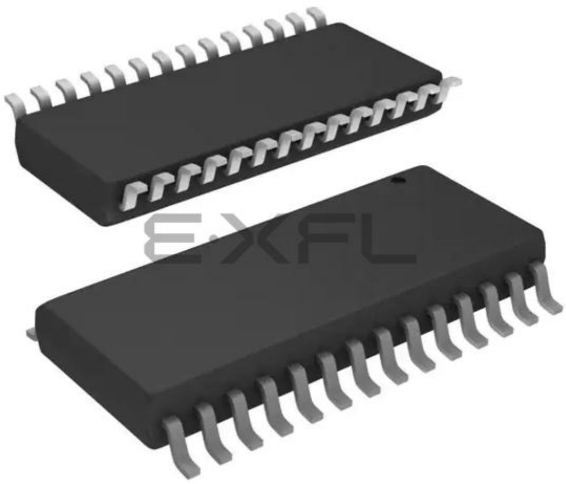


Welcome to E-XFL.COM

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	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K 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/pic18lf26k83t-i-so

TABLE 1-1: DEVICE FEATURES

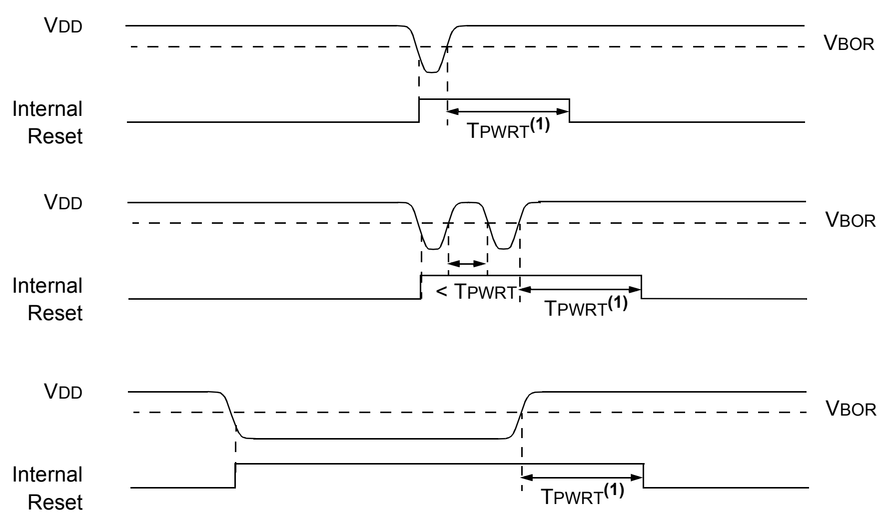
Features	PIC18(L)F25K83	PIC18(L)F26K83
Program Memory (Bytes)	32768	65536
Program Memory (Instructions)	16384	32768
Data Memory (Bytes)	2048	4096
Data EEPROM Memory (Bytes)	1024	1024
Packages	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN
I/O Ports	A,B,C,E(1)	A,B,C,E(1)
12-Bit Analog-to-Digital Conversion Module (ADC^2) with Computation Accelerator	5 internal 24 external	5 internal 24 external
Capture/Compare/PWM Modules (CCP)	4	
10-Bit Pulse-Width Modulator (PWM)	4	
Timers (16-/8-bit)	4/3	
Serial Communications	2 UARTs with DMX/DALI/LIN, 2 I^2C, 1 SPI	
Complementary Waveform Generator (CWG)	3	
Zero-Cross Detect (ZCD)	1	
Data Signal Modulator (DSM)	1	
Signal Measurement Timer (SMT)	2	
5-bit Digital to Analog Converter (DAC)	1	
Numerically Controlled Oscillator (NCO)	1	
Comparator Module	2	
Direct Memory Access (DMA)	2	
Configurable Logic Cell (CLC)	4	
Control Area Network (CAN)	Yes	
Peripheral Module Disable (PMD)	Yes	
16-bit CRC with Scanner	Yes	
Programmable High/Low-Voltage Detect (HLVD)	Yes	
Resets (and Delays)	POR, Programmable BOR, RESET Instruction, Stack Overflow, Stack Underflow (PWRT, OST), $\overline{MCLR}$, WDT, $\overline{MEMV}$	
Instruction Set	81 Instructions; 87 with Extended Instruction Set enabled	
Maximum Operating Frequency	64 MHz	

Note 1: PORTE contains the single RE3 input-only pin.

