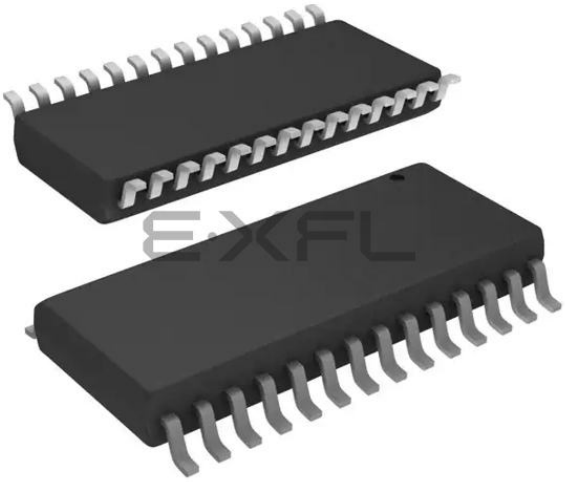


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 10x10b
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/pic18f26j11-i-so

3.2.5.1 OSCTUNE Register

The internal oscillator's output has been calibrated at the factory but can be adjusted in the user's application. This is done by writing to the OSCTUNE register (Register 3-1).

When the OSCTUNE register is modified, the INTOSC frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has completed.

The OSCTUNE register also contains the INTSRC bit. The INTSRC bit allows users to select which internal oscillator provides the clock source when the 31 kHz frequency option is selected. This is covered in more detail in **Section 3.3.1 "Oscillator Control Register"**.

The 4x Phase Locked Loop (PLL) can be used with the internal oscillator block to produce faster device clock speeds than are normally possible with the internal oscillator sources. When enabled, the PLL produces a clock speed up to 32 MHz.

PLL operation is controlled through software. The control bit, PLLEN (OSCTUNE<6>), is used to enable or disable its operation. The PLL is available only to INTOSC when the device is configured to use one of the INTPLL modes as the primary clock source, SCS<1:0> `= 00` (FOSC<2:0> `= 011` or `010`). Additionally, the PLL will only function when the selected output frequency is either 4 MHz or 8 MHz (OSCCON<6:4> = `111` or `110`).

When configured for one of the PLL enabled modes, setting the PLLEN bit does not immediately switch the device clock to the PLL output. The PLL requires up to two milliseconds to start-up and lock, during which time, the device continues to be clocked. Once the PLL output is ready, the microcontroller core will automatically switch to the PLL derived frequency.

3.2.5.2 Internal Oscillator Output Frequency and Drift

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8.0 MHz. However, this frequency may drift as VDD or temperature changes, which can affect the controller operation in a variety of ways.

The low-frequency INTRC oscillator operates independently of the INTOSC source. Any changes in INTOSC across voltage and temperature are not necessarily reflected by changes in INTRC and vice versa.

3.2.5.3 Compensating for INTOSC Drift

It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. This has no effect on the INTRC clock source frequency.

Tuning the INTOSC source requires knowing when to make the adjustment, in which direction it should be made, and in some cases, how large a change is needed. When using the EUSART, for example, an adjustment may be required when it begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high; to adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low; to compensate, increment OSCTUNE to increase the clock frequency.

It is also possible to verify device clock speed against a reference clock. Two timers may be used: one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator. Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

Finally, an ECCP module can use free-running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is greater than the calculated time, the internal oscillator block is running too fast; to compensate, decrement the OSCTUNE register. If the measured time is less than the calculated time, the internal oscillator block is running too slow; to compensate, increment the OSCTUNE register.

11.2.4 BUFFERED PARALLEL SLAVE PORT MODE

Buffered Parallel Slave Port mode is functionally identical to the legacy PSP mode with one exception, the implementation of 4-level read and write buffers. Buffered PSP mode is enabled by setting the INCM bits in the PMMODEH register. If the INCM<1:0> bits are set to '`11`', the PMP module will act as the Buffered PSP.

When the Buffered mode is active, the PMDIN1L, PMDIN1H, PMDIN2L and PMDIN2H registers become the write buffers and the PMDOUT1L, PMDOUT1H, PMDOUT2L and PMDOUT2H registers become the read buffers. Buffers are numbered 0 through 3, starting with the lower byte of PMDIN1L to PMDIN2H as the read buffers and PMDOUT1L to PMDOUT2H as the write buffers.

11.2.4.1 READ FROM SLAVE PORT

For read operations, the bytes will be sent out sequentially, starting with Buffer 0 (PMDOUT1L<7:0>) and ending with Buffer 3 (PMDOUT2H<7:0>) for every read strobe. The module maintains an internal pointer to keep track of which buffer is to be read. Each buffer has a corresponding read status bit, OBxE, in the PMSTATL register. This bit is cleared when a buffer contains data that has not been written to the bus, and is set when data is written to the bus. If the current buffer location being read from is empty, a buffer underflow is generated, and the Buffer Overflow flag bit, OBUF, is set. If all four OBxE status bits are set, then the Output Buffer Empty flag (OBE) will also be set.

