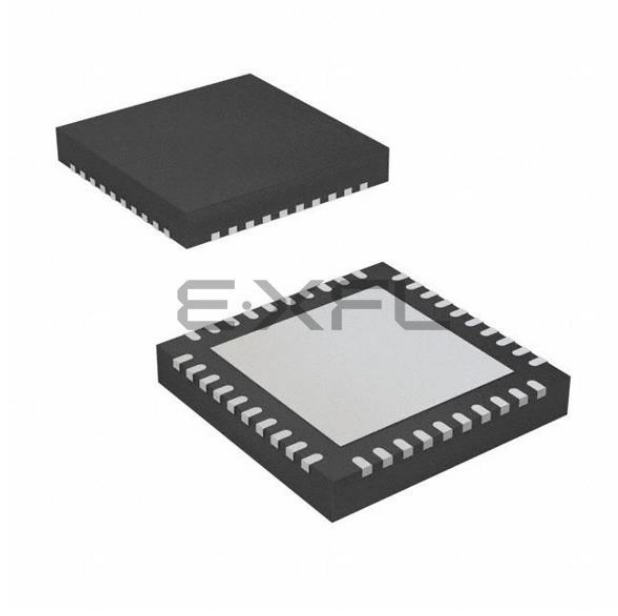


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf47k42t-i-mv

3.1 System Arbitration

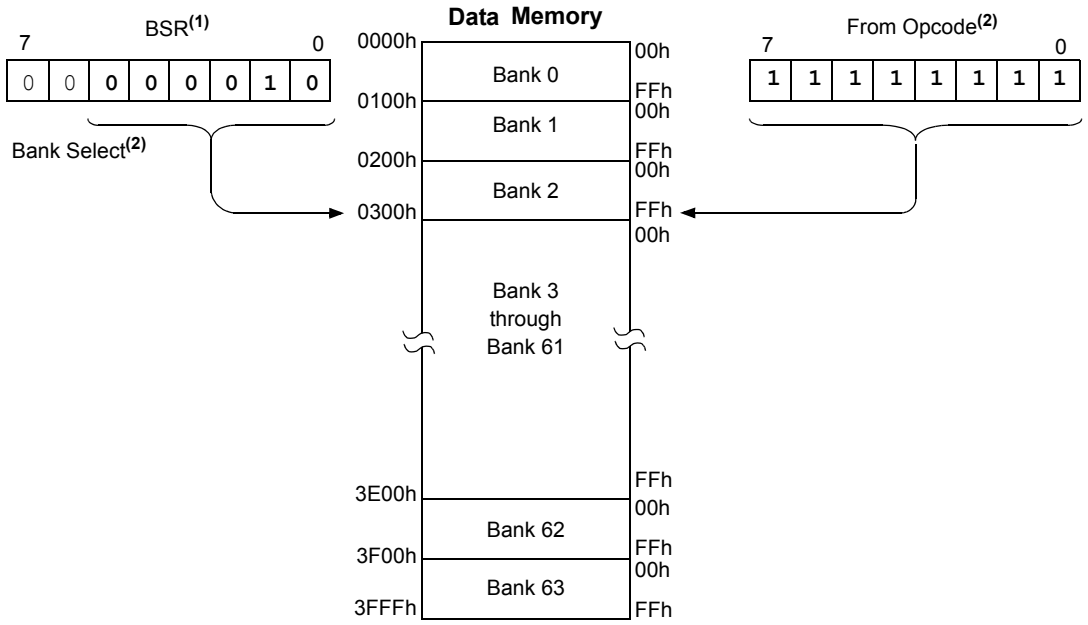
The System Arbiter resolves memory access between the System Level Selections (i.e., Main, Interrupt Service Routine) and Peripheral Selection (i.e., DMA and Scanner) based on user-assigned priorities. Each of the system level and peripheral selections has its own priority selection registers. Memory access priority is resolved using the number written to the corresponding Priority registers, 0 being the highest priority and 4 the lowest. The default priorities are listed in Table 3-1.

In case the user wants to change priorities, ensure each Priority register is written with a unique value from 0 to 4.

TABLE 3-1: DEFAULT PRIORITIES

Selection		Priority register Reset value
System Level	ISR	0
	MAIN	1
Peripheral	DMA1	2
	DMA2	3
	SCANNER	4