TABLE 6-1: BOR OPERATING MODES

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Instruction Execution upon:	
				Release of POR	Wake-up from Sleep
11	X	X	Active	Wait for release of BOR (BORRDY = 1)	Begins immediately
10	X	Awake	Active	Wait for release of BOR (BORRDY = 1)	N/A
		Sleep	Hibernate	N/A	Wait for release of BOR (BORRDY = 1)
01	1	X	Active	Wait for release of BOR (BORRDY = 1)	Begins immediately
	0	X	Hibernate		
00	X	X	Disabled	Begins immediately	

FIGURE 6-3: BROWN-OUT SITUATIONS



Note 1: TPWRT delay depends on PWRTS<1:0> Configuration bits.

11.1 Independent Clock Source

The WWDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of WDTE<1:0> Configuration bits.

If WDTE = `0b1x`, then the clock source will be enabled depending on the WDTCCS<2:0> Configuration bits.

If WDTE = `0b01`, the SEN bit should be set by software to enable WWDT, and the clock source is enabled by the CS bits in the WDTCON1 register.

Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section 45.0 “Electrical Specifications”** for LFINTOSC and MFINTOSC tolerances.

11.2 WWDT Operating Modes

The Windowed Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 11-1.

11.2.1 WWDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to ‘`11`’, the WWDT is always on.

WWDT protection is active during Sleep.

11.2.2 WWDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to ‘`10`’, the WWDT is on, except in Sleep.

WWDT protection is not active during Sleep.

11.2.3 WWDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to ‘`01`’, the WWDT is controlled by the SEN bit of the WDTCON0 register.

WWDT protection is unchanged by Sleep. See Table 11-1 for more details.

TABLE 11-1: WWDT OPERATING MODES

WDTE<1:0>	SEN	Device Mode	WWDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

11.3 Time-out Period

If the WDTCPS<4:0> Configuration bits default to `0b11111`, then the PS bits of the WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). If any value other than the default value is assigned to WDTCPS<4:0> Configuration bits, then the timer period will be based on the WDTCPS<4:0> bits in the CONFIG3L register. After a Reset, the default time-out period is 2s.

11.4 Watchdog Window

The Windowed Watchdog Timer has an optional Windowed mode that is controlled by the WDTCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the `CLRWDT` instruction must occur within the allowed window of the WDT period. Any `CLRWDT` instruction that occurs outside of this window will trigger a window violation and will cause a WWDT Reset, similar to a WWDT time out. See Figure 11-2 for an example.

The window size is controlled by the WINDOW<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCWS<2:0> = `111`.

The five Most Significant bits of the WDTTMR register are used to determine whether the window is open, as defined by the WINDOW<2:0> bits of the WDTCON1 register.

In the event of a window violation, a Reset will be generated and the $\overline{\text{WDTWV}}$ bit of the PCON0 register will be cleared. This bit is set by a POR or can be set in firmware.

11.5 Clearing the WWDT

The WWDT is cleared when any of the following conditions occur:

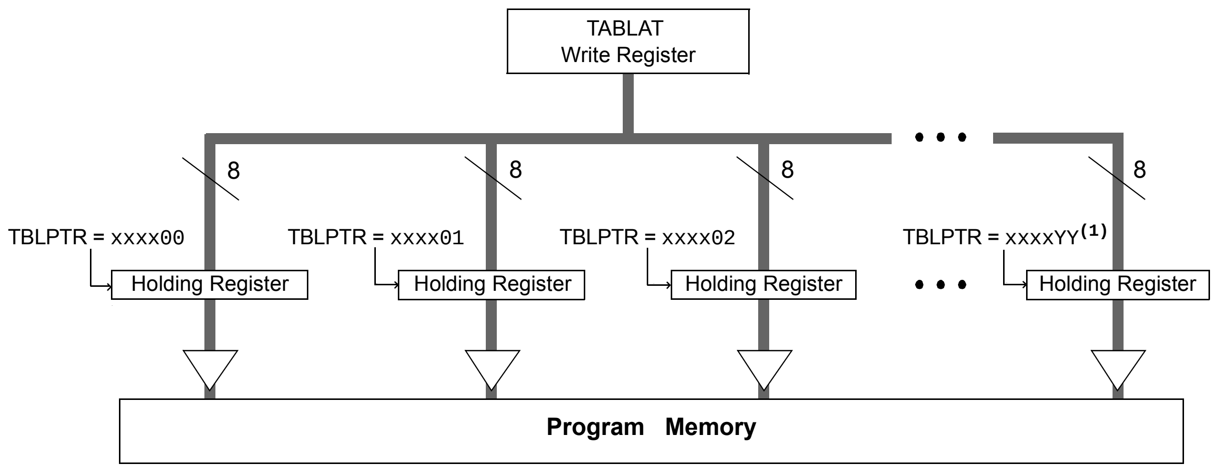
- Any Reset
- Valid `CLRWDT` instruction is executed
- Device enters Sleep
- Exit Sleep by Interrupt
- WWDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