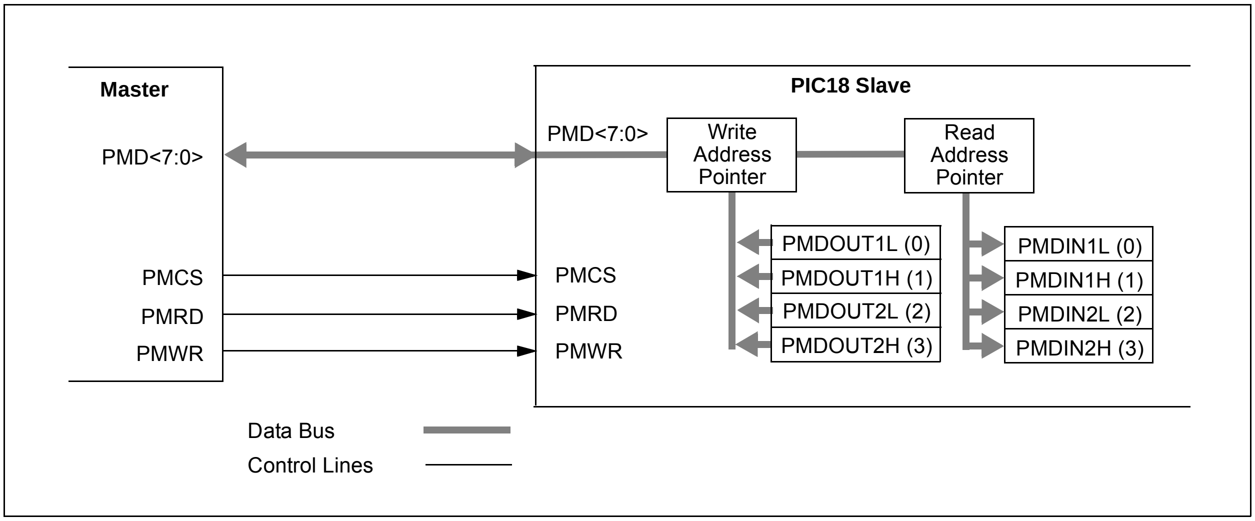
11.2.4.2 WRITE TO SLAVE PORT

For write operations, the data has to be stored sequentially, starting with Buffer 0 (PMDIN1L<7:0>) and ending with Buffer 3 (PMDIN2H<7:0>). As with read operations, the module maintains an internal pointer to the buffer that is to be written next.

The input buffers have their own write status bits, IBxF in the PMSTATH register. The bit is set when the buffer contains unread incoming data, and cleared when the data has been read. The flag bit is set on the write strobe. If a write occurs on a buffer when its associated IBxF bit is set, the Buffer Overflow flag, IBOV, is set; any incoming data in the buffer will be lost. If all four IBxF flags are set, the Input Buffer Full Flag (IBF) is set.

In Buffered Slave mode, the module can be configured to generate an interrupt on every read or write strobe (IRQM<1:0> = `01`). It can be configured to generate an interrupt on a read from Read Buffer 3 or a write to Write Buffer 3, which is essentially an interrupt every fourth read or write strobe (RQM<1:0> = `11`). When interrupting every fourth byte for input data, all input buffer registers should be read to clear the IBxF flags. If these flags are not cleared, then there is a risk of hitting an overflow condition.

FIGURE 11-5: PARALLEL MASTER/SLAVE CONNECTION BUFFERED EXAMPLE



11.3.11 MASTER MODE TIMING

This section contains a number of timing examples that represent the common Master mode configuration options. These options vary from 8-bit to 16-bit data, fully demultiplexed to fully multiplexed address and Wait states.