FIGURE 4-5: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



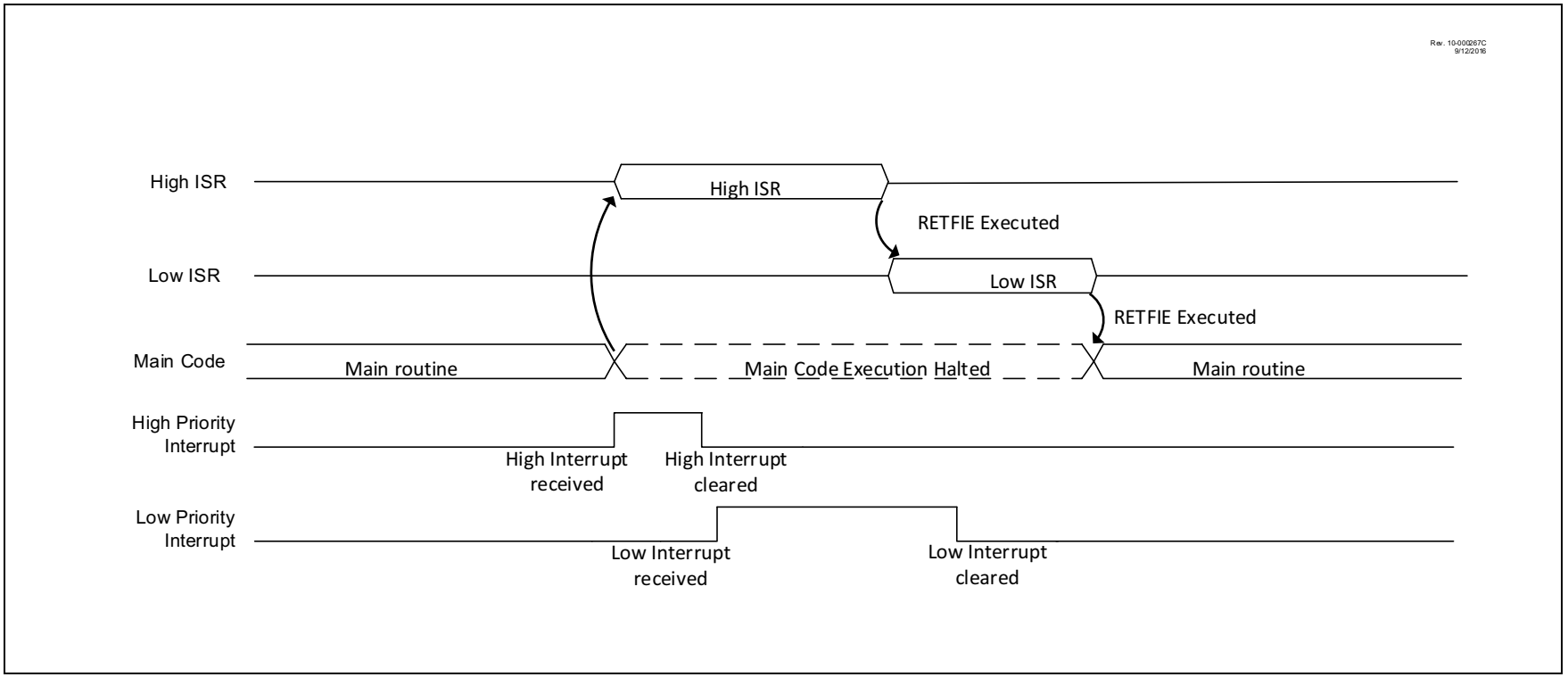
Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<5:0>) to the registers of the Access Bank.

9.4.2 SERVING A HIGH PRIORITY INTERRUPT WHILE A LOW PRIORITY INTERRUPT PENDING

A high priority interrupt request will always take precedence over any interrupt of a lower priority. The high priority interrupt is acknowledged first, then the low-priority interrupt is acknowledged. Upon a return from the high priority ISR (by executing the `RETFIE` instruction), the low priority interrupt is serviced, see Figure 9-3.

If any other high priority interrupts are pending and enabled, then they are serviced before servicing the pending low priority interrupt. If no other high priority interrupt requests are active, the low priority interrupt is serviced.

FIGURE 9-3: INTERRUPT EXECUTION: HIGH PRIORITY INTERRUPT WITH A LOW PRIORITY INTERRUPT PENDING



REGISTER 9-28: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP
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 **TMR0IP:** TMR0 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **U1IP:** UART1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **U1EIP:** UART1 Framing Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **U1TXIP:** UART1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **U1RXIP:** UART1 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **I2C1EIP:** I^2C1 Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **I2C1IP:** I^2C1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **I2C1TXIP:** I^2C1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

13.1.4 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- PFM Row Erase
- Write of PFM write latches to PFM memory
- Write of PFM write latches to User IDs
- Write to Data EEPROM Memory
- Write to Configuration Words

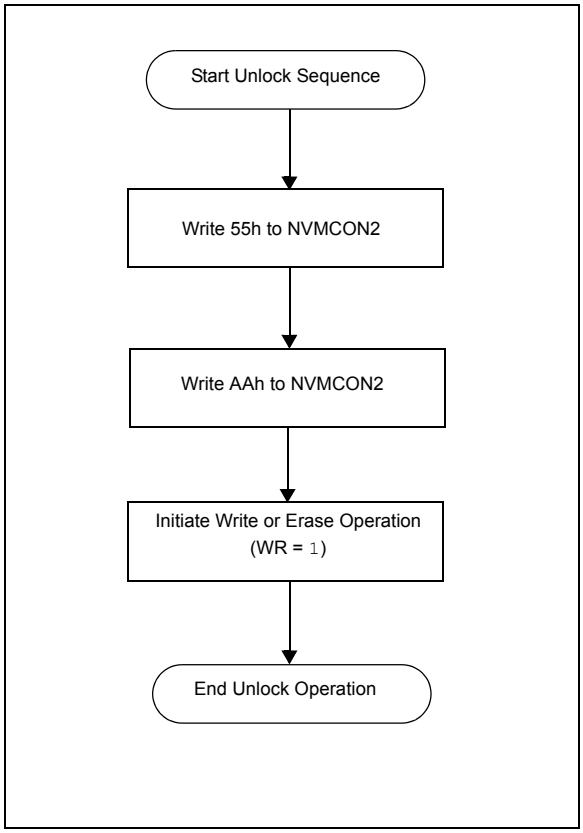
The unlock sequence consists of the following steps and must be completed in order:

- Write 55h to NVMCON2
- Write AAh to NMVCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 13-6: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 13-2: NVM UNLOCK SEQUENCE

```
BCF                 INTCON0,GIE             ; Recommended so sequence is not interrupted
BANKSEL             NVMCON1
BSF                 NVMCON1,WREN            ; Enable write/erase
MOVLW               55h                     ; Load 55h

MOVWF               NVMCON2                 ; Step 1: Load 55h into NVMCON2
MOVLW               AAh                     ; Step 2: Load W with AAh
MOVWF               NVMCON2                 ; Step 3: Load AAh into NVMCON2
BSF                 INTCON1,WR              ; Step 4: Set WR bit to begin write/erase

BSF                 INTCON0,GIE             ; Re-enable interrupts
```

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown. If the timing of the steps 1 to 4 is corrupted by an interrupt or a debugger Halt, the action will not take place.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

14.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE WITH MEMORY SCANNER

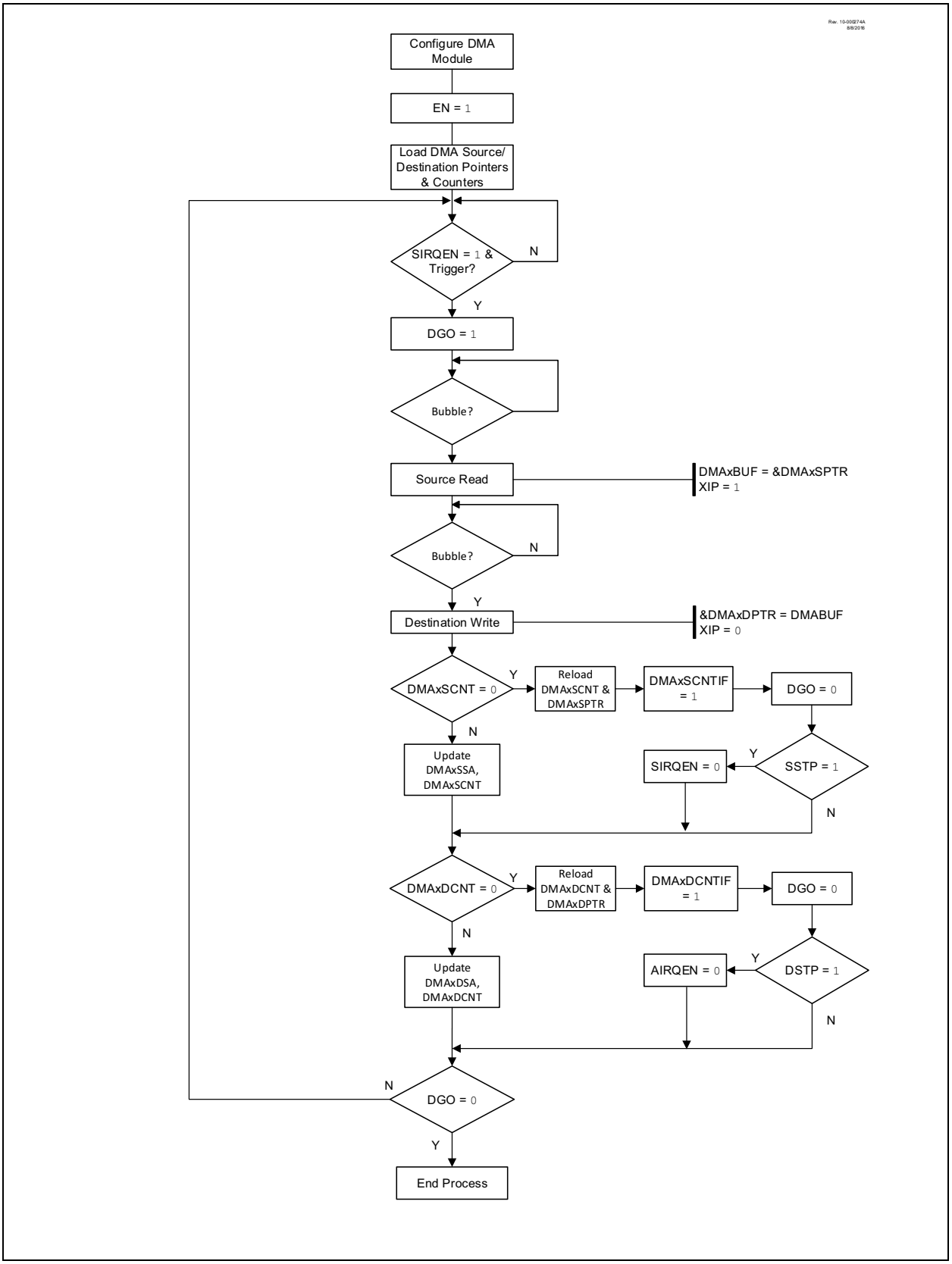
The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program/Data EEPROM memory user data
- Software loadable data registers for communication CRC's

14.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program/Data EEPROM memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using a scanner.

