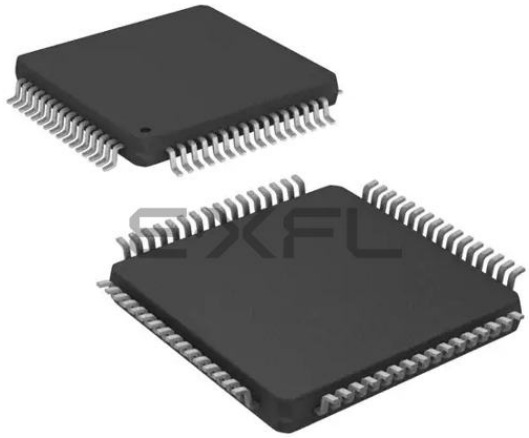
# Microchip Technology - PIC18F66K90T-I/PT Datasheet



**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

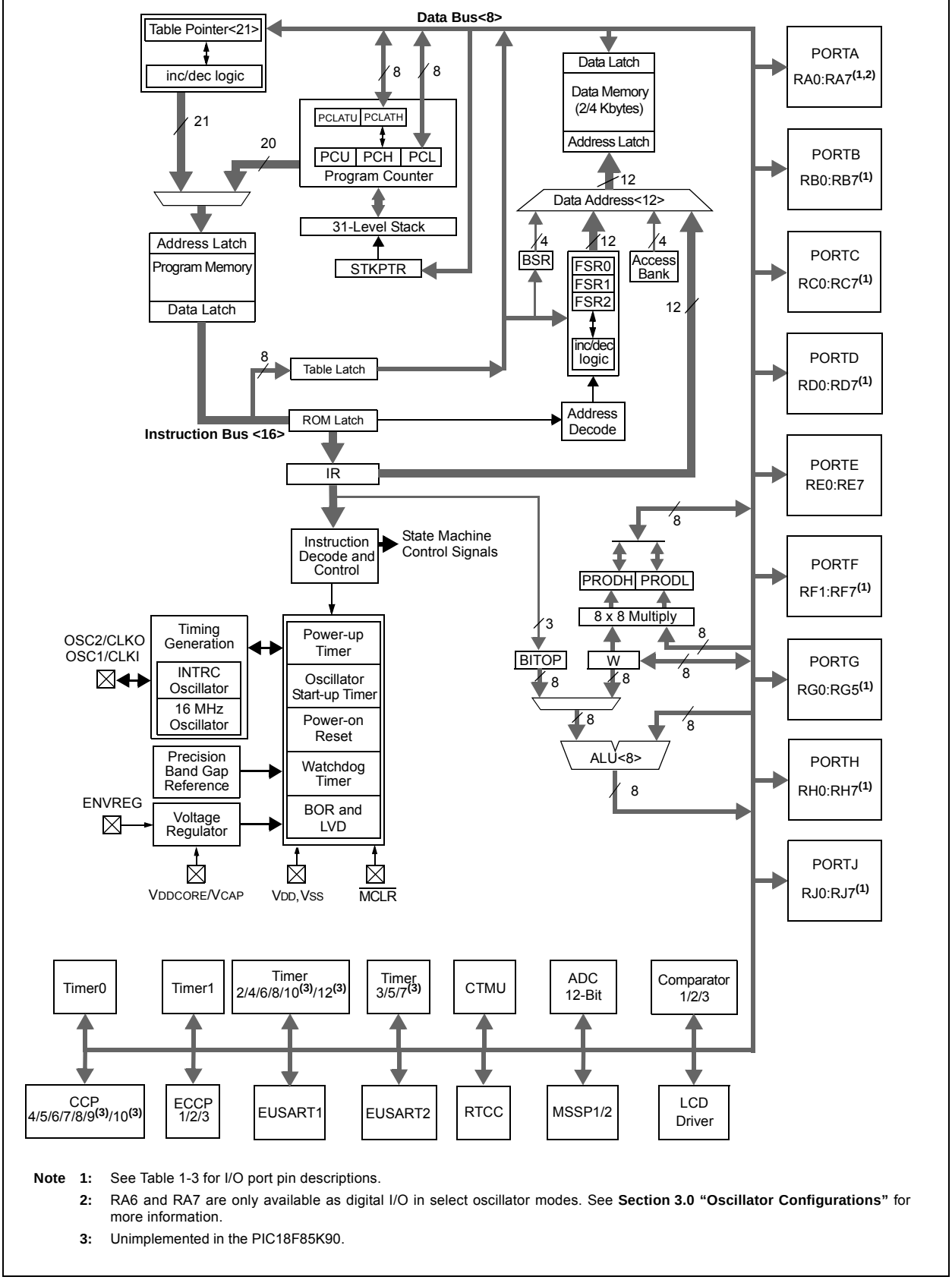
## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LCD, POR, PWM, WDT |
| Number of I/O | 53 |
| 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 ~ 5.5V |
| Data Converters | A/D 16x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-TQFP |
| Supplier Device Package | 64-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f66k90t-i-pt |

**NOTES:**


© 2009-2011 Microchip Technology Inc.