FIGURE 11-12: READ AND WRITE TIMING, 8-BIT DATA, DEMULTIPLEXED ADDRESS

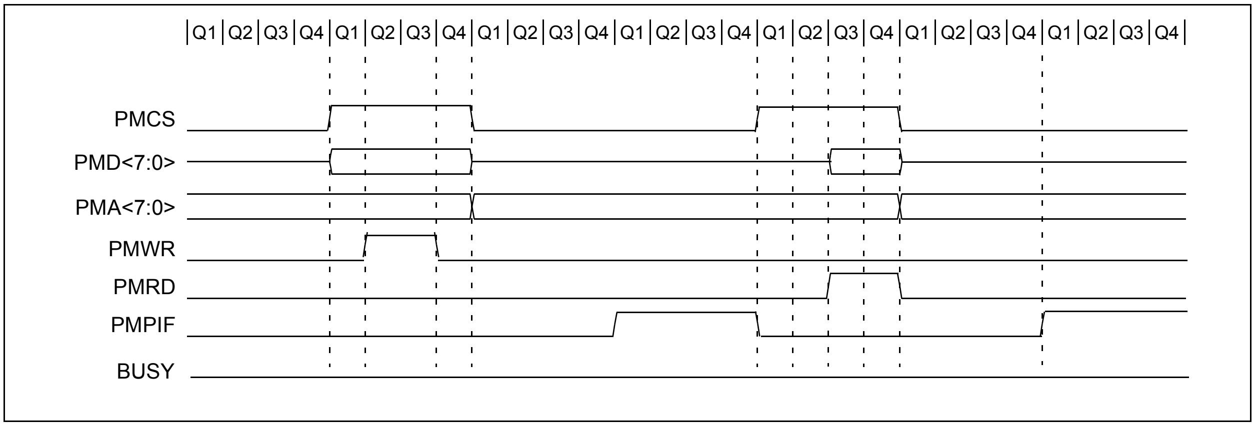


FIGURE 11-13: READ TIMING, 8-BIT DATA, PARTIALLY MULTIPLEXED ADDRESS

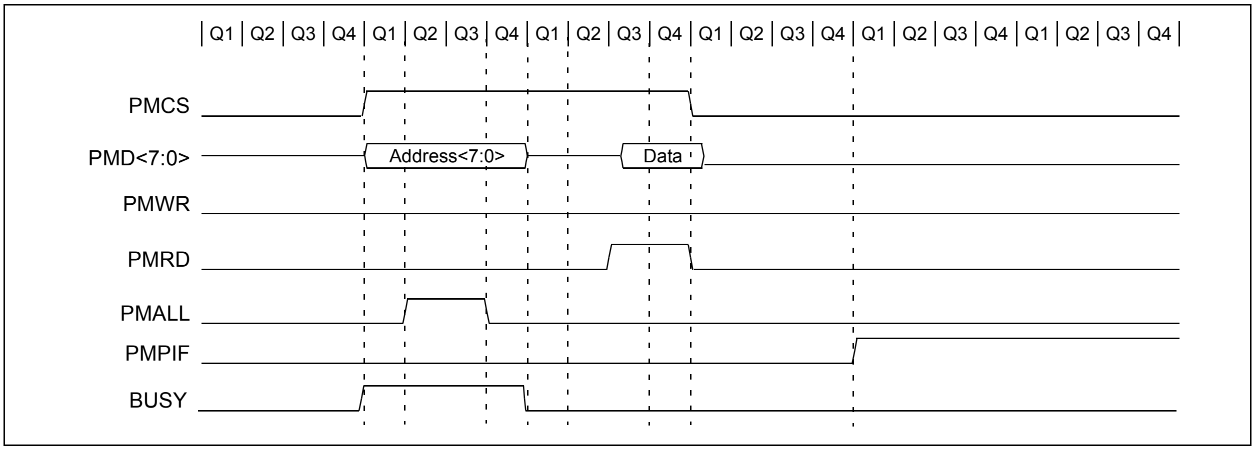
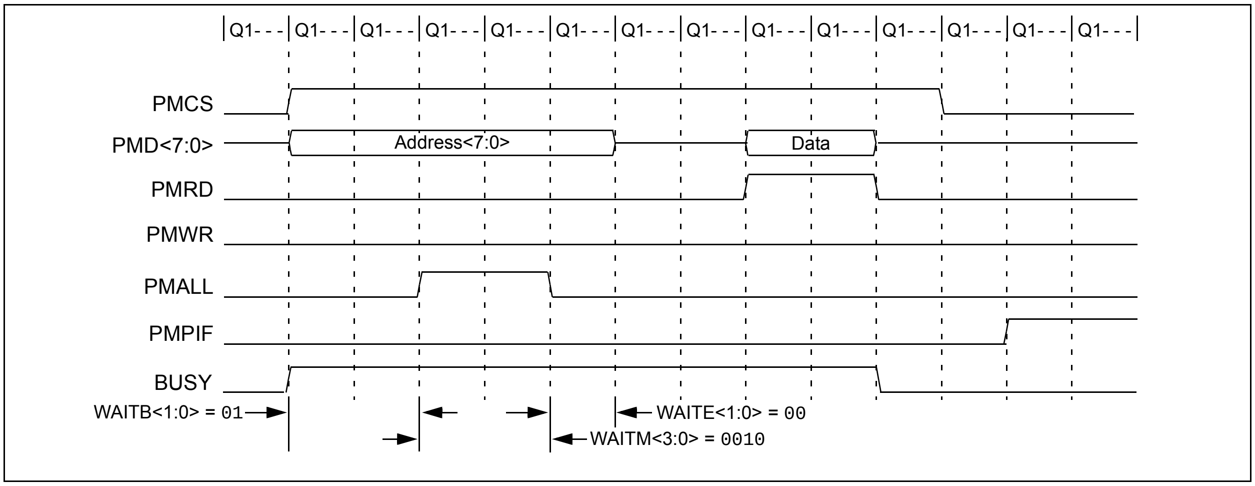


FIGURE 11-14: READ TIMING, 8-BIT DATA, WAIT STATES ENABLED, PARTIALLY MULTIPLEXED ADDRESS



17.1.2 RTCVALH AND RTCVALL REGISTER MAPPINGS

REGISTER 17-6: RESERVED REGISTER (ACCESS F99h, PTR 11b)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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	**Unimplemented:** Read as ‘`0`’

REGISTER 17-7: YEAR: YEAR VALUE REGISTER (ACCESS F98h, PTR 11b)(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
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 7-4	**YRTEN<3:0>:** Binary Coded Decimal Value of Year’s Tens Digit bits Contains a value from 0 to 9.
bit 3-0	**YRONE<3:0>:** Binary Coded Decimal Value of Year’s Ones Digit bits Contains a value from 0 to 9.

Note 1: A write to the YEAR register is only allowed when RTCWREN = `1`.

REGISTER 17-8: MONTH: MONTH VALUE REGISTER (ACCESS F99h, PTR 10b)(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
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 7-5	**Unimplemented:** Read as ‘`0`’
bit 4	**MTHTEN0:** Binary Coded Decimal Value of Month’s Tens Digit bit Contains a value of 0 or 1.
bit 3-0	**MTHONE<3:0>:** Binary Coded Decimal Value of Month’s Ones Digit bits Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = `1`.

19.4.6 USING THE SPI DMA MODULE

The following steps would typically be taken to enable and use the SPI DMA module:

1. Configure the I/O pins, which will be used by MSSP2.
    a) Assign SCK2, SDO2, SDI2 and $\overline{\text{SS2}}$ to RPn pins as appropriate for the SPI mode which will be used. Only functions which will be used need to be assigned to a pin.
    b) Initialize the associated LATx registers for the desired Idle SPI bus state.
    c) If Open-Drain Output mode on SDO2 and SCK2 (Master mode) is desired, set ODCON3<1>.
    d) Configure corresponding TRISx bits for each I/O pin used
2. Configure and enable MSSP2 for the desired SPI operating mode.
    a) Select the desired operating mode (Master or Slave, SPI Mode 0, 1, 2 and 3) and configure the module by writing to the SSP2STAT and SSP2CON1 registers.
    b) Enable MSSP2 by setting SSP2CON1<5> = 1.
3. Configure the SPI DMA engine.
    a) Select the desired operating mode by writing the appropriate values to DMACON2 and DMACON1.
    b) Initialize the TXADDRH/TXADDRL Pointer (Full-Duplex or Half-Duplex Transmit Only mode).
    c) Initialize the RXADDRH/RXADDRL Pointer (Full-Duplex or Half-Duplex Receive Only mode).
    d) Initialize the DMABCH/DMABCL Byte Count register with the number of bytes to be transferred in the next SPI DMA operation.
    e) Set the DMAEN bit (DMACON1<0>).

        In SPI Master modes, this will initiate a DMA transaction. In SPI Slave modes, this will complete the initialization process, and the module will now be ready to begin receiving and/or transmitting data to the master device once the master starts the transaction.
4. Detect the SSP2IF interrupt condition (PIR3<7).
    a) If the interrupt was configured to occur at the completion of the SPI DMA transaction, the DMAEN bit (DMACON1<0>) will be clear. User firmware may prepare the module for another transaction by repeating steps 3.b through 3.e.
    b) If the interrupt was configured to occur prior to the completion of the SPI DMA transaction, the DMAEN bit may still be set, indicating the transaction is still in progress. User firmware would typically use this interrupt condition to begin preparing new data for the next DMA transaction. Firmware should not repeat steps 3.b. through 3.e. until the DMAEN bit is cleared by the hardware, indicating the transaction is complete.