FIGURE 15-4: DMA OPERATION WITH HARDWARE TRIGGER



16.2.1 DATA REGISTER

PORTx is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISx (Register 16-2). Setting a TRISx bit ('1') will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISx bit ('0') will make the corresponding PORTx pin an output (i.e., it enables output driver and puts the contents of the output latch on the selected pin). Example 16-1 shows how to initialize PORTx.

Reading the PORTx register (Register 16-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATx).

The PORT data latch LATx (Register 16-3) holds the output port data and contains the latest value of a LATx or PORTx write.

EXAMPLE 16-1: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL  PORTA        ;
CLRF     PORTA        ;Init PORTA
BANKSEL  LATA         ;Data Latch
CLRF     LATA         ;
BANKSEL  ANSELA       ;
CLRF     ANSELA       ;digital I/O
BANKSEL  TRISA        ;
MOVLW    B'11111000'  ;Set RA<7:3> as inputs
MOVWF    TRISA        ;and set RA<2:0> as
                      ;outputs
```

16.2.2 DIRECTION CONTROL

The TRISx register (Register 16-2) controls the PORTx pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISx register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

16.2.3 ANALOG CONTROL

The ANSELx register (Register 16-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELx bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELx bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELx bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software.

16.2.4 OPEN-DRAIN CONTROL

The ODCONx register (Register 16-6) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONx bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONx bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note:	It is necessary to set open-drain control when using the pin for I^2C.

16.2.5 SLEW RATE CONTROL

The SLRCONx register (Register 16-7) controls the slew rate option for each port pin. Slew rate for each port pin can be controlled independently. When an SLRCONx bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONx bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

REGISTER 16-8: INLVLx: INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLx7	INLVLx6	INLVLx5	INLVLx4	INLVLx3	INLVLx2	INLVLx1	INLVLx0
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 **INLVLx<7:0>**: Input Level Select on Pins Rx<7:0>, respectively
1 = ST input used for port reads and interrupt-on-change