11.5.1 `CLRWDT` CONSIDERATIONS (WINDOWED MODE)

When in Windowed mode, the WWDT must be armed before a `CLRWDT` instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a `CLRWDT` instruction without performing such an arming action will trigger a window violation regardless of whether the window is open or not.

See Table 11-2 for more information.

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 Data EEPROM Memory

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, which is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire V_{DD} range.

Four SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- NVMCON1
- NVMCON2
- NVMDAT
- NVMADRL
- NVMADRH

The data EEPROM allows byte read and write. When interfacing to the data memory block, NVMDAT holds the 8-bit data for read/write and the NVMADRH:NVMADRL register pair holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an internal programming timer; it will vary with voltage and temperature as well as from chip-to-chip. Refer to the Data EEPROM Memory parameters in **Section 45.0 “Electrical Specifications”** for limits.

13.3.1 NVMADRL AND NVMADRH REGISTERS

The NVMADRH:NVMADRL registers are used to address the data EEPROM for read and write operations.

13.3.2 NVMCON1 AND NVMCON2 REGISTERS

Access to the data EEPROM is controlled by two registers: NVMCON1 and NVMCON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The NVMCON1 register (Register 13-1) is the control register for data and program memory access. Control bits REG<1:0> determine if the access will be to program, Data EEPROM Memory or the User IDs, Configuration bits, Revision ID and Device ID.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear.

The WRERR bit is set by hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

The WR control bit initiates write operations. The bit can be set but not cleared by software. It is cleared only by hardware at the completion of the write operation.

The NVMIF Interrupt Flag bit of the PIR0 register is set when the write is complete. It must be cleared by software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (REG<1:0> = 0x10). Program memory is read using table read instructions. See **Section 13.1.1 “Table Reads and Table Writes”** regarding table reads.

REGISTER 15-2: DMAxCON1: DMAx CONTROL REGISTER1

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
DMODE<1:0>		DSTP	SMR<1:0>		SMODE<1:0>		SSTP
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

bit 7-6 **DMODE<1:0>:** Destination Address Mode Selection bits
- `11` = Reserved, Do not use
- `10` = DMAxDPTR<15:0> is decremented after each transfer completion
- `01` = DMAxDPTR<15:0> is incremented after each transfer completion
- `00` = DMAxDPTR<15:0> remains unchanged after each transfer completion

bit 5 **DSTP:** Destination Counter Reload Stop bit
- `1` = SIRQEN bit is cleared when Destination Counter reloads
- `0` = SIRQEN bit is not cleared when Destination Counter reloads

bit 4-3 **SMR[1:0]:** Source Memory Region Select bits
- `1x` = DMAxSSA<21:0> points to Data EEPROM
- `01` = DMAxSSA<21:0> points to Program Flash Memory
- `00` = DMAxSSA<21:0> points to SFR/GPR Data Space

bit 2-1 **SMODE[1:0]:** Source Address Mode Selection bits
- `11` = Reserved, Do not use
- `10` = DMAxSPTR<21:0> is decremented after each transfer completion
- `01` = DMAxSPTR<21:0> is incremented after each transfer completion
- `00` = DMAxSPTR<21:0> remains unchanged after each transfer completion

bit 0 **SSTP:** Source Counter Reload Stop bit
- `1` = SIRQEN bit is cleared when Source Counter reloads
- `0` = SIRQEN bit is not cleared when Source Counter reloads

24.2 Register Definitions: PWM Control

Long bit name prefixes for the PWM peripherals are shown below. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

Peripheral	Bit Name Prefix
PWM3	PWM3
PWM4	PWM4

REGISTER 24-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
EN	—	OUT	POL	—	—	—	—
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 **EN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **OUT:** PWM Module Output Level When Bit is Read

bit 4 **POL:** PWM Output Polarity Select bit
1 = PWM output is inverted
0 = PWM output is normal

