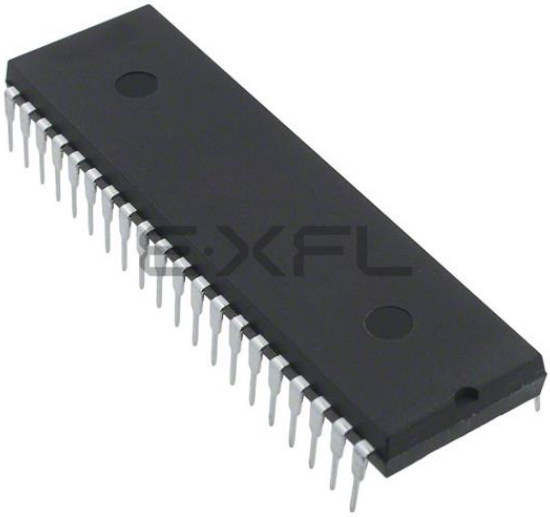


**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 368 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4V ~ 5.5V |
| Data Converters | A/D 14x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 40-DIP (0.600", 15.24mm) |
| Supplier Device Package | 40-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f777-e-p |

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION (CONTINUED)**

| Pin Name | PDIP Pin # | QFN Pin # | TQFP Pin # | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. |
| RB0/INT/AN12 | 33 | 9 | 8 | | TTL/ST(1) | |
| RB0 | | | | I/O | | Digital I/O. |
| INT | | | | I | | External interrupt. |
| AN12 | | | | I | | Analog input channel 12. |
| RB1/AN10 | 34 | 10 | 9 | | TTL | |
| RB1 | | | | I/O | | Digital I/O. |
| AN10 | | | | I | | Analog input channel 10. |
| RB2/AN8 | 35 | 11 | 10 | | TTL | |
| RB2 | | | | I/O | | Digital I/O. |
| AN8 | | | | I | | Analog input channel 8. |
| RB3/CCP2/AN9 | 36 | 12 | 11 | | TTL | |
| RB3 | | | | I/O | | Digital I/O. |
| CCP2(5) | | | | I/O | | CCP2 capture input, compare output, PWM output. |
| AN9 | | | | I | | Analog input channel 9. |
| RB4/AN11 | 37 | 14 | 14 | | TTL | |
| RB4 | | | | I/O | | Digital I/O. |
| AN11 | | | | I | | Analog input channel 11 |
| RB5/AN13/CCP3 | 38 | 15 | 15 | | TTL | |
| RB5 | | | | I/O | | Digital I/O. |
| AN13 | | | | I | | Analog input channel 13. |
| CCP3 | | | | I | | CCP3 capture input, compare output, PWM output. |
| RB6/PGC | 39 | 16 | 16 | | TTL/ST(2) | |
| RB6 | | | | I/O | | Digital I/O. |
| PGC | | | | I/O | | In-Circuit Debugger and ICSP™ programming clock. |
| RB7/PGD | 40 | 17 | 17 | | TTL/ST(2) | |
| RB7 | | | | I/O | | Digital I/O. |
| PGD | | | | I/O | | In-Circuit Debugger and ICSP programming data. |

**Legend:** I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as an external interrupt.
**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
**3:** This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
**4:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
**5:** Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Details on page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 30, 180 |
| 81h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 22, 180 |
| 82h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 29, 180 |
| 83h(4) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 21, 180 |
| 84h(4) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 30, 180 |
| 85h | TRISA | PORTA Data Direction Register | | | | | | | | 1111 1111 | 55, 181 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 64, 181 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 66, 181 |
| 88h(5) | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 67, 181 |
| 89h(5) | TRISE | IBF(5) | OBF(5) | IBOV(5) | PSPMODE(5) | —(8) | PORTE Data Direction bits | | | 0000 1111 | 69, 181 |
| 8Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | 23, 180 |
| 8Bh(4) | INTCON | GIE | PEIE | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 25, 180 |
| 8Ch | PIE1 | PSPIE(3) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 24, 181 |
| 8Dh | PIE2 | OSFIE | CMIE | LVDIE | — | BCLIE | — | CCP3IE | CCP2IE | 000- 0-00 | 26, 181 |
| 8Eh | PCON | — | — | — | — | — | SBOREN | $\overline{POR}$ | $\overline{BOR}$ | ---- -1qq | 28, 181 |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS(7) | IOFS | SCS1 | SCS0 | -000 1000 | 38, 181 |
| 90h | OSCTUNE | — | — | TUN5 | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | --00 0000 | 36, 181 |
| 91h | SSPCON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 105 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 86, 181 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$™ mode) Address Register | | | | | | | | 0000 0000 | 101, 181 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 101, 181 |
| 95h | CCPR3L | Capture/Compare/PWM Register 3 (LSB) | | | | | | | | xxxx xxxx | 92 |
| 96h | CCPR3H | Capture/Compare/PWM Register 3 (MSB) | | | | | | | | xxxx xxxx | 92 |
| 97h | CCP3CON | — | — | CCP3X | CCP3Y | CCP3M3 | CCP3M2 | CCP3M1 | CCP3M0 | --00 0000 | 92 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 145, 181 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 145, 181 |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | ADCON2 | — | — | ACQT2 | ACQT1 | ACQT0 | — | — | — | --00 0--- | 154 |
| 9Ch | CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0111 | 55, 161 |
| 9Dh | CVRCON | CVREN | CVROE | CVRR | — | CVR3 | CVR2 | CVR1 | CVR0 | 000- 0000 | 55, 167 |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | 180 |
| 9Fh | ADCON1 | ADFM | ADCS2 | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 0000 0000 | 153, 181 |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, — = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> bits, whose contents are transferred to the upper byte of the program counter during branches (CALL or GOTO).

