIGURE 33-1: I²C MODULE BLOCK DIAGRAM

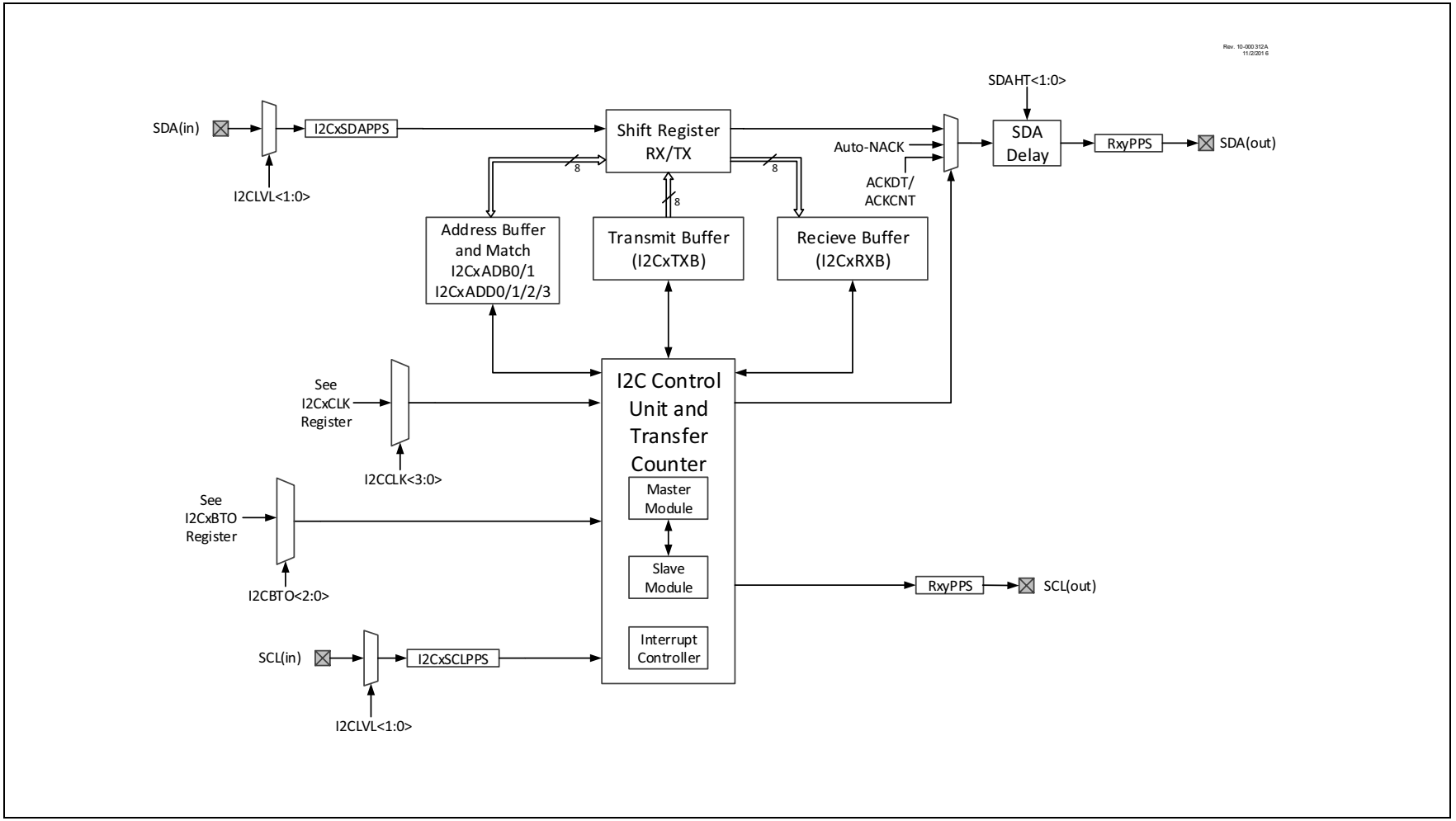


FIGURE 33-8: I^2C SLAVE, 7-BIT ADDRESS, RECEPTION NO I2CxCNT (ACKTIE = 0, ADRIE = 1, WRIE = 1)