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 27-7: CLCxGLS0: GATE 0 LOGIC SELECT REGISTER

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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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 **G1D4T:** Gate 0 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 0
0 = CLCIN3 (true) is not gated into CLCx Gate 0

bit 6 **G1D4N:** Gate 0 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 0
0 = CLCIN3 (inverted) is not gated into CLCx Gate 0

bit 5 **G1D3T:** Gate 0 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 0
0 = CLCIN2 (true) is not gated into CLCx Gate 0

bit 4 **G1D3N:** Gate 0 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 0
0 = CLCIN2 (inverted) is not gated into CLCx Gate 0

bit 3 **G1D2T:** Gate 0 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 0
0 = CLCIN1 (true) is not gated into l CLCx Gate 0

bit 2 **G1D2N:** Gate 0 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 0
0 = CLCIN1 (inverted) is not gated into CLCx Gate 0

bit 1 **G1D1T:** Gate 0 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 0
0 = CLCIN0 (true) is not gated into CLCx Gate 0

bit 0 **G1D1N:** Gate 0 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 0
0 = CLCIN0 (inverted) is not gated into CLCx Gate 0

REGISTER 31-16: UxP3H: UART PARAMETER 3 HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	P3<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 7-6 **Unimplemented:** Read as ‘0’

bit 0 **P3<8>:** Most Significant Bit of Parameter 3
DMX mode:
Most Significant bit of last address of receive block
Other modes:
Not used

REGISTER 31-17: UxP3L: UART PARAMETER 3 LOW REGISTER

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
P3<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	

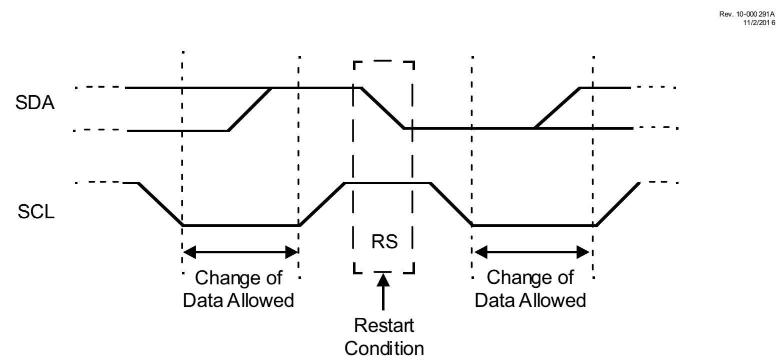
bit 7-0 **P3<7:0>:** Least Significant Bits of Parameter 3
DMX mode:
Least Significant Byte of last address of receive block
LIN Slave mode:
Number of data bytes to receive
Asynchronous Address mode:
Receiver address mask. Received address is XOR’d with UxP2L then AND’d with UxP3L
Match occurs when result is zero
Other modes:
Not used

33.3.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 33-4 shows the waveform for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes (SMA = 1), the master can issue a Restart and the high address byte with the R/W bit set. The slave logic will then hold the clock and prepare to clock out data.

FIGURE 33-4: RESTART CONDITION



33.3.8 ACKNOWLEDGE SEQUENCE

The ninth SCL pulse for any transferred byte in I^2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge (ACK) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an ACK is placed in the ACKSTAT bit of the I2CxCON1 register. The ACKSTAT bit is cleared when the receiving device sends an Acknowledge and is set when the receiving device does not Acknowledge. A slave sends an Acknowledge when it has recognized its address. When in a mode that is receiving data, the ACK data being sent to the transmitter depends on the value of I2CxCNT register. ACKDT is the value sent when I2CxCNT! = 0. When I2CxCNT = 0, the ACKCNT value is used instead.

In Slave mode, if the ADRIE or WRIE bits are set, clock stretching is initiated when there is an address match or when there is an attempt to write to slave. This allows the user to set the ACK value sent back to the transmitter. The ACKDT bit of the I2CxCON1 register is set/cleared to determine the response. Slave hardware will generate an ACK response if the ADRIE or WRIE bits are clear.

Certain conditions will cause a not-ACK (NACK) to be sent automatically. If any of the RXRE, TXRE, RXO, or TXU bits is set, the hardware response is forced to NACK. All subsequent responses from the device for address matches or data will be a NACK response.