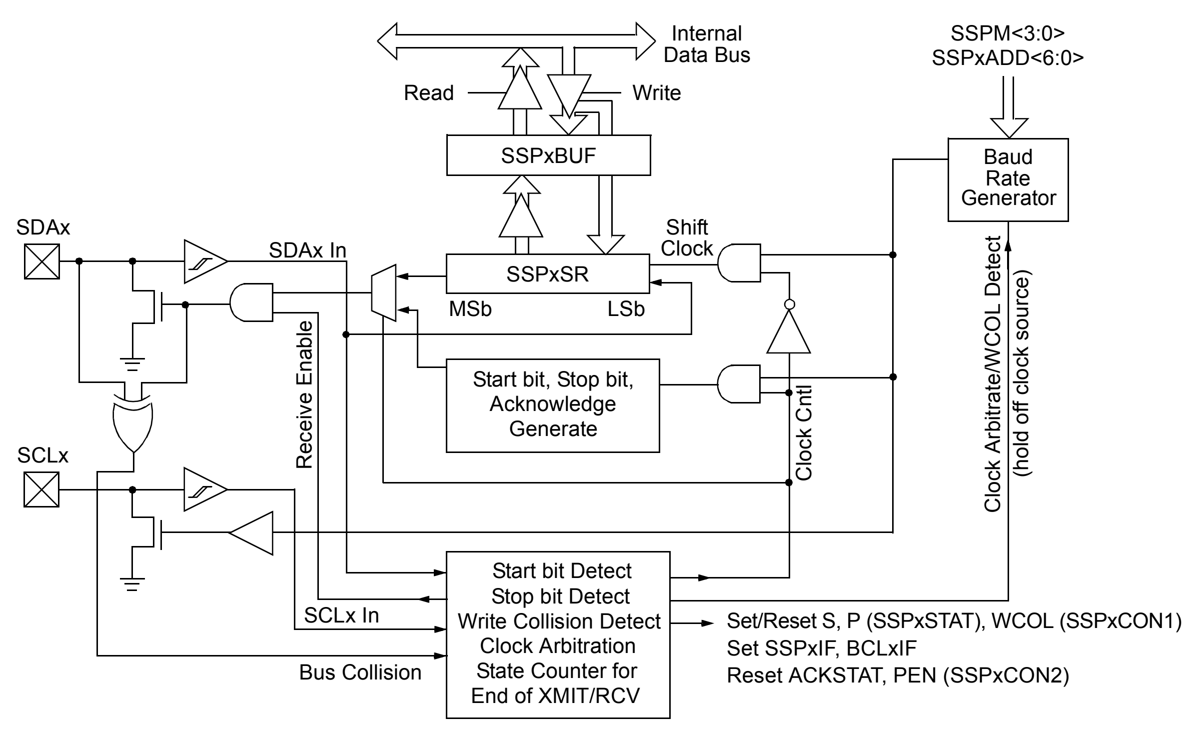
Example 19-2 provides example code demonstrating the initialization process and the steps needed to use the SPI DMA module to perform a 512-byte Full-Duplex, Master mode transfer.

Note: The MSSP module, when configured in I^{2}C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

The following events will cause the MSSP Interrupt Flag bit, SSPxIF, to be set (and MSSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 19-18: MSSPx BLOCK DIAGRAM (I^{2}C™ MASTER MODE)



19.5.6.1 I^{2}C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I^{2}C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/$\overline{\text{Write}}$ ($R/\overline{W}$) bit. In this case, the $R/\overline{W}$ bit will be logic ‘`0`’. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. S and P conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the $R/\overline{W}$ bit. In this case, the $R/\overline{W}$ bit will be logic ‘`1`’. Thus, the first byte transmitted is a 7-bit slave address, followed by a ‘`1`’ to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. S and P conditions indicate the beginning and end of transmission.

The BRG, used for the SPI mode operation, is used to set the SCLx clock frequency for either 100 kHz, 400 kHz or 1 MHz I^{2}C operation. See **Section 19.5.7 “Baud Rate”** for more details.

FIGURE 19-29: BUS COLLISION DURING START CONDITION (SCLx = 0)