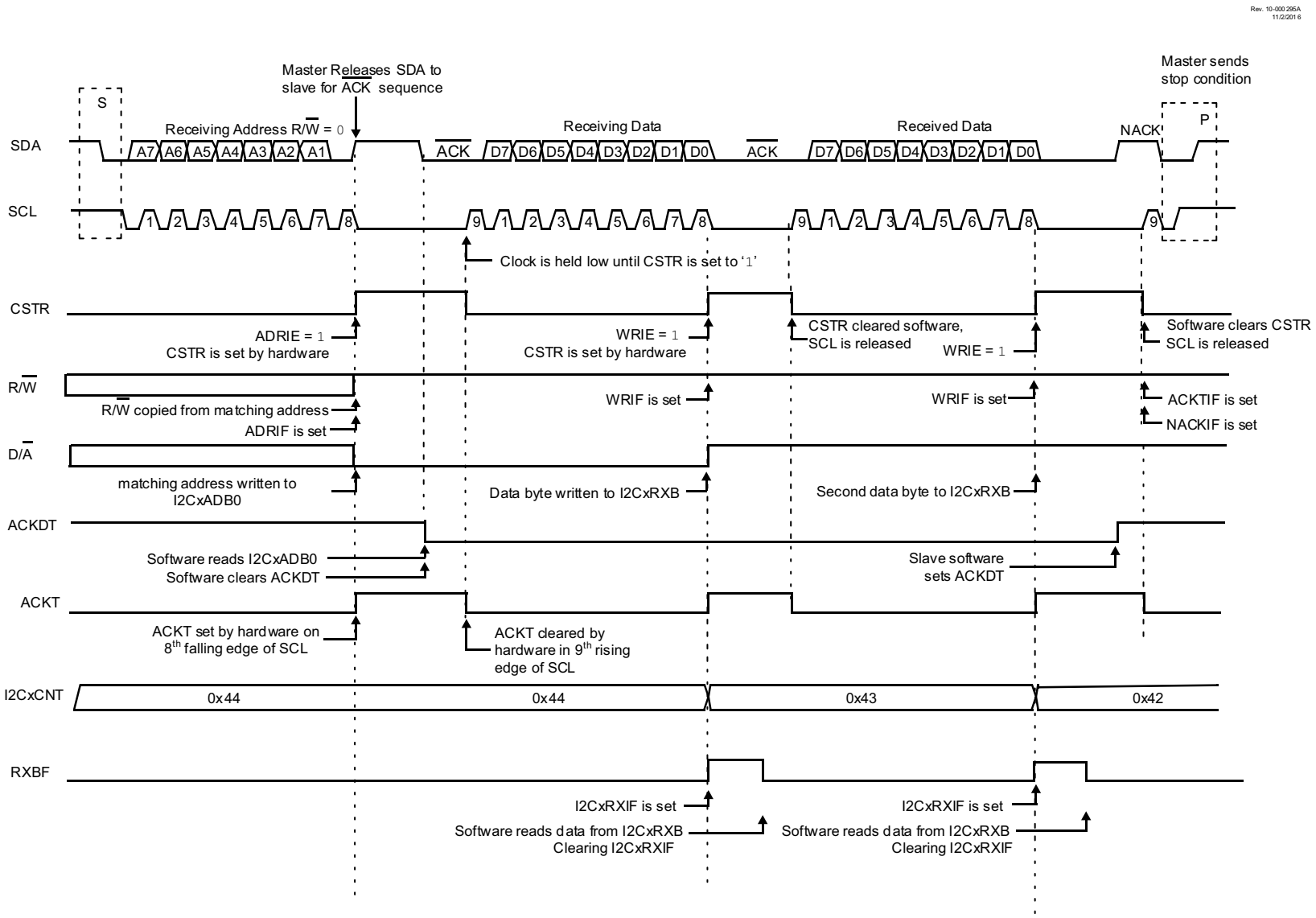