33.3.9 BUS TIME-OUT

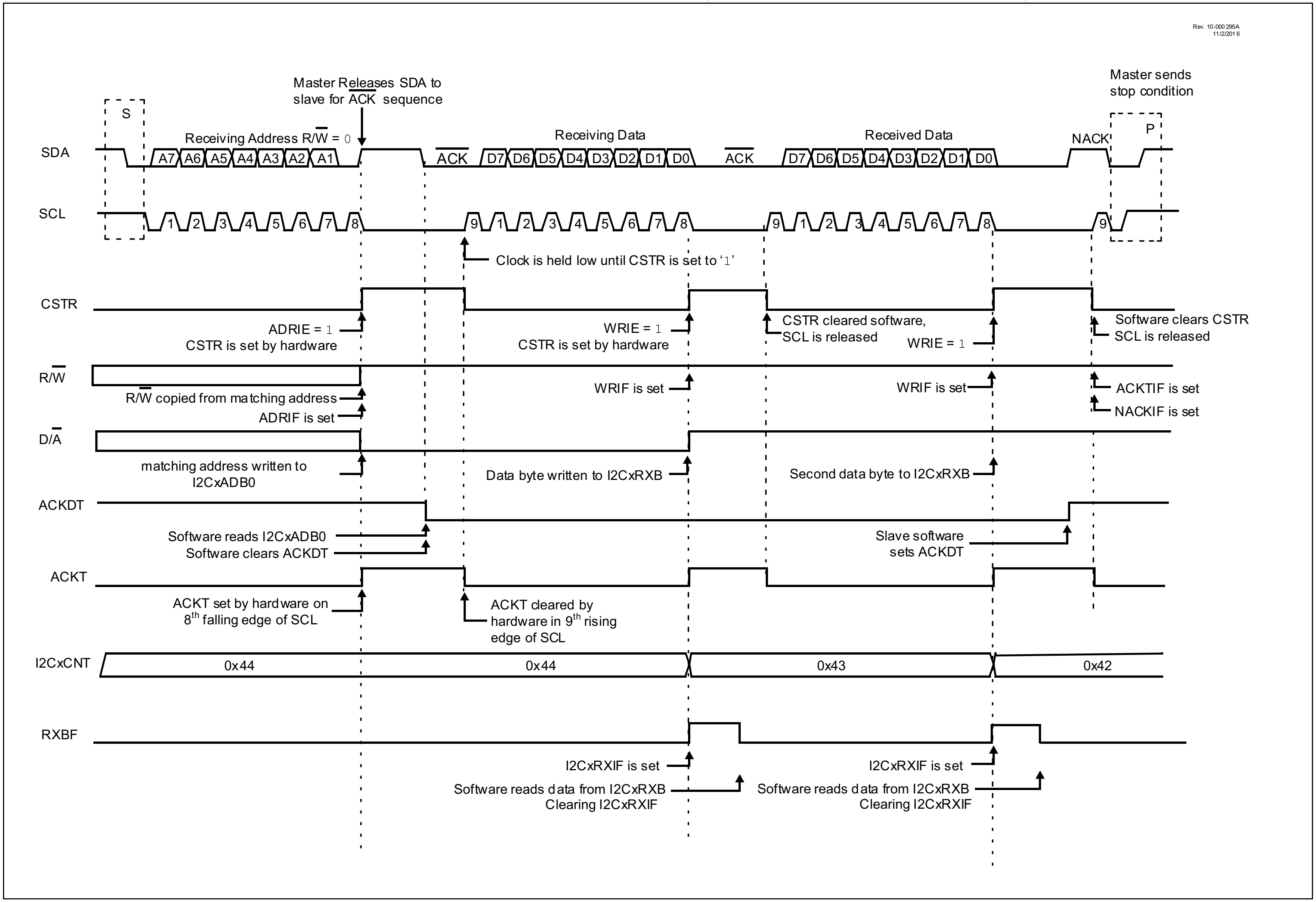
The I2CxBTO register can be used to select the time-out source for the module. The I^2C module is reset when the selected bus time out signal goes high. This feature is useful for SMBus and PMBus™ compatibility.

For example, Timer2 can be selected as the bus time-out source and configured to count when the SCL pin is low. If the timer runs over before the SCL pin transitioned high, the timer-out pulse will reset the module.

Note:	The bus time-out source should produce a rising edge.

If the module is configured as a slave and a BTO event occurs when the slave is active (i.e., the SMA bit is set), the module is immediately reset. The SMA and CSTR bits are also cleared, and the BTOIF bit is set.