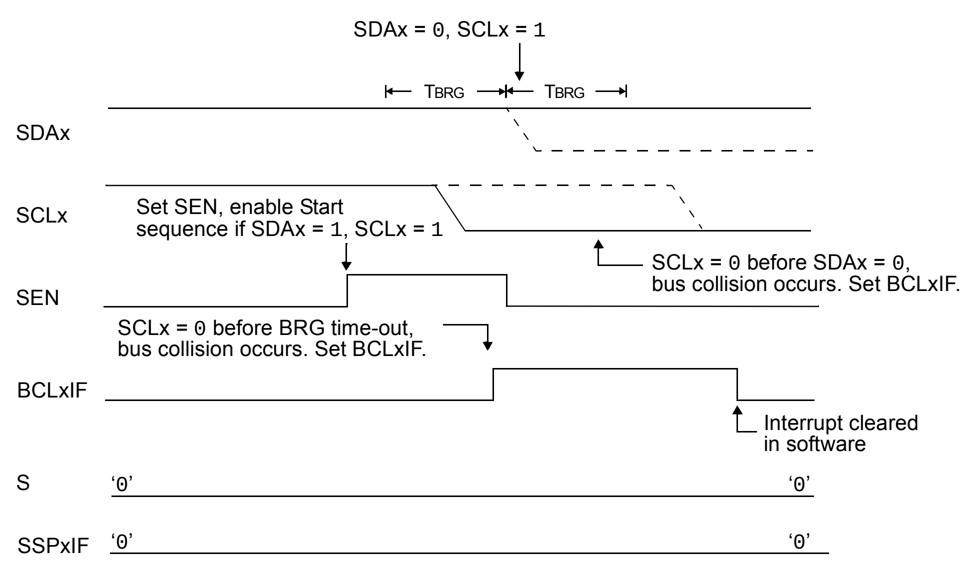
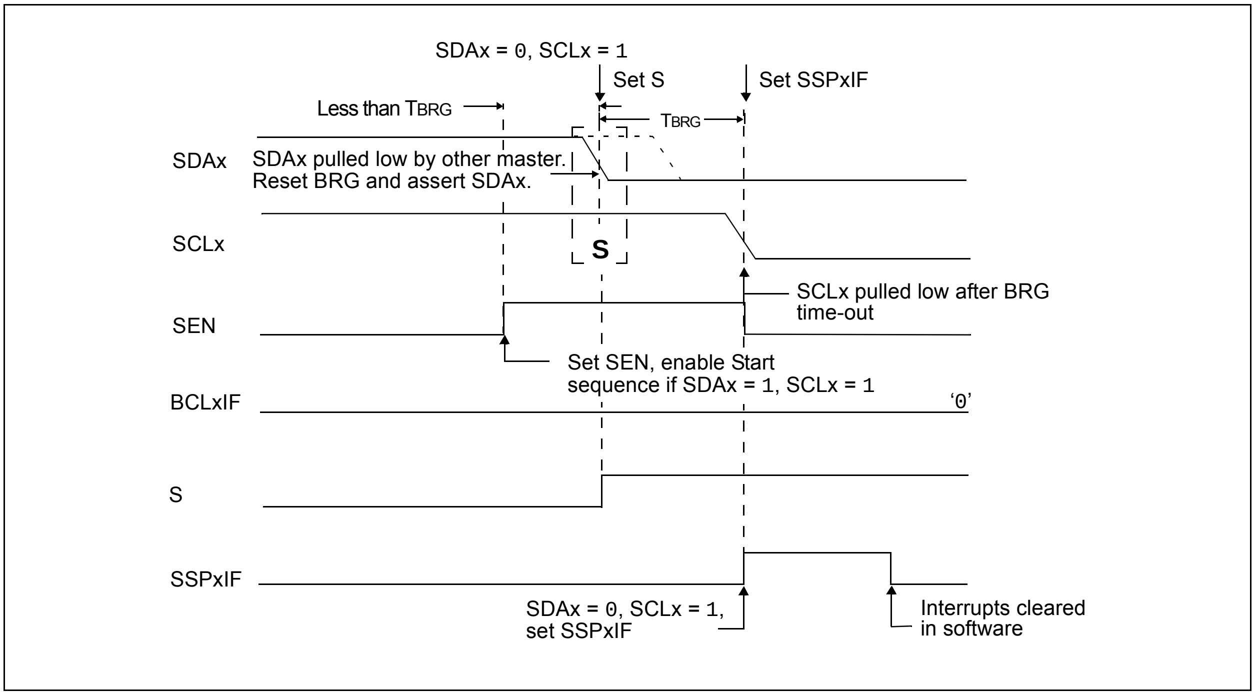
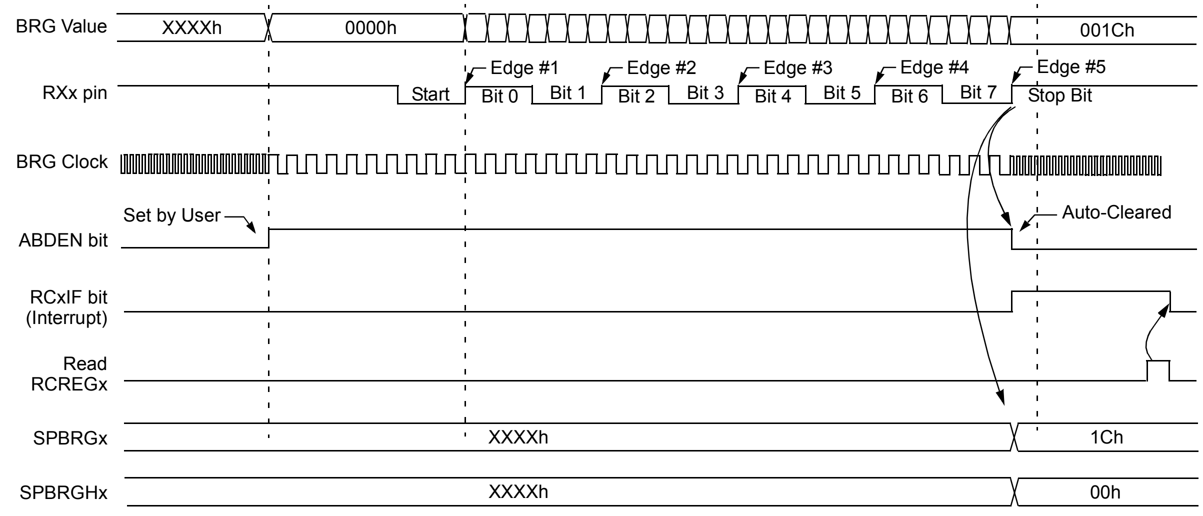


FIGURE 19-30: BRG RESET DUE TO SDAx ARBITRATION DURING START CONDITION



NOTES:

FIGURE 20-1: AUTOMATIC BAUD RATE CALCULATION



Note: The ABD sequence requires the EUSART module to be configured in Asynchronous mode and WUE = 0.

FIGURE 20-2: BRG OVERFLOW SEQUENCE

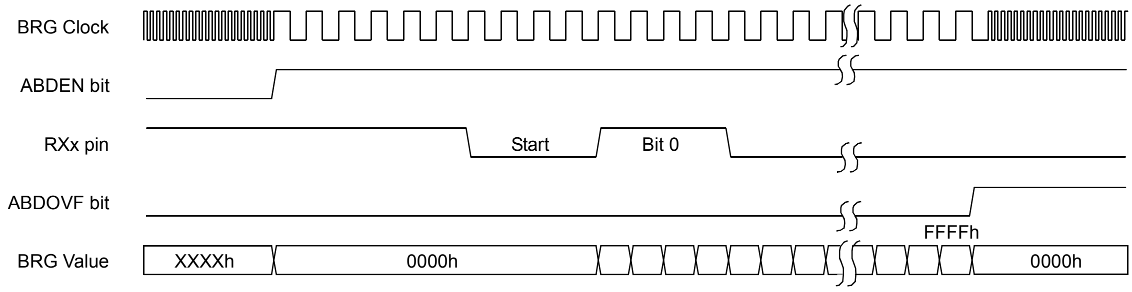


TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF[(1)]	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE[(1)]	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP[(1)]	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	72
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	72
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	72
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
RCREGx	EUSARTx Receive Register								72
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	72
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								73
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								72
ODCON2	—	—	—	—			U2OD	U1OD	74

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous master reception.

Note 1: These pins are only available on 44-pin devices.