**2:** Other (non Power-up) Resets include external Reset through $\overline{MCLR}$ and Watchdog Timer Reset.

**3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**4:** These registers can be addressed from any bank.

**5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices (except for RE3), read as '0'.

**6:** This bit always reads as a '1'.

**7:** OSCCON<OSTS> bit resets to '0' with dual-speed start-up and LP, HS or HS-PLL selected as the oscillator.

**8:** RE3 is an input only. The state of the TRISE3 bit has no effect and will always read '1'.

### 2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

| **Note:** | Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt. |
|---|---|

**REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS 8Ch)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

bit 7 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit(1)
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

**Note 1:** PSPIE is reserved on 28-pin devices; always maintain this bit clear.

bit 6 **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5 **RCIE**: AUSART Receive Interrupt Enable bit
1 = Enables the AUSART receive interrupt
0 = Disables the AUSART receive interrupt

bit 4 **TXIE**: AUSART Transmit Interrupt Enable bit
1 = Enables the AUSART transmit interrupt
0 = Disables the AUSART transmit interrupt

bit 3 **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2 **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

| **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 |

### 4.7.2 SEC_RUN MODE

The core and peripherals can be configured to be clocked by T1OSC using a 32.768 kHz crystal. The crystal must be connected to the T1OSO and T1OSI pins. This is the same configuration as the low-power timer circuit (see **Section 7.6 "Timer1 Oscillator"**). When SCS bits are configured to run from T1OSC, a clock transition is generated. It will clear the OSTS bit, switch the system clock from either the primary system clock or INTRC, depending on the value of SCS<1:0> and FOSC<2:0>, to the external low-power Timer1 oscillator input (T1OSC) and shut-down the primary system clock to conserve power.

After a clock switch has been executed, the internal Q clocks are held in the Q1 state until eight falling edge clocks are counted on the T1OSC. After the eight clock periods have transpired, the clock input to the Q clocks is released and operation resumes (see Figure 4-8). In addition, T1RUN (in T1CON) is set to indicate that T1OSC is being used as the system clock.

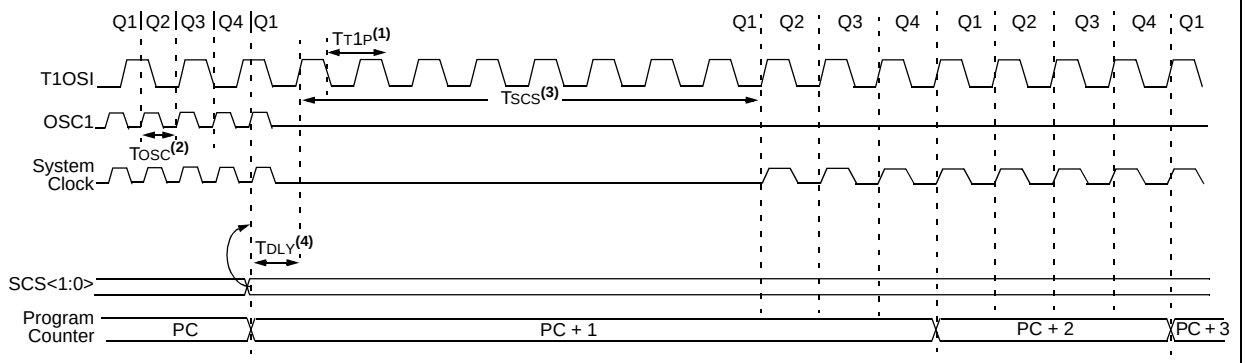
**Note 1:** The T1OSCEN bit must be enabled and it is the user's responsibility to ensure T1OSC is stable before clock switching to the T1OSC input clock can occur.