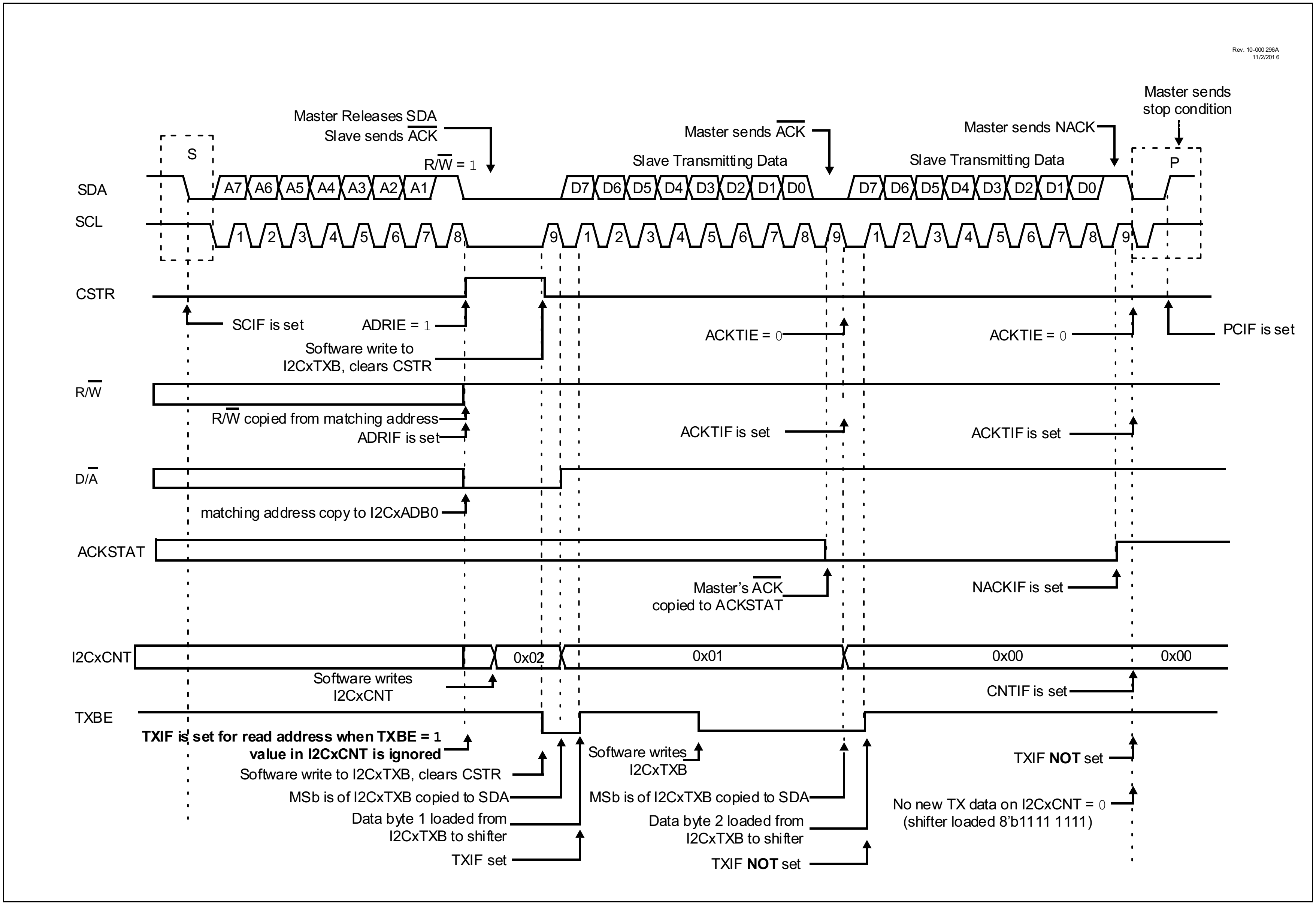
FIGURE 33-8: I²C SLAVE, 7-BIT ADDRESS, RECEPTION NO I2CxCNT (ACKTIE = 0, ADRIE = 1, WRIE = 1)



© 2017 Microchip Technology Inc.

Preliminary

FIGURE 33-9: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION



© 2017 Microchip Technology Inc.