21.7 A/D Converter Calibration

The A/D Converter in the PIC18F46J11 family of devices includes a self-calibration feature, which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON1<6>). The next time the GO/$\overline{\text{DONE}}$ bit is set, the module will perform a "dummy" conversion (that is, with reading none of the input channels) and store the resulting value internally to compensate for the offset. Thus, subsequent offsets will be compensated.

Example 21-1 provides an example of a calibration routine.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset or if there are other major changes in operating conditions.

21.8 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT<2:0> and ADCS<2:0> bits in ADCON1 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed. If desired, the device may be placed into the corresponding power-managed Idle mode during the conversion.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

Operation in the Sleep mode requires the A/D RC clock to be selected. If bits, ACQT<2:0>, are set to '`000`' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the `SLEEP` instruction and entry to Sleep mode. The IDLEN and SCS bits in the OSCCON register must have already been cleared prior to starting the conversion.

EXAMPLE 21-1: SAMPLE A/D CALIBRATION ROUTINE

```
        BCF     ANCON0,PCFG0       ;Make Channel 0 analog
        BSF     ADCON0,ADON        ;Enable A/D module
        BSF     ADCON1,ADCAL       ;Enable Calibration
        BSF     ADCON0,GO          ;Start a dummy A/D conversion
CALIBRATION                        ;
        BTFSC   ADCON0,GO          ;Wait for the dummy conversion to finish
        BRA     CALIBRATION        ;
        BCF     ADCON1,ADCAL       ;Calibration done, turn off calibration enable
                                   ;Proceed with the actual A/D conversion
```

SUBWFB	**Subtract W from f with Borrow**
Syntax:	SUBWFB f {,d {,a}}
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [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 (default). If 'a' is '`0`' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 27.2.3 "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 1011)`
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 1101)`
C = 1
After Instruction
REG = F5h `(1111 0100)` ; [2's comp]
W = 0Eh `(0000 1101)`
C = 0
Z = 0
N = 1 ; result is negative

SWAPF	**Swap f**
Syntax:	SWAPF f {,d {,a}}
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	(f<3:0>) → dest<7:4>, (f<7:4>) → dest<3:0>
Status Affected:	None
Encoding:	`0011` `10da` `ffff` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '`0`', the result is placed in W. If 'd' is '`1`', the result is placed 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 (default). If 'a' is '`0`' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 27.2.3 "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: `SWAPF REG, 1, 0`

Before Instruction
REG = 53h
After Instruction
REG = 35h

27.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, the PIC18F46J11 family of devices also provides an optional extension to the core CPU functionality. The added features include eight additional instructions that augment Indirect and Indexed Addressing operations and the implementation of Indexed Literal Offset Addressing for many of the standard PIC18 instructions.

The additional features of the extended instruction set are enabled by default on unprogrammed devices. Users must properly set or clear the XINST Configuration bit during programming to enable or disable these features.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers (FSR), or use them for Indexed Addressing. Two of the instructions, `ADDFSR` and `SUBFSR`, each have an additional special instantiation for using FSR2. These versions (`ADDULNK` and `SUBULNK`) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- Dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- Function Pointer invocation
- Software Stack Pointer manipulation
- Manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 27-3. Detailed descriptions are provided in **Section 27.2.2 “Extended Instruction Set”**. The opcode field descriptions in Table 27-1 (page 414) apply to both the standard and extended PIC18 instruction sets.

Note: The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

27.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the FSRs and some offset to specify a source or destination register. When an argument for an instruction serves as part of Indexed Addressing, it is enclosed in square brackets (“[ ]”). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 27.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**.

Note: In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces (“{ }”).

TABLE 27-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word	Status Affected
				MSb LSb	
ADDFSR	f, k	Add Literal to FSR	1	1110 1000 ffkk kkkk	None
ADDULNK	k	Add Literal to FSR2 and Return	2	1110 1000 11kk kkkk	None
CALLW		Call Subroutine using WREG	2	0000 0000 0001 0100	None
MOVSF	z_s, f_d	Move z_s (source) to 1st word	2	1110 1011 0zzz zzzz	None
		f_d (destination) 2nd word		1111 ffff ffff ffff	—
MOVSS	z_s, z_d	Move z_s (source) to 1st word	2	1110 1011 1zzz zzzz	None
		z_d (destination) 2nd word		1111 xxxx xzzz zzzz	—
PUSHL	k	Store Literal at FSR2, Decrement FSR2	1	1110 1010 kkkk kkkk	None —
SUBFSR	f, k	Subtract Literal from FSR	1	1110 1001 ffkk kkkk	None
SUBULNK	k	Subtract Literal from FSR2 and Return	2	1110 1001 11kk kkkk	None

SUBFSR — Subtract Literal from FSR

Syntax:	SUBFSR f, k
Operands:	0 ≤ k ≤ 63 f ∈ [0, 1, 2]
Operation:	FSRf – k → FSRf
Status Affected:	None
Encoding:	`1110` `1001` `ffkk` `kkkk`
Description:	The 6-bit literal ‘k’ is subtracted from the contents of the FSR specified by ‘f’.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: `SUBFSR 2, 0x23`

Before Instruction
FSR2 = 03FFh

After Instruction
FSR2 = 03DCh

SUBULNK — Subtract Literal from FSR2 and Return

Syntax:	SUBULNK k
Operands:	0 ≤ k ≤ 63
Operation:	FSR2 – k → FSR2, (TOS) → PC
Status Affected:	None
Encoding:	`1110` `1001` `11kk` `kkkk`
Description:	The 6-bit literal ‘k’ is subtracted from the contents of the 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 `SUBFSR` instruction, where f = 3 (binary ‘`11`’); it operates only on FSR2.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: `SUBULNK 0x23`

Before Instruction
FSR2 = 03FFh
PC = 0100h

After Instruction
FSR2 = 03DCh
PC = (TOS)

TABLE 29-1: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Program Flash Memory					
D130	EP	Cell Endurance	10K	—	—	E/W	-40°C to +85°C
D131	VPR	VDDcore for Read	VMIN	—	2.75	V	VMIN = Minimum operating voltage
D132B	VPEW	VDDCORE for Self-Timed Erase or Write	2.25	—	2.75	V	
D133A	TIW	Self-Timed Write Cycle Time	—	2.8	—	ms	64 bytes
D133B	TIE	Self-Timed Block Erase Cycle Time	—	33.0	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	3	—	mA	

† Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

TABLE 29-2: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 3.6V, -40°C < TA < +85°C (unless otherwise stated)

Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D300	VIOFF	Input Offset Voltage	—	±5	±25	mV	
D301	VICM	Input Common Mode Voltage	0	—	VDD	V	
	VIRV	Internal Reference Voltage	0.57	0.60	0.63	V	
D302	CMRR	Common Mode Rejection Ratio	55	—	—	dB	
D303	TRESP	Response Time(1)	—	150	400	ns	
D304	TMC2OV	Comparator Mode Change to Output Valid	—	—	10	µs	

Note 1: Response time measured with one comparator input at VDD/2, while the other input transitions from VSS to VDD.

TABLE 29-3: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
	IOUT1	CTMU Current Source, Base Range	—	550	—	nA	CTMUICON<1:0> = `01`
	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	µA	CTMUICON<1:0> = `10`
	IOUT3	CTMU Current Source, 100x Range	—	55	—	µA	CTMUICON<1:0> = `11`

Note 1: Nominal value at center point of current trim range (CTMUICON<7:2> = `000000`).

TABLE 29-4: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 3.6V, -40°C < TA < +85°C (unless otherwise stated)

Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D310	VRES	Resolution	VDD/24	—	VDD/32	LSb	
D311	VRAA	Absolute Accuracy	—	—	1/2	LSb	
D312	VRUR	Unit Resistor Value (R)	—	2k	—	Ω	
310	TSET	Settling Time(1)	—	—	10	µs	

Note 1: Settling time measured while CVRR = `1` and CVR<3:0> bits transition from ‘`0000`’ to ‘`1111`’.

TABLE 29-5: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < T_A < +85°C (unless otherwise stated)

Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
	V_{RGOUT}	Regulator Output Voltage	2.35	2.5	2.7	V	Regulator enabled, V_{DD} = 3.0V
	C_{EFC}	External Filter Capacitor Value(1)	5.4	10	18	μF	ESR < 3Ω recommended ESR < 5Ω required

Note 1: C_{EFC} applies for PIC18F devices in the family. For PIC18LF devices in the family, there is no specific minimum or maximum capacitance for V_{DDCORE}, although proper supply rail bypassing should still be used.

TABLE 29-6: ULPWU SPECIFICATIONS

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ T_A ≤ +85°C for industrial

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D100	I_{ULP}	**Ultra Low-Power Wake-up Current**	—	60	—	nA	Net of I/O leakage and current sink at 1.6V on pin, V_{DD} = 3.3V See Application Note *AN879,* "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879)

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

29.4 AC (Timing) Characteristics

29.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS		3. TCC:ST	(I^{2}C specifications only)
2. TppS		4. Ts	(I^{2}C specifications only)

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKO	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T13CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance
I^{2}C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^{2}C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	Stop condition
STA	Start condition		

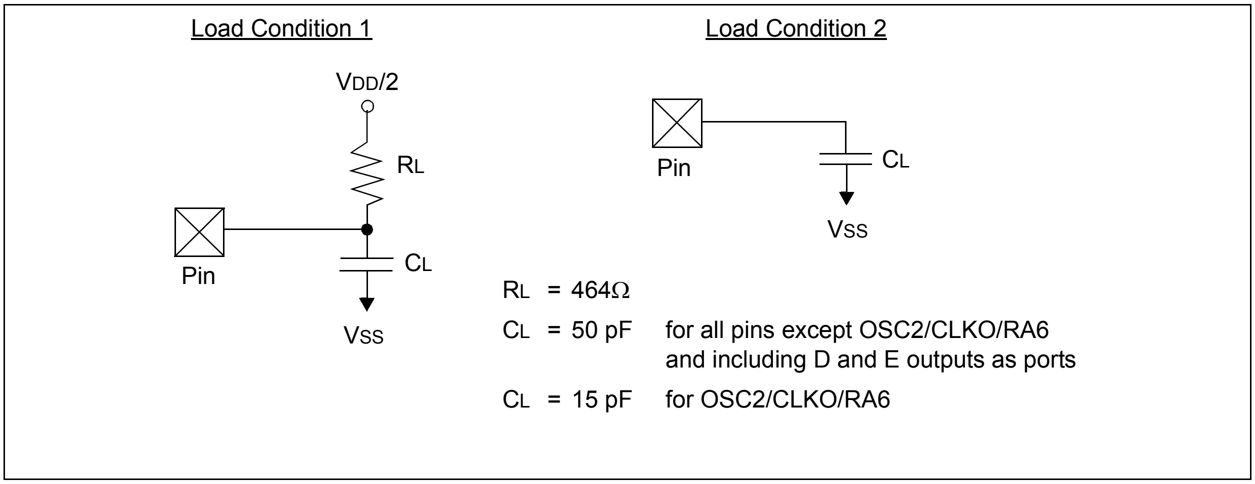
29.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 29-8 apply to all timing specifications unless otherwise noted. Figure 29-4 specifies the load conditions for the timing specifications.

TABLE 29-8: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial Operating voltage VDD range as described in **Section 29.1** and **Section 29.3**.