**2:** When T1OSCEN = 0, the following possible effects result.

| Original SCS<1:0> | Modified SCS<1:0> | Final SCS<1:0> |
|---|---|---|
| 00 | 01 | 00 – no change |
| 00 | 11 | 10 – INTRC |
| 10 | 11 | 10 – no change |
| 10 | 01 | 00 – Oscillator defined by FOSC<2:0> |

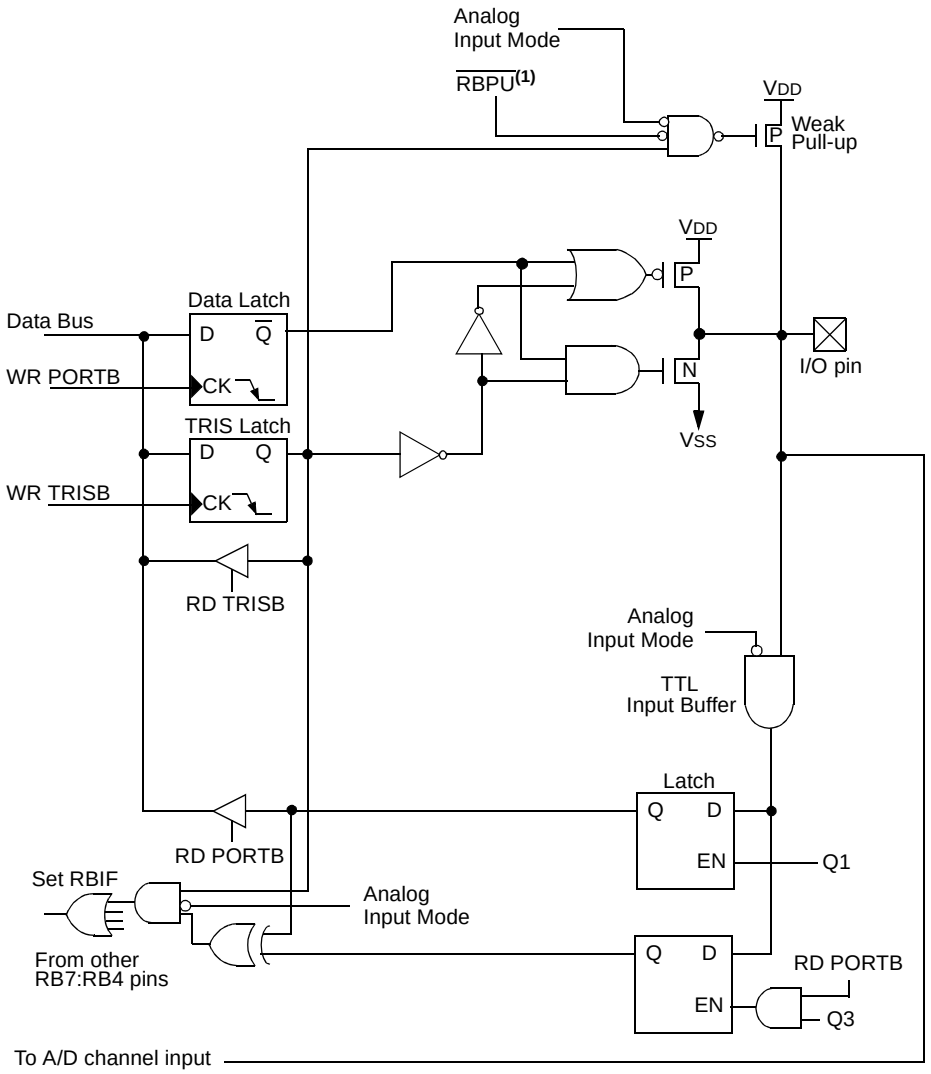
A clock switching event will occur if the final state of the SCS bits is different from the original.

**FIGURE 4-8: TIMING DIAGRAM FOR SWITCHING TO SEC_RUN MODE**




**Note 1:** TT1P = 30.52 μs.
**2:** TOSC = 50 ns minimum.
**3:** TSCS = 8 TT1P
**4:** TDLY = 1 TT1P.

**FIGURE 5-12: BLOCK DIAGRAM OF RB4/AN11 PIN**




**Note 1:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit.

**REGISTER 7-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

| U-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **T1RUN**: Timer1 System Clock Status bit
1 = System clock is derived from Timer1 oscillator
0 = System clock is derived from another source

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

| **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 |

## 7.9 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR, or any other Reset, except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

## 7.10 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

## 7.11 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 7.6 "Timer1 Oscillator"**) gives users the option to include RTC functionality in their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, `RTCisr`, shown in Example 7-3, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it. The simplest method is to set the MSb of TMR1H with a `BSF` instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = `1`) as shown in the routine, `RTCinit`. The Timer1 oscillator must also be enabled and running at all times.