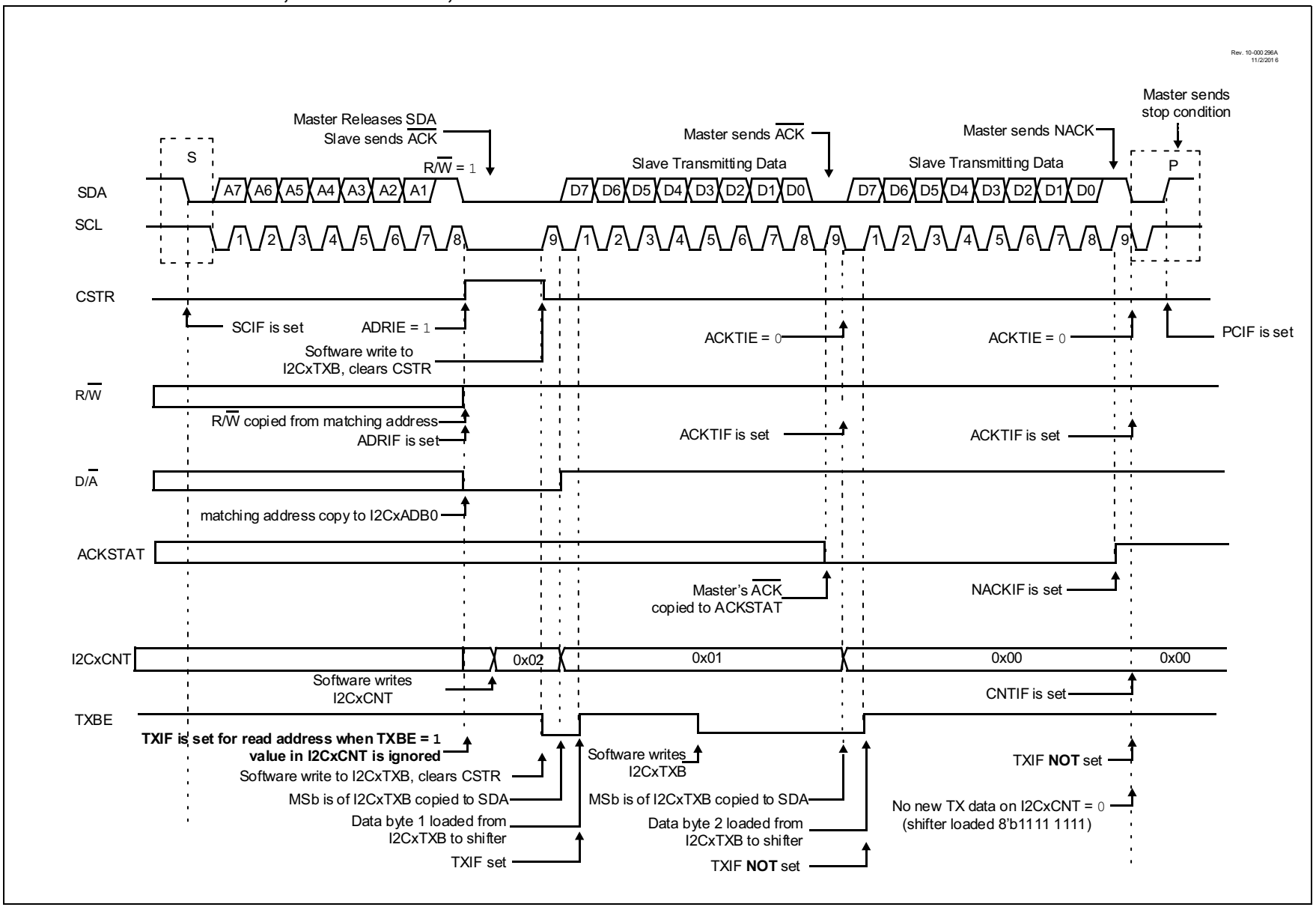