Preliminary

REGISTER 33-3: I2CxCON2: I^{2}C CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACNT	GCEN	FME	ADB	SDAHT<1:0>		BFRET<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	HS = Hardware set HC = Hardware clear

bit 7 **ACNT:** Auto-Load I^{2}C Count Register Enable bit
1 = The first received or transmitted byte after the address, is automatically loaded into the I2CCNT register. The I2CCNT register is loaded at the same time as the value is moved to/from the shifter. ACKDT is used to determine the ACK/NACK value for the address bytes and first data byte of a received message. This prevents a I2CCNT<NACK> from being sent for the byte that would update the I2CCNT register.
0 = Auto-load of I2CCNT disabled

bit 6 **GCEN:** General Call Address Enable bit (MODE<2:0> = 00x & 11x)
1 = General call address, 0x00, causes address match event
0 = General call address disabled

bit 5 **FME:** Fast Mode Enable bit
1 = SCL is sampled high only once before driving SCL low. ($F_{SCL} = F_{CLK}/4$)
0 = SCL is sampled high twice before driving SCL low. ($F_{SCL} = F_{CLK}/5$)

bit 4 **ADB:** Address Data Buffer Disable bit
1 = Received address data is loaded into both the I2CADB and I2CRXB
Transmitted address data is loaded from the I2CTXB
0 = Received address data is loaded only into the I2CADB
Transmitted address data is loaded from the I2CADB0/1 registers.

bit 3-2 **SDAHT<1:0>:** SDA Hold Time Selection bits
11 = Reserved
10 = Minimum of 30 ns hold time on SDA after the falling edge of SCL
01 = Minimum of 100 ns hold time on SDA after the falling edge of SCL
00 = Minimum of 300 ns hold time on SDA after the falling edge of SCL

bit 1-0 **BFRET<1:0>:** Bus Free Time Selection bits
11 = 64 I^{2}C Clock pulses
10 = 32 I^{2}C Clock pulses
01 = 16 I^{2}C Clock pulses
00 = 8 I^{2}C Clock pulses

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

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.

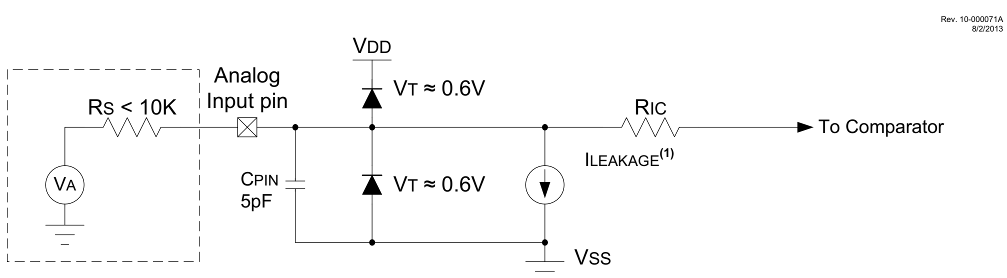
39.8 Analog Input Connection Considerations

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 VDD and VSS. The analog input, therefore, must be between VSS and VDD. 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:
- CPIN = Input Capacitance
- ILEAKAGE = Leakage Current at the pin due to various junctions
- RIC = Interconnect Resistance
- RS = Source Impedance
- VA = Analog Voltage
- VT = Threshold Voltage

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

REGISTER 39-2: CMxCON1: COMPARATOR x CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-2 Unimplemented: Read as ‘0’

bit 1 **INTP**: Comparator Interrupt on Positive-Going Edge Enable bit
1 = The CxIF interrupt flag will be set upon a positive-going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive-going edge of the CxOUT bit

bit 0 **INTN**: Comparator Interrupt on Negative-Going Edge Enable bit
1 = The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative-going edge of the CxOUT bit

REGISTER 39-3: CMxNCH: COMPARATOR x INVERTING CHANNEL SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	NCH<2:0>		
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 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **NCH<2:0>:** Comparator Inverting Input Channel Select bits
111 = VSS
110 = FVR_Buffer2
101 = NCH not connected
100 = NCH not connected
011 = CxIN3-
010 = CxIN2-
001 = CxIN1-
000 = CxIN0-

SUBWF Subtract W from f

Syntax:	SUBWF f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f) – (W) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	`0101` `11da` `ffff` `ffff`
Description:	Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored 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 $f \le 95$ (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 1: `SUBWF REG, 1, 0`

```
Before Instruction
    REG   =   3
    W     =   2
    C     =   ?
After Instruction
    REG   =   1
    W     =   2
    C     =   1     ; result is positive
    Z     =   0
    N     =   0
```