**EXAMPLE 7-3: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE**

```
RTCinit     BANKSEL     TMR1H
            MOVLW       0x80                ; Preload TMR1 register pair
            MOVWF       TMR1H               ; for 1 second overflow
            CLRF        TMR1L
            MOVLW       b'00001111'         ; Configure for external clock,
            MOVWF       T1CON               ; Asynchronous operation, external oscillator
            CLRF        secs                ; Initialize timekeeping registers
            CLRF        mins
            MOVLW       .12
            MOVWF       hours
            BANKSEL     PIE1
            BSF         PIE1, TMR1IE        ; Enable Timer1 interrupt
            RETURN
RTCisr      BANKSEL     TMR1H
            BSF         TMR1H, 7            ; Preload for 1 sec overflow
            BCF         PIR1, TMR1IF        ; Clear interrupt flag
            INCF        secs, F             ; Increment seconds
            MOVF        secs, w
            SUBLW       .60
            BTFSS       STATUS, Z           ; 60 seconds elapsed?
            RETURN                          ; No, done
            CLRF        seconds             ; Clear seconds
            INCF        mins, f             ; Increment minutes
            MOVF        mins, w
            SUBLW       .60
            BTFSS       STATUS, Z           ; 60 seconds elapsed?
            RETURN                          ; No, done
            CLRF        mins                ; Clear minutes
            INCF        hours, f            ; Increment hours
            MOVF        hours, w
            SUBLW       .24
            BTFSS       STATUS, Z           ; 24 hours elapsed?
            RETURN                          ; No, done
            CLRF        hours               ; Clear hours
            RETURN                          ; Done
```