0 = TTL input used for port reads and interrupt-on-change

TABLE 16-9: INPUT LEVEL PORT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2(1)	INLVLB1(1)	INLVLB0
INLVLC	INLVLC7	INLVLC6	INLVLC5	INLVLC4(1)	INLVLC3(1)	INLVLC2	INLVLC1	INLVLC0
INLVLD(2)	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1(1)	INLVLD0(1)
INLVLE	—	—	—	—	INLVLE3	INLVLE2(2)	INLVLE1(2)	INLVLE0(2)
INLVLF(3)	INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0

Note 1: Any peripheral using the I^2C pins read the I^2C ST inputs when enabled via RxyI2C.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented in PIC18(L)F26/27/45/46/47K42.

FIGURE 25-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

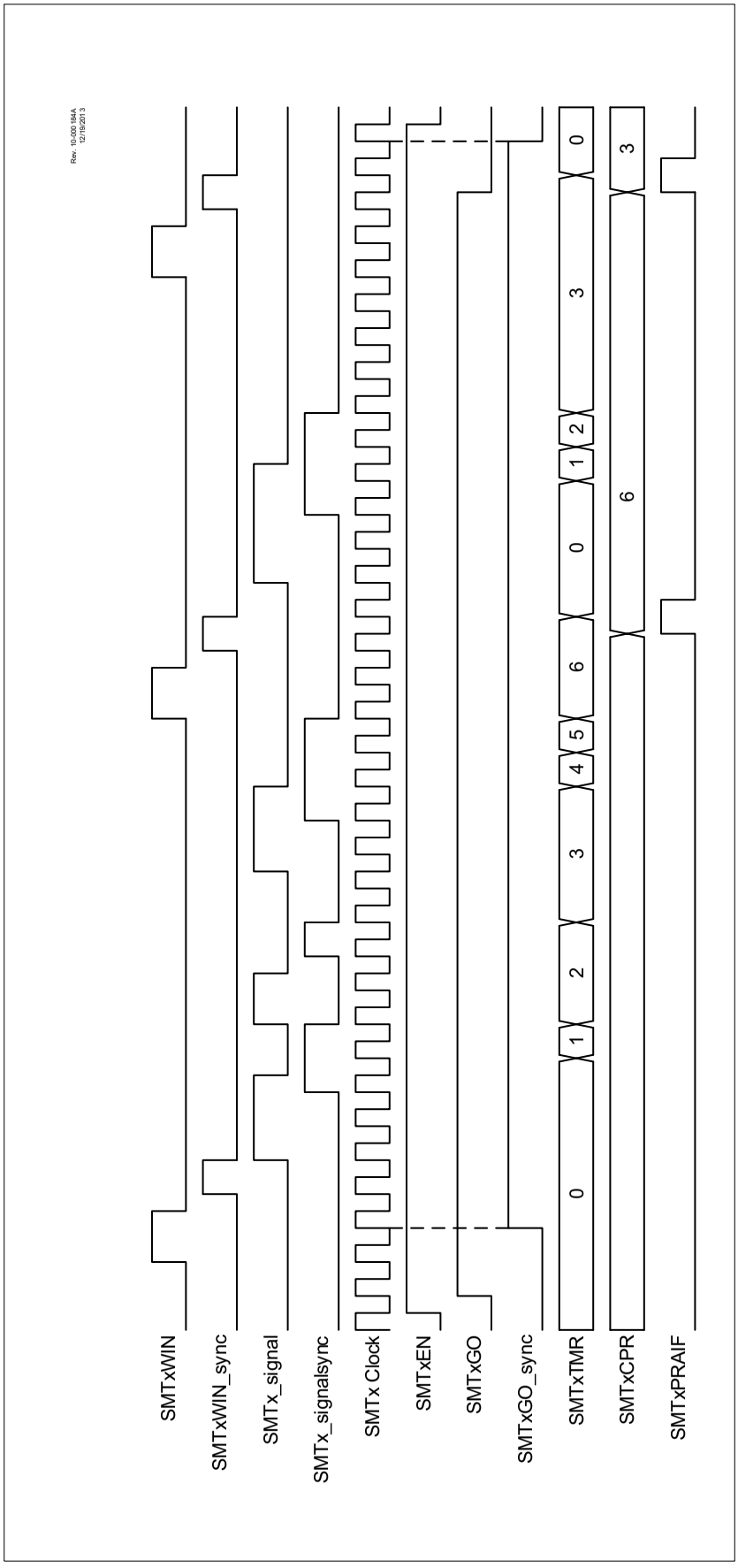


TABLE 25-3: SUMMARY OF REGISTERS ASSOCIATED WITH SMT1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		394
SMT1CON1	GO	REPEAT	—	—	MODE<3:0>				395
SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	396
SMT1CLK	—	—	—	—	—	CSEL<2:0>			397
SMT1SIG	—	—	—	SSEL<4:0>					399
SMT1WIN	—	—	—	WSEL<4:0>					398
SMT1TMRL	TMR<7:0>								400
SMT1TMRH	TMR<15:8>								400
SMT1TMRU	TMR<23:16>								400
SMT1CPRL	CPR<7:0>								401
SMT1CPRH	CPR<15:8>								401
SMT1CPRU	CPR<23:16>								401
SMT1CPWL	CPW<7:0>								402
SMT1CPWH	CPW<15:8>								402
SMT1CPWU	CPW<23:16>								402
SMT1PRL	PR<7:0>								403
SMT1PRH	PR<15:8>								403
SMT1PRU	PR<23:16>								403

Legend: — = unimplemented read as '0'. Shaded cells are not used for SMT1 module.

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

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PWS<2:0>(1,2)			—	CKS<3:0>			
bit 7							bit 0

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

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

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

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

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

32.4.3 TRANSFER COUNTER IN SLAVE MODE

In Slave Mode, the transfer counter will still decrement as data is shifted in and out of the SPI module, but it will not control data transfers. In addition, in slave mode, the BMODE bit along with the transfer counter is used to determine when the device should look for Slave Select faults. If BMODE = 0, the SSFLT bit will be set if Slave Select transitions from its active to inactive state during bytes of data, as well as if it transitions before the last bit sent during the final byte (if SPIx-TWIDTH≠0). If