FIGURE 33-9: I^2C SLAVE, 7-BIT ADDRESS, TRANSMISSION



REGISTER 37-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0
PPOL	IPEN	GPOL	—	—	—	—	DSEN
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 **PPOL:** Precharge Polarity bit
If PRE>0x00:

PPOL	Action During 1st Precharge Stage	
	External (selected analog I/O pin)	Internal (AD sampling capacitor)
1	Connected to V_{DD}	C_{HOLD} connected to V_{SS}
0	Connected to V_{SS}	C_{HOLD} connected to V_{DD}

Otherwise:
The bit is ignored

bit 6 **IPEN:** A/D Inverted Precharge Enable bit
If DSEN = 1
1 = The precharge and guard signals in the second conversion cycle are the opposite polarity of the first cycle
0 = Both Conversion cycles use the precharge and guards specified by ADPPOL and ADGPOL
Otherwise:
The bit is ignored

bit 5 **GPOL:** Guard Ring Polarity Selection bit
1 = ADC guard Ring outputs start as digital high during Precharge stage
0 = ADC guard Ring outputs start as digital low during Precharge stage

bit 4-1 **Unimplemented:** Read as '0'

bit 0 **DSEN:** Double-sample enable bit
1 = Two conversions are performed on each trigger. Data from the first conversion appears in PREV
0 = One conversion is performed for each trigger

CPFSLT **Compare f with W, skip if f < W**

Syntax: CPFSLT f {,a}

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

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```

Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1] then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>;

If [W<7:4> + DC > 9] or [C = 1] then
(W<7:4>) + 6 + DC → W<7:4> ;
else
(W<7:4>) + DC → W<7:4>

Status Affected: C

Encoding:

0000	0000	0000	0111

Description: DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register W	Process Data	Write W

Example1:
```
DAW
```
Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

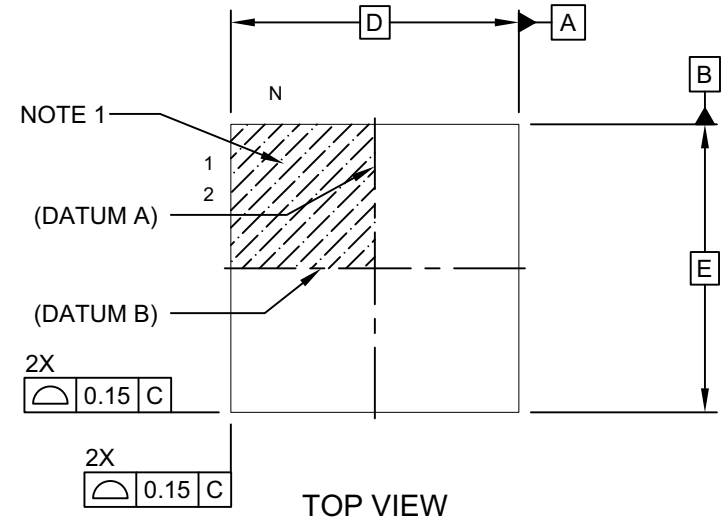
Example 2:
Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