**FIGURE 10-13: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)**

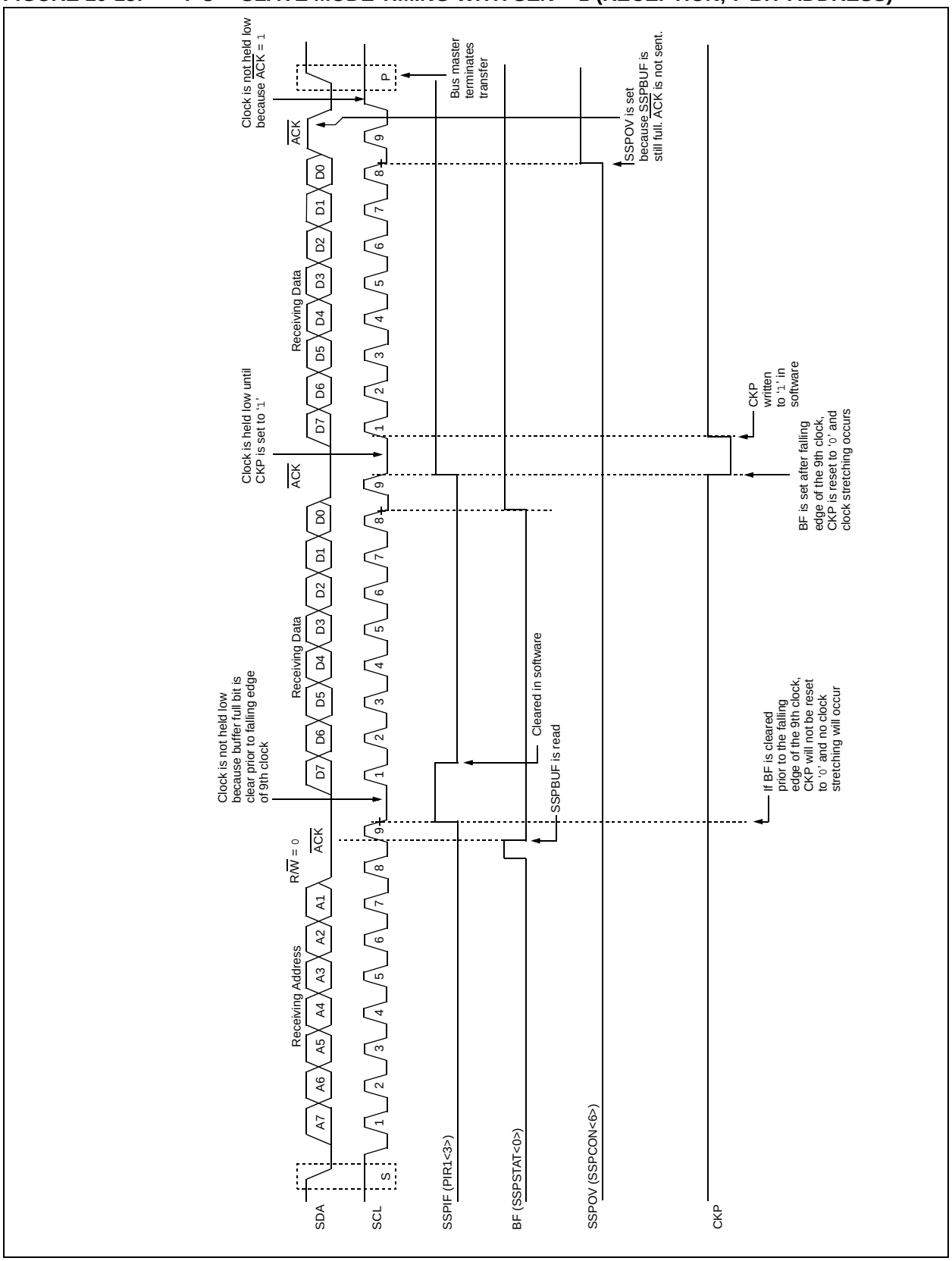


**FIGURE 10-27: BUS COLLISION DURING START CONDITION (SCL = 0)**

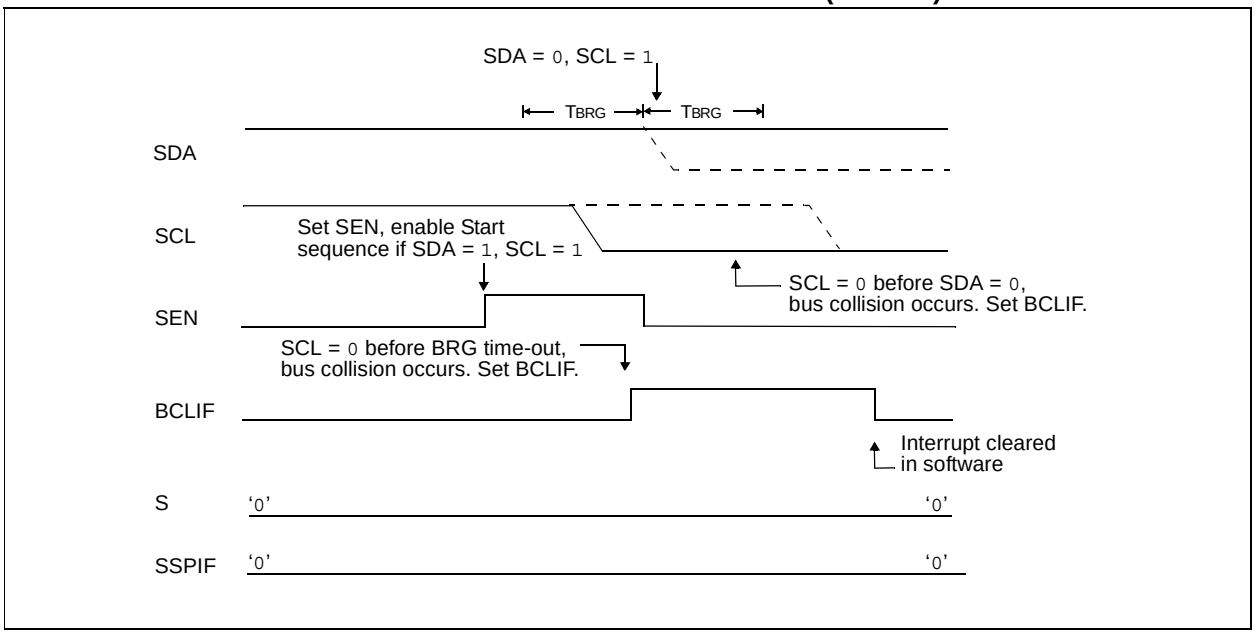
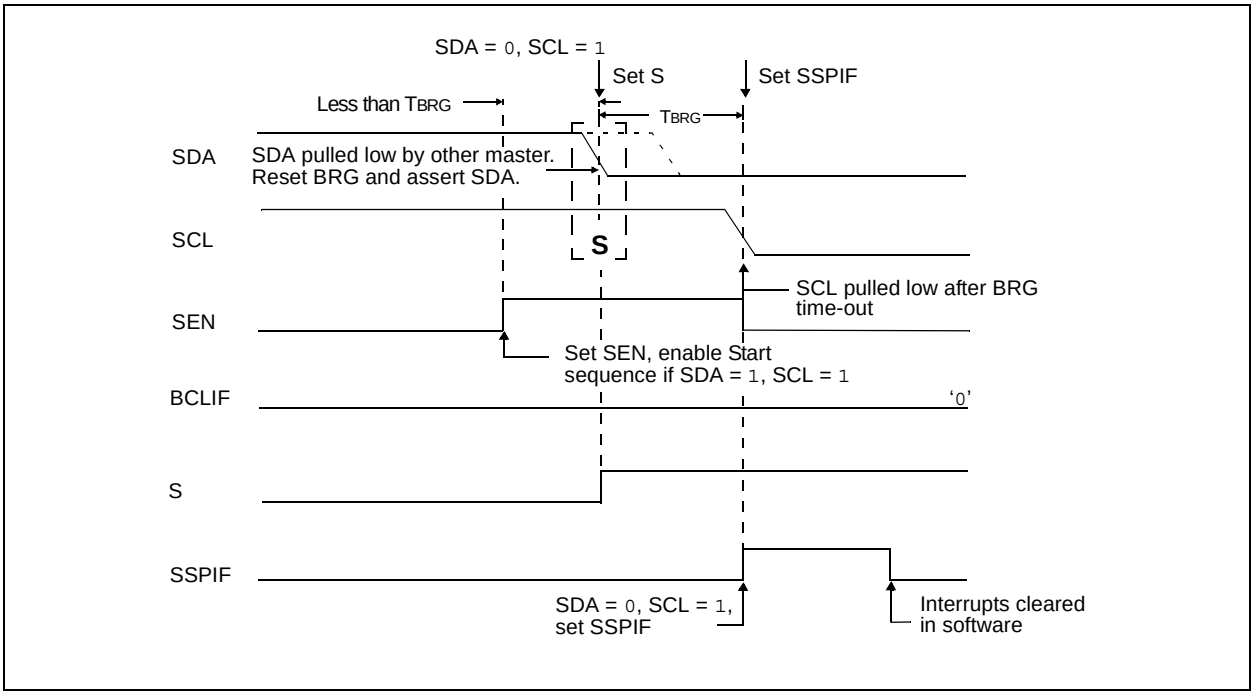