Example 2: `SUBWF REG, 0, 0`

```
Before Instruction
    REG   =   2
    W     =   2
    C     =   ?
After Instruction
    REG   =   2
    W     =   0
    C     =   1     ; result is zero
    Z     =   1
    N     =   0
```

Example 3: `SUBWF REG, 1, 0`

```
Before Instruction
    REG   =   1
    W     =   2
    C     =   ?
After Instruction
    REG   =   FFh   ;(2's complement)
    W     =   2
    C     =   0     ; result is negative
    Z     =   0
    N     =   1
```

SUBWFB Subtract W from f with Borrow

Syntax:	SUBWFB f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f) – (W) – ($\overline{C}$) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	`0101` `10da` `ffff` `ffff`
Description:	Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored 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 $f \le 95$ (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 1: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG   =   19h   (0001 1001)
    W     =   0Dh   (0000 1101)
    C     =   1
After Instruction
    REG   =   0Ch   (0000 1100)
    W     =   0Dh   (0000 1101)
    C     =   1
    Z     =   0
    N     =   0     ; result is positive
```

Example 2: `SUBWFB REG, 0, 0`

```
Before Instruction
    REG   =   1Bh   (0001 1011)
    W     =   1Ah   (0001 1010)
    C     =   0
After Instruction
    REG   =   1Bh   (0001 1011)
    W     =   00h
    C     =   1
    Z     =   1     ; result is zero
    N     =   0
```

Example 3: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG   =   03h   (0000 0011)
    W     =   0Eh   (0000 1110)
    C     =   1
After Instruction
    REG   =   F5h   (1111 0101)
                    ; [2's comp]
    W     =   0Eh   (0000 1110)
    C     =   0
    Z     =   0
    N     =   1     ; result is negative
```

42.2.2 EXTENDED INSTRUCTION SET

ADDULNK	**Add Literal to FSR2 and Return**
Syntax:	ADDULNK k
Operands:	$0 \leq k \leq 63$
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

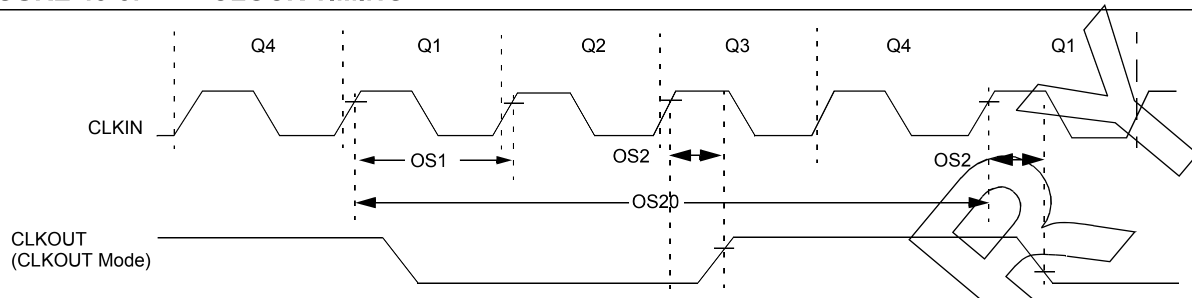
<u>Example:</u> `ADDULNK 23h`

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 45-5: CLOCK TIMING



Note 1: See Table 45-7.

TABLE 45-7: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
ECL Oscillator							
OS1	F_{ECL}	Clock Frequency	—	—	500	kHz	
OS2	T_{ECL_DC}	Clock Duty Cycle	40	—	60	%	
ECM Oscillator							
OS3	F_{ECM}	Clock Frequency	—	—	4	MHz	
OS4	T_{ECM_DC}	Clock Duty Cycle	40	—	60	%	
ECH Oscillator							
OS5	F_{ECH}	Clock Frequency	—	—	32	MHz	
OS6	T_{ECH_DC}	Clock Duty Cycle	40	—	60	%	
LP Oscillator							
OS7	F_{LP}	Clock Frequency	—	—	100	kHz	**Note 4**
XT Oscillator							
OS8	F_{XT}	Clock Frequency	—	—	4	MHz	**Note 4**
HS Oscillator							
OS9	F_{HS}	Clock Frequency	—	—	20	MHz	**Note 4**
Secondary Oscillator							
OS10	F_{SEC}	Clock Frequency	32.4	32.768	33.1	kHz	
System Oscillator							
OS20	F_{OSC}	System Clock Frequency	—	—	64	MHz	**(Note 2, Note 3)**

* 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.

PRELIMINARY