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

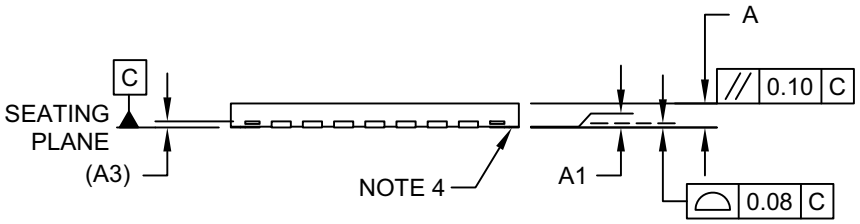
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
37D2h	TXB1SIDL	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	610
37D1h	TXB1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	610
37D0h	TXB1CON	TXBIF	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	609
37CFh	CANCON_R03	CANCON_RO3								603
37CEh	CANSTAT_R03	CANSTAT_RO3								604
37CDh	TXB2D7	TXB2D7								611
37CCh	TXB2D6	TXB2D6								611
37CBh	TXB2D5	TXB2D5								611
37CAh	TXB2D4	TXB2D4								611
37C9h	TXB2D3	TXB2D3								611
37C8h	TXB2D2	TXB2D2								611
37C7h	TXB2D1	TXB2D1								611
37C6h	TXB2D0	TXB2D0								611
37C5h	TXB2DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	612
37C4h	TXB2EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	611
37C3h	TXB2EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	610
37C2h	TXB2SIDL	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	610
37C1h	TXB2SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	610
37C0h	TXB2CON	TXBIF	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI1	TXPRI0	609
37BFh	RXM1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	633
37BEh	RXM1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	632
37BDh	RXM1SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	632
37BCh	RXM1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	631
37BBh	RXM0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	633
37BAh	RXM0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	632
37B9h	RXM0SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	631
37B8h	RXM0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	631
37B7h	RXF5EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37B6h	RXF5EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37B5h	RXF5SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
37B4h	RXF5SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
37B3h	RXF4EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37B2h	RXF4EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37B1h	RXF4SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
37B0h	RXF4SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
37AFh	RXF3EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37AEh	RXF3EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37ADh	RXF3SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
37ACh	RXF3SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
37ABh	RXF2EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37AAh	RXF2EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37A9h	RXF2SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
37A8h	RXF2SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
37A7h	RXF1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37A6h	RXF1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37A5h	RXF1SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630
37A4h	RXF1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
37A3h	RXF0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	631
37A2h	RXF0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	631
37A1h	RXF0SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	630

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

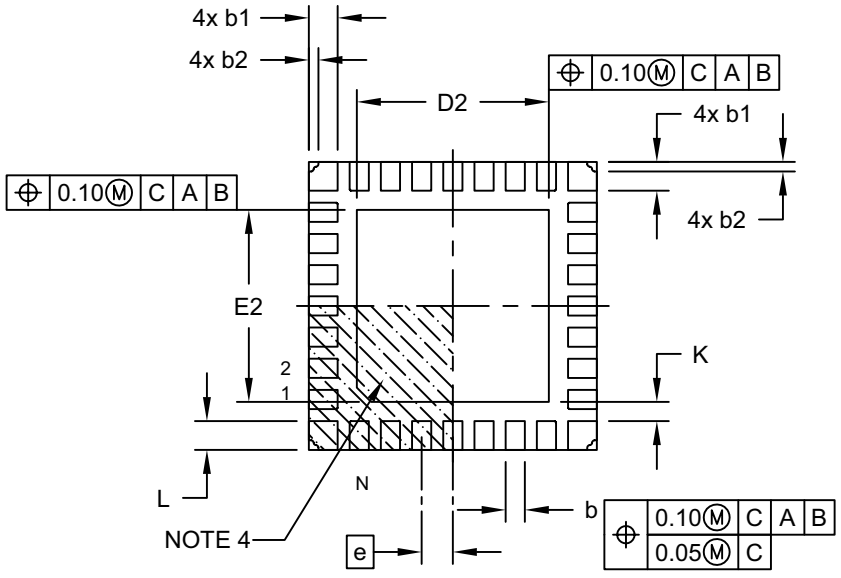
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



BOTTOM VIEW

Microchip Technology Drawing C04-0209 Rev C Sheet 1 of 2