**FIGURE 10-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (see **Section 11.1 "AUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**FIGURE 11-2: ASYNCHRONOUS MASTER TRANSMISSION**

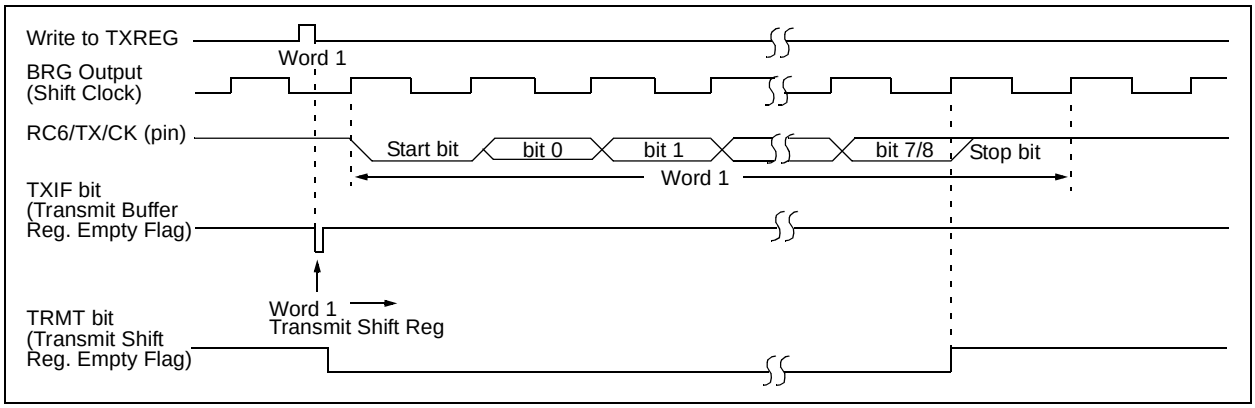
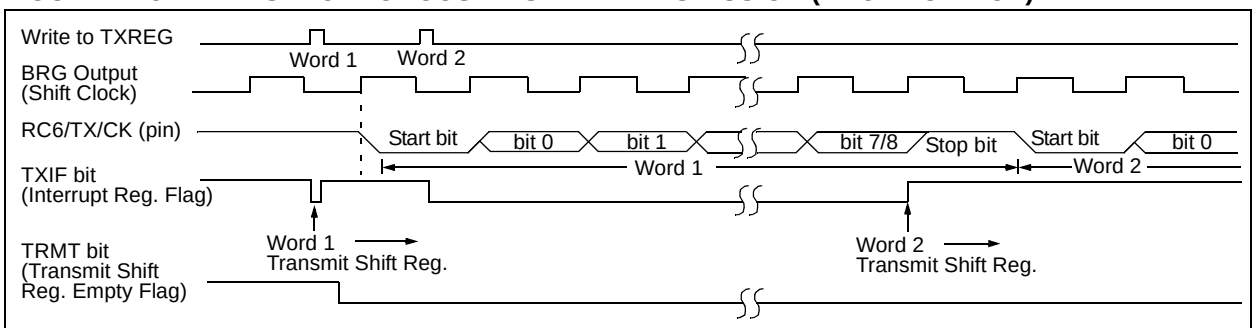



**FIGURE 11-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**




**Note:** This timing diagram shows two consecutive transmissions.

**TABLE 11-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | AUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

**Legend:** x = unknown, — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

### 11.3.2 AUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit, SREN (RCSTA<5>) or enable bit, CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit, RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then Overrun Error bit, OERR (RCSTA<1>), is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The ninth receive bit is buffered the same way as the receive data. Reading the RCREG register will load bit RX9D with a new value; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

When setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate (see **Section 11.1 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | AUSART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

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

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

## 12.2 Selecting and Configuring Automatic Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQT0 bits (ADCON2<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

## 12.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The seven possible options for TAD are:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module, RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μs.

Table 12-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

**TABLE 12-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (F))**

| AD Clock Source (TAD) | | Maximum Device Frequency |
|---|---|---|
| Operation | ADCS2:ADCS1:ADCS0 | |
| 2 TOSC | 000 | 1.25 MHz |
| 4 TOSC | 100 | 2.5 MHz |
| 8 TOSC | 001 | 5 MHz |
| 16 TOSC | 101 | 10 MHz |
| 32 TOSC | 010 | 20 MHz |
| 64 TOSC | 110 | 20 MHz |
| RC(1,2,3) | x11 | **(Note 1)** |

**Note 1:** The RC source has a typical TAD time of 4 μs but can vary between 2-6 μs.

**2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

**3:** For extended voltage devices (LF), please refer to **Section 18.0 "Electrical Characteristics"**.

## 15.0 SPECIAL FEATURES OF THE CPU

These devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
  - Low-Voltage Detect (LVD)
- Interrupts
- Watchdog Timer (WDT)
- Two-Speed Start-up
- Fail-Safe Clock Monitor
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT) which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in Reset while the power supply stabilizes and is enabled or disabled using a configuration bit. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low-current power-down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer wake-up or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. Configuration bits are used to select the desired oscillator mode.