FIGURE 29-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



RL = 464Ω

CL = 50 pF for all pins except OSC2/CLKO/RA6 and including D and E outputs as ports

CL = 15 pF for OSC2/CLKO/RA6

29.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 29-5: EXTERNAL CLOCK TIMING

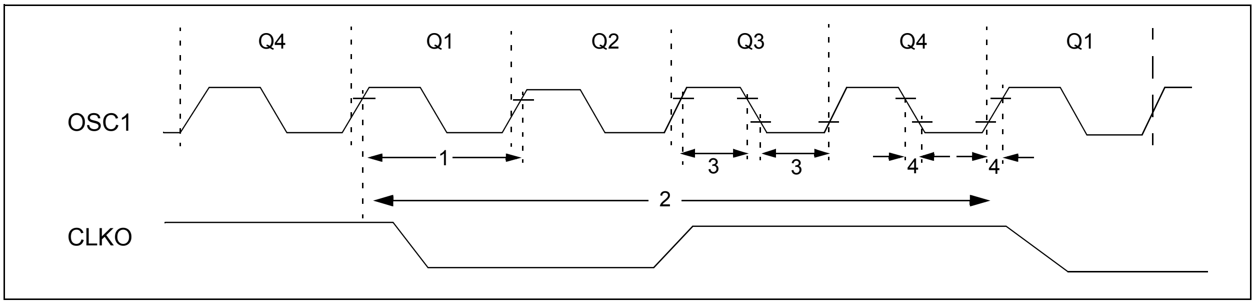
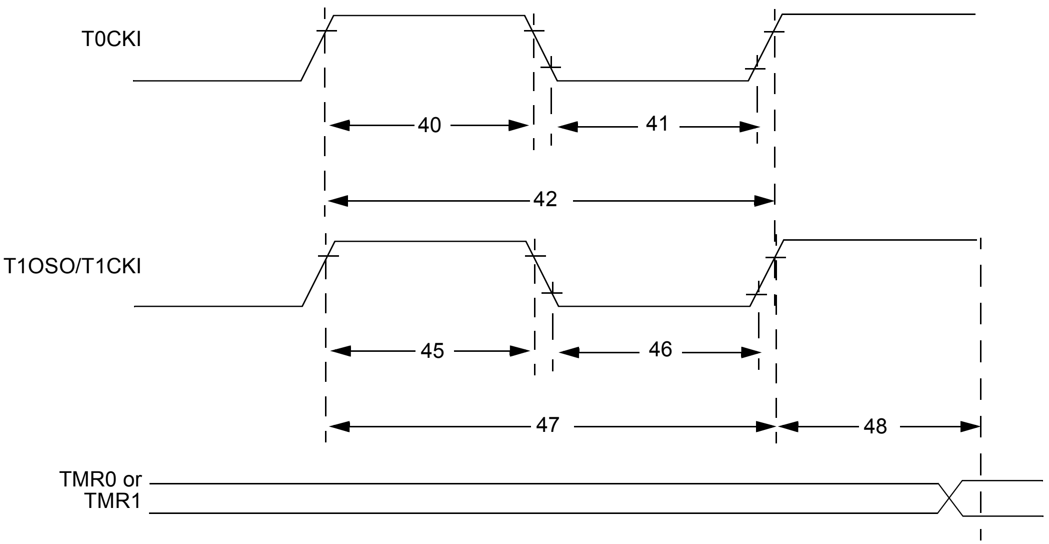


FIGURE 29-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



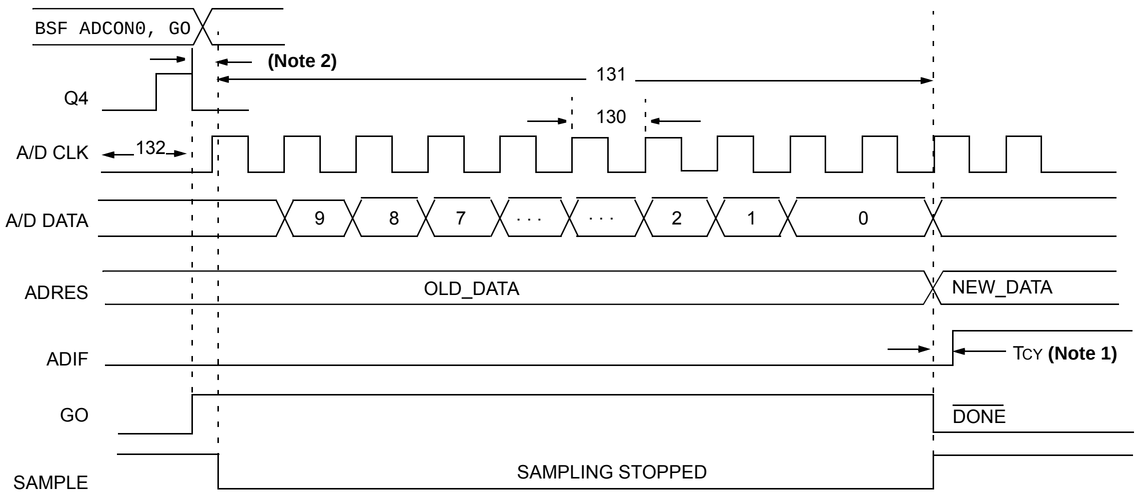
Note: Refer to Figure 29-4 for load conditions.

TABLE 29-15: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
40	TT0H	T0CKI High Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
41	TT0L	T0CKI Low Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
42	TT0P	T0CKI Period	No prescaler	TCY + 10	—	ns	
			With prescaler	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4,..., 256)
45	TT1H	T1CKI/T3CKI High Time	Synchronous, no prescaler	0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
46	TT1L	T1CKI/T3CKI Low Time	Synchronous, no prescaler	0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
47	TT1P	T1CKI/T3CKI Input Period	Synchronous	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	83	—	ns	
	FT1	T1CKI Input Frequency Range(1)		DC	12	MHz	
48	TCKE2TMRI	Delay from External T1CKI Clock Edge to Timer Increment		2 TOSC	7 TOSC	—	

Note 1: The Timer1 oscillator is designed to drive 32.768 kHz crystals. When T1CKI is used as a digital input, frequencies up to 12 MHz are supported.

FIGURE 29-23: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

2: This is a minimal RC delay (typically 100 ns), which also disconnects the holding capacitor from the analog input.

TABLE 29-31: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
130	TAD	A/D Clock Period	0.7	25.0(1)	μs	TOSC based, VREF ≥ 3.0V
131	TCNV	Conversion Time (not including acquisition time)(2)	11 —	12 1	TAD μs	A/D RC Mode
132	TACQ	Acquisition Time(3)	1.4	—	μs	-40°C to +85°C
135	TSWC	Switching Time from Convert → Sample	—	**(Note 4)**		
137	TDIS	Discharge Time	0.2	—	μs	

Note 1: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.

2: ADRES registers may be read on the following TCY cycle.

3: The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD). The source impedance (RS) on the input channels is 50Ω.

4: On the following cycle of the device clock.