BMODE=1, the SSFLT bit will be set if Slave Select transitions from its active to inactive state before the final bit of each individual transfer is completed. Note that SSFLT does not have an associated interrupt, so it should be checked in software. An ideal time to do this is when the End of Slave Select Interrupt (EOSIF) is triggered (see **Section 32.8.3.3 "Start of Slave Select and End of Slave Select Interrupts"**).

32.5 Master mode

In master mode, the device controls the SCK line, and as such, initiates data transfers and determines when any slaves broadcast data onto the SPI bus.

Master mode of this device can be configured in four different modes, configured by the TXR and RXR bits:

- Full Duplex mode
- Receive Only mode
- Transmit Only mode
- Transfer Off mode

The modes are illustrated in Table 32-1, below:

TABLE 32-1: MASTER MODE TXR/RXR SETTINGS

	TXR = 1	**TXR = 0**
RXR = 1	Full Duplex Mode If BMODE = 1, transfer when RxFIFO is not full and TxFIFO is not empty If BMODE = 0, Transfer when RXFIFO is not full, TXFIFO is not empty, and the Transfer Counter is non-zero	Receive Only mode Transfer when RxFIFO is not full and the Transfer Counter is non-zero Transmitted data is either the top of the FIFO or the most recently received data
RXR = 0	Transmit Only Mode If BMODE = 1, transfer when TxFIFO is not empty If BMODE = 0, Transfer when TXFIFO is not empty and the Transfer Counter is non-zero Received data is not stored	No Transfers

REGISTER 32-12: SPIxTxB: SPI TRANSMIT BUFFER REGISTER

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-0 **TXB<7:0>**: Transmit Buffer bits (write only)

If TXFIFO is not full:

Writing to this register adds the data to the top of the TXFIFO and increases the occupancy of the TXFIFO write pointer

If TXFIFO is full:

Writing to this register does not affect the data in the TXFIFO or the write pointer, and the TXWE bit of SPIxSTATUS will be set

REGISTER 32-13: SPIxCLK: SPI CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CLKSEL3	CLKSEL2	CLKSEL1	CLKSEL0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

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

bit 3-0 **CLKSEL<3:0>**: SPI Clock Source Selection bits

1111-1001 = Reserved
1000 = SMT_match
0111 = TMR6_Postscaled
0110 = TMR4_Postscaled
0101 = TMR2_Postscaled
0100 = TMR0_overflow
0011 = CLKREF
0010 = MFINTOSC
0001 = HFINTOSC
0000 = FOSC

FIGURE 33-6: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (ACKTIE = 0, ADRIE = 0, WRIE = 0)