Additional information on special features is available in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

### 15.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory locations 2007h and 2008h.

The user will note that address 2007h is beyond the user program memory space which can be accessed only during programming.

## 17.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 17.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

## 17.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 17.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 17.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 18.3 DC Characteristics: Internal RC Accuracy PIC16F737/747/767/777 (Industrial, Extended) PIC16LF737/747/767/777 (Industrial)

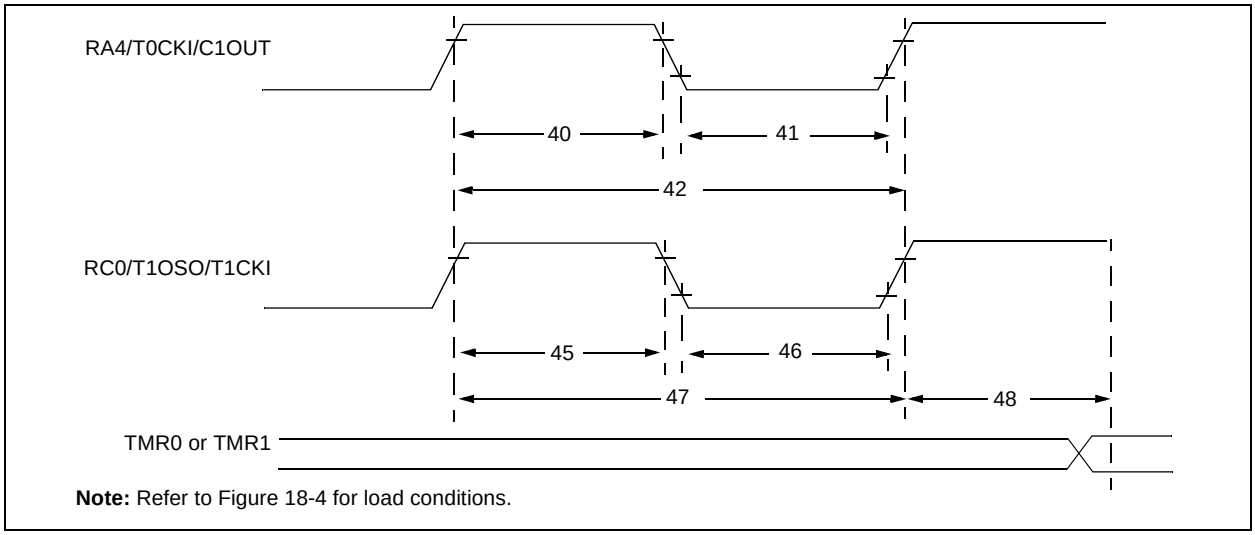
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LF737/747/767/777** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC16F737/747/767/777** (Industrial, Extended) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
| **Param No.** | **Device** | **Min** | **Typ** | **Max** | **Units** | **Conditions** | |
| | **INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz(1)** | | | | | | |
| | PIC16LF7X7 | -2 | ±1 | 2 | % | +25°C | VDD = 2.7V-3.3V |
| | | -5 | — | 5 | % | -10°C to +85°C | |
| | | -10 | — | 10 | % | -40°C to +85°C | |
| | PIC16F7X7 | -2 | ±1 | 2 | % | +25°C | VDD = 4.5V-5.5V |
| | | -5 | — | 5 | % | -10°C to +85°C | |
| | | -10 | — | 10 | % | -40°C to +85°C | |
| | Extended devices | -15 | — | 15 | % | -40°C to +125°C | |
| | **INTRC Accuracy @ Freq = 31 kHz(2)** | | | | | | |
| | PIC16LF7X7 | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 2.7V-3.3V |
| | PIC16F7X7 | 26.562 | — | 35.938 | kHz | -40°C to +85°C | VDD = 4.5V-5.5V |

**Legend:** Shading of rows is to assist in readability of the table.

**Note 1:** Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

**2:** INTRC is used to calibrate INTOSC.

**FIGURE 18-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




**Note:** Refer to Figure 18-4 for load conditions.

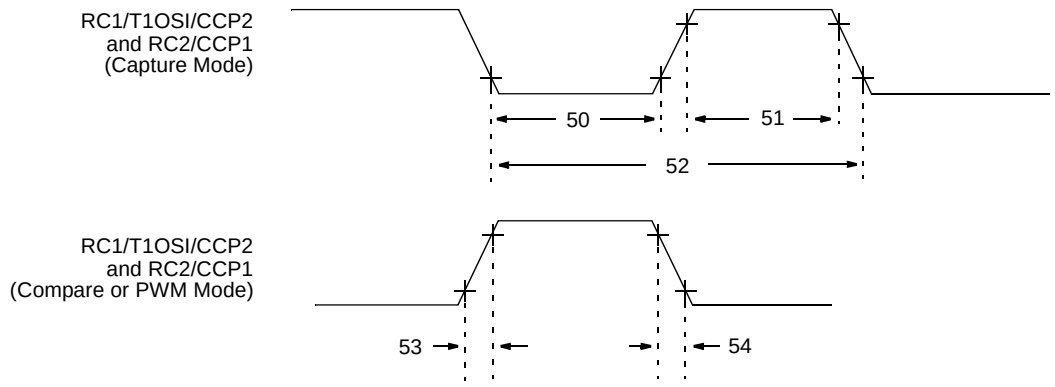
**TABLE 18-7: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

| Param No. | Symbol | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | TT0H | T0CKI High Pulse Width | | No prescaler | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With prescaler | 10 | — | — | ns | |
| 41* | TT0L | T0CKI Low Pulse Width | | No prescaler | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With prescaler | 10 | — | — | ns | |
| 42* | TT0P | T0CKI Period | | No prescaler | TCY + 40 | — | — | ns | |
| | | | | With prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | TT1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2, 4, 8 | PIC16F7X7 | 15 | — | — | ns | |
| | | | | PIC16LF7X7 | 25 | — | — | ns | |
| | | | Asynchronous | PIC16F7X7 | 30 | — | — | ns | |
| | | | | PIC16LF7X7 | 50 | — | — | ns | |
| 46* | TT1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5 TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2, 4, 8 | PIC16F7X7 | 15 | — | — | ns | |
| | | | | PIC16LF7X7 | 25 | — | — | ns | |
| | | | Asynchronous | PIC16F7X7 | 30 | — | — | ns | |
| | | | | PIC16LF7X7 | 50 | — | — | ns | |
| 47* | TT1P | T1CKI Input Period | Synchronous | PIC16F7X7 | Greater of: 30 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16LF7X7 | Greater of: 50 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16F7X7 | 60 | — | — | ns | |
| | | | | PIC16LF7X7 | 100 | — | — | ns | |
| | FT1 | Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZTMR1 | Delay from External Clock Edge to Timer Increment | | | 2 TOSC | — | 7 TOSC | — | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 18-10: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**




**Note:** Refer to Figure 18-4 for load conditions.

**TABLE 18-8: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)**

| Param No. | Symbol | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1, CCP2 and CCP3 Input Low Time | No prescaler | | 0.5 TCY + 20 | — | — | ns | |
| | | | With prescaler | PIC16F7X7 | 10 | — | — | ns | |
| | | | | PIC16LF7X7 | 20 | — | — | ns | |
| 51* | TccH | CCP1, CCP2 and CCP3 Input High Time | No prescaler | | 0.5 TCY + 20 | — | — | ns | |
| | | | With prescaler | PIC16F7X7 | 10 | — | — | ns | |
| | | | | PIC16LF7X7 | 20 | — | — | ns | |
| 52* | TccP | CCP1, CCP2 and CCP3 Input Period | | | $\frac{3\ T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 4 or 16) |
| 53* | TccR | CCP1, CCP2 and CCP3 Output Rise Time | | PIC16F7X7 | — | 10 | 25 | ns | |
| | | | | PIC16LF7X7 | — | 25 | 50 | ns | |
| 54* | TccF | CCP1, CCP2 and CCP3 Output Fall Time | | PIC16F7X7 | — | 10 | 25 | ns | |
| | | | | PIC16LF7X7 | — | 25 | 45 | ns | |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# INDEX

## A



## B

**W**

## PIC16F7X7 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | |
|---|---|
| Device | PIC16F7X7[(1)], PIC16F7X7T[(1)]; $V_{DD}$ range 4.0V to 5.5V<br>PIC16LF7X7[(1)], PIC16LF7X7T[(1)]; $V_{DD}$ range 2.0V to 5.5V |
| Temperature Range | I = -40°C to +85°C (Industrial)<br>E = -40°C to +125°C (Extended) |
| Package | ML = QFN (Micro Lead Frame)<br>PT = TQFP (Thin Quad Flatpack)<br>SO = SOIC<br>SP = Skinny Plastic DIP<br>P = PDIP<br>SS = SSOP |
| Pattern | QTP, SQTP, Code or Special Requirements (blank otherwise) |

**Examples:**

a) PIC16F777-I/P 301 = Industrial temp., PDIP package, normal $V_{DD}$ limits, QTP pattern #301.

b) PIC16LF767-I/SO = Industrial temp., SOIC package, extended $V_{DD}$ limits.

c) PIC16F747-E/P = Extended temp., PDIP package, normal $V_{DD}$ limits.

**Note 1:** F = CMOS Flash
LF = Low-Power CMOS Flash

**2:** T = in tape and reel – SOIC, SSOP, TQFP packages only.