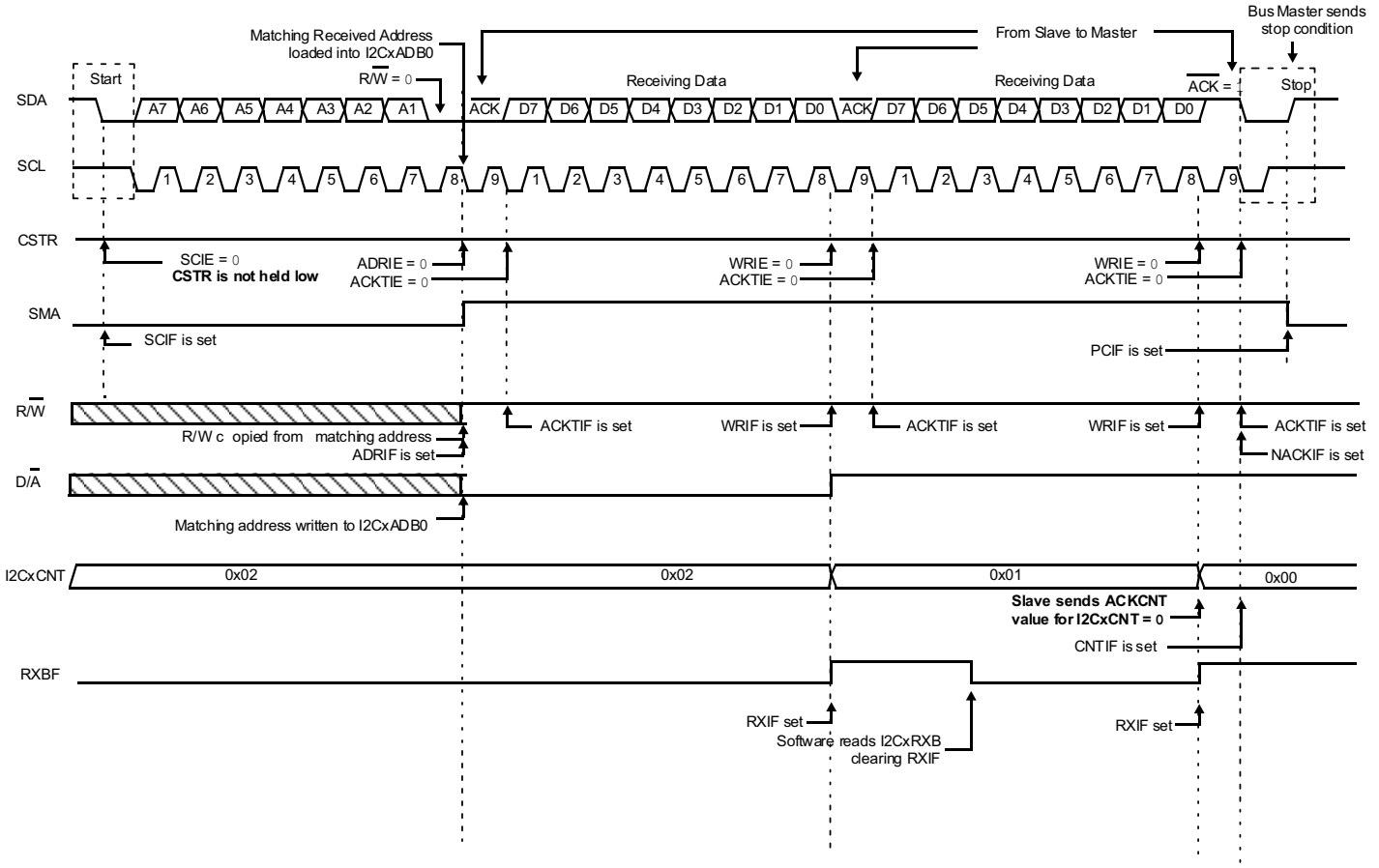
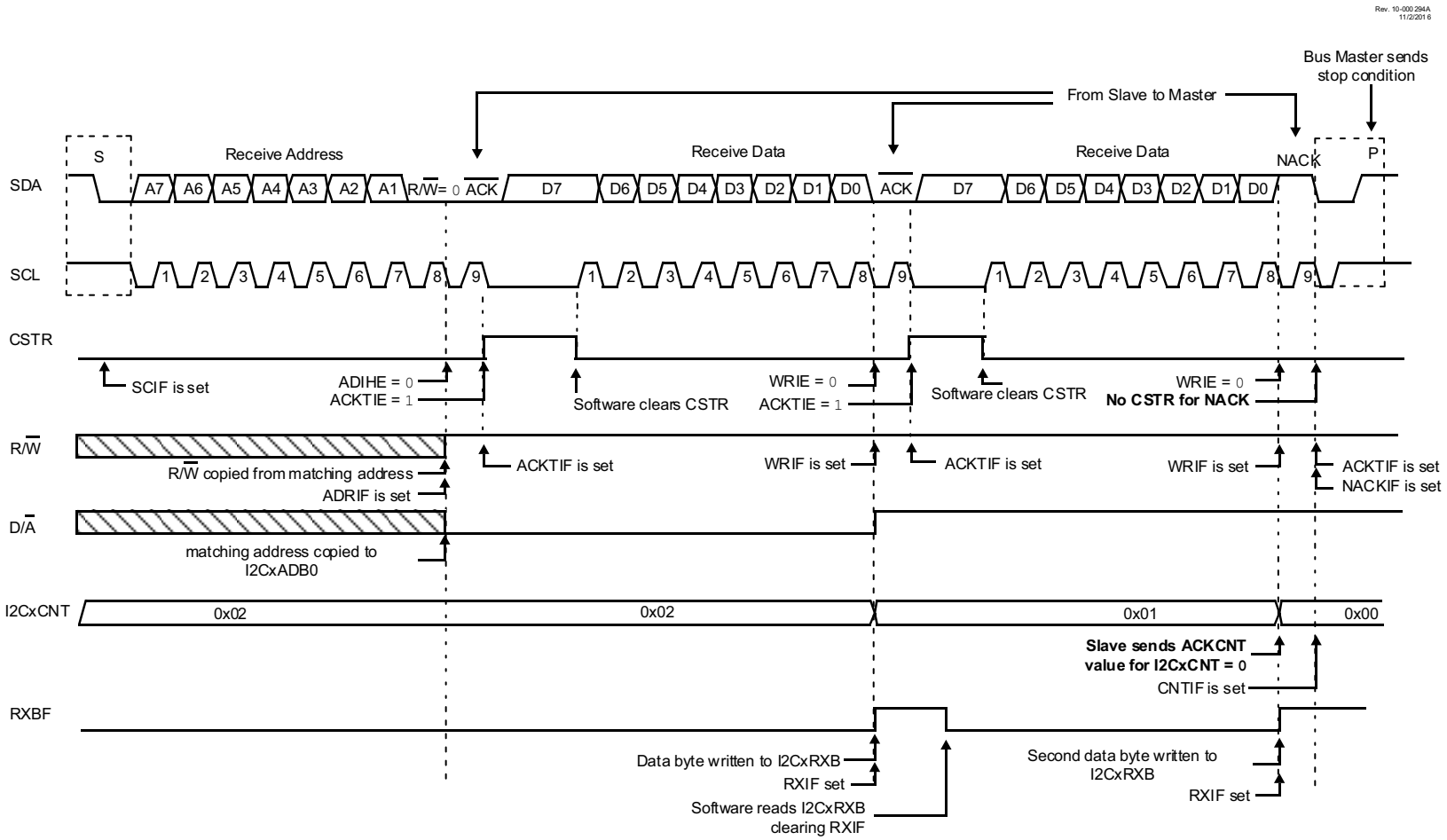


FIGURE 33-7: I²C SLAVE, 7-BIT ADDRESS, RECEPTION WITH I2CxCNT (ACKTIE = 1, ADRIE = 0, WRIE = 0)



bit 3 **MDR:** Master Data Request *(Master pause)*
1 = Master state mechine pauses until data is read/written to proceed (SCL is output held low)
0 = Master clocking of data is enabled.

MMA = 1 & RXBF = 1
pause_for_rx - Set by hardware on 7th falling SCL edge
- User must read from I2CRXB to release SCL
MMA = 1 & TXBE = 1 & I2CCNT!= 0
pause_for_tx - Set by hardware on 8th falling SCL edge
- User must write to I2CTXB to release SCL
ADB = 1
- I2CCNT is ignored for the high and low address in 10-bit mode
pause_for_restart - Set by hardware on 9th falling SCL edge
RSEN = 1 & MMA = 1 && I2CCNT = 0 || ACKSTAT = 1
- User must set START or write to I2CTXB to release SCL and shift Restart onto bus

bit 2-0 **MODE<2:0>:** I^2C Mode Select bits
111 = I^2C Muti-Master mode (SMBus 2.0 Host), **(5)**
Works as both mode<2:0> = 001 and mode<2:0> = 100
110 = I^2C Muti-Master mode (SMBus 2.0 Host), **(5)**
Works as both mode<2:0> = 000 and mode<2:0> = 100
101 = I^2C Master mode, 10-bit address
100 = I^2C Master mode, 7-bit address
011 = I^2C Slave mode, one 10-bit address with masking
010 = I^2C Slave mode, two 10-bit address
001 = I^2C Slave mode, two 7-bit address with masking
000 = I^2C Slave mode, four 7-bit address

Note 1: SDA and SCL pins must be configured for open-drain with internal or external pull-up
2: SDA and SCL pins must be selected as both input and output in PPS
3: CSTR can be set by more than one hardware source, all sources must be addressed by user software before the SCL line is released. CSTR is a module status bit, and does not show the true bus state.
4: SMA is set on the same SCL edge as CSTR for a matching received address
5: In this mode, ADRIE should be set, this allows an interrupt to clear the BCLIF condition and allow the ACK of matching address.
6: In 10-bit Slave mode, when ADB = 1, CSTR will set when the high address has not been read out of I2CxRXB before the low address is shifted in.

34.3 Register Definitions: FVR Control

REGISTER 34-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	RDY(1)	TSEN(3)	TSRNG(3)	CDAFVR<1:0>		ADFVR<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	q = Value depends on condition

bit 7 **EN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **RDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = VOUT = 3VT (High Range)
0 = VOUT = 2VT (Low Range)

bit 3-2 **CDAFVR<1:0>:** Comparator FVR Buffer Gain Selection bits
11 = FVR Buffer 1 Gain is 4x, (4.096V)(2)
10 = FVR Buffer 1 Gain is 2x, (2.048V)(2)
01 = FVR Buffer 1 Gain is 1x, (1.024V)
00 = FVR Buffer 1 is off

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bit
11 = FVR Buffer 2 Gain is 4x, (4.096V)(2)
10 = FVR Buffer 2 Gain is 2x, (2.048V)(2)
01 = FVR Buffer 2 Gain is 1x, (1.024V)
00 = FVR Buffer 2 is off

Note 1: FVRRDY is always '1'.
2: Fixed Voltage Reference output cannot exceed VDD.
3: See **Section 35.0 "Temperature Indicator Module"** for additional information.

TABLE 34-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	EN	RDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		597

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used with the Fixed Voltage Reference.

41.2.2 EXTENDED INSTRUCTION SET

ADDULNK — Add Literal to FSR2 and Return

Syntax:	ADDULNK k
Operands:	$0 \le k \le 63$
Operation:	FSR2 + k → FSR2, (TOS) → PC
Status Affected:	None
Encoding:	1110 | 1000 | 11kk | kkkk
Description:	The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS. The instruction takes two cycles to execute; a NOP is performed during the second cycle. This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

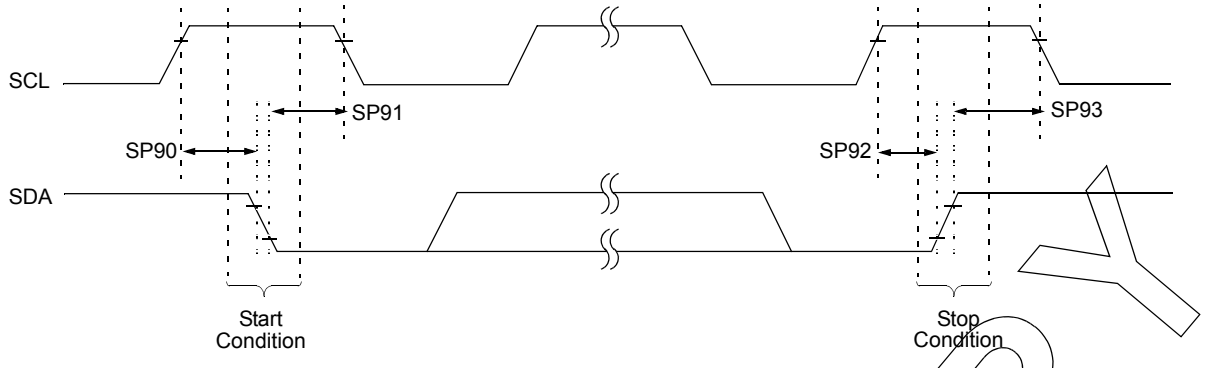
Example: ADDULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h

After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction syntax then becomes: {label} instruction argument(s).

FIGURE 44-18: I²C BUS START/STOP BITS TIMING



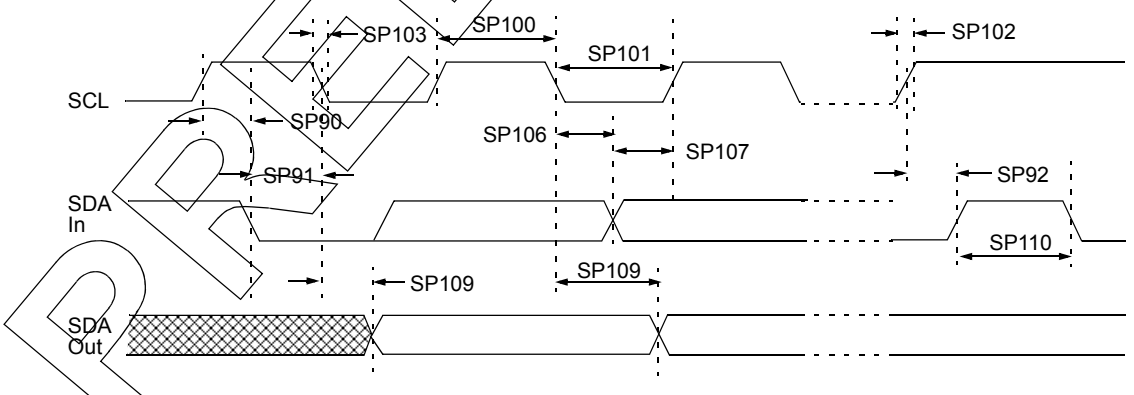
Note: Refer to Figure 44-4 for load conditions.

TABLE 44-25: I²C BUS START/STOP BITS REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)								
Param No.	**Symbol**	**Characteristic**		**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
SP90*	TSU:STA	Start condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
			1 MHz mode	260	—	—		
SP91*	THD:STA	Start condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
			1 MHz	260	—	—		
SP92*	TSU:STO	Stop condition Setup time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		
			1 MHz	260	—	—		
SP93	THD:STO	Stop condition Hold time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	1300	—	—		
			1 MHz	500	—	—		

* These parameters are characterized but not tested.

FIGURE 44-19: I²C BUS DATA TIMING



Note: Refer to Figure 44-4 for load conditions.