**FIGURE 1-2: PIC18F8XK90 (80-PIN) BLOCK DIAGRAM**




**Note 1:** See Table 1-3 for I/O port pin descriptions.

**2:** RA6 and RA7 are only available as digital I/O in select oscillator modes. See **Section 3.0 "Oscillator Configurations"** for more information.

**3:** Unimplemented in the PIC18F85K90.

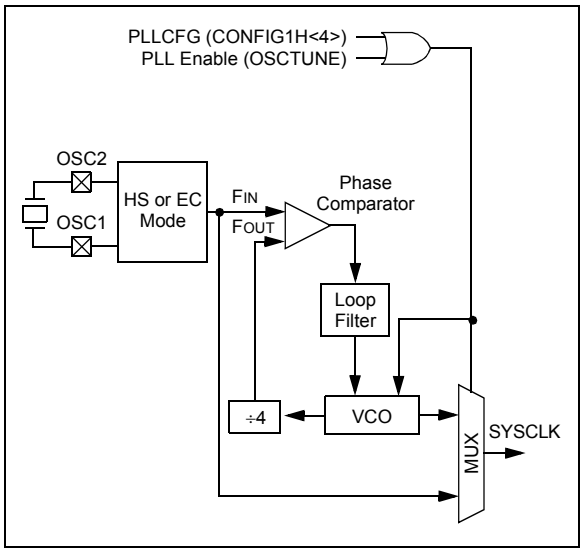
**TABLE 1-3: PIC18F6XK90 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|---|---|---|---|---|
| | QFN/TQFP | | | |
| | | | | PORTG is a bidirectional I/O port. |
| RG0/ECCP3/P3A | 3 | | | |
| RG0 | | I/O | ST | Digital I/O. |
| ECCP3 | | I/O | ST | Capture 3 input/Compare 3 output/PWM3 output. |
| P3A | | O | — | ECCP3 PWM Output A. |
| RG1/TX2/CK2/AN19/ C3OUT | 4 | | | |
| RG1 | | I/O | ST | Digital I/O. |
| TX2 | | O | — | USART asynchronous transmit. |
| CK2 | | I/O | ST | USART synchronous clock (see related RX2/DT2). |
| AN19 | | I | Analog | Analog Input 19. |
| C3OUT | | O | — | Comparator 3 output. |
| RG2/RX2/DT2/AN18/ C3INA | 5 | | | |
| RG2 | | I/O | ST | Digital I/O. |
| RX2 | | I | ST | EUSART asynchronous receive. |
| DT2 | | I/O | ST | EUSART synchronous data (see related TX2/CK2). |
| AN18 | | I | Analog | Analog Input 18. |
| C3INA | | I | Analog | Comparator 3 Input A. |
| RG3/CCP4/AN17/P3D/ C3INB | 6 | | | |
| RG3 | | I/O | ST | Digital I/O. |
| CCP4 | | I/O | S/T | Capture 4 input/Compare 4 output/PWM4 output. |
| AN17 | | I | Analog | Analog Input 18. |
| P3D | | O | — | ECCP3 PWM Output D. |
| C3INB | | I | Analog | Comparator 3 Input B. |
| RG4/SEG26/RTCC/ T7CKI/T5G/CCP5/AN16/ P1D/C3INC | 8 | | | |
| RG4 | | I/O | ST | Digital I/O. |
| SEG26 | | O | Analog | SEG26 output for LCD. |
| RTCC | | O | — | RTCC output |
| T7CKI(3) | | I | ST | Timer7 clock input. |
| T5G | | I | ST | Timer5 external clock gate input. |
| CCP5 | | I/O | ST | Capture 5 input/Compare 5 output/PWM5 output. |
| AN16 | | I | Analog | Analog Input 16. |
| P1D | | O | — | ECCP1 PWM Output D. |
| C3INC | | I | Analog | Comparator 3 Input C. |
| RG5 | 7 | | | See the $\overline{\text{MCLR}}$/RG5 pin. |

**Legend:** TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to $V_{DD}$); $I^2C$™ = $I^2C$/SMBus

**Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.
**2:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.
**3:** Not available on PIC18F65K90 and PIC18F85K90 devices.

**FIGURE 3-7: PLL BLOCK DIAGRAM**




#### 3.5.2.2 PLL and HF-INTOSC

The PLL is available to the internal oscillator block when the internal oscillator block is configured as the primary clock source. In this configuration, the PLL is enabled in software and generates a clock output of up to 64 MHz.

The operation of INTOSC with the PLL is described in **Section 3.6.2 "INTPLL Modes"**. Care should be taken that the PLL is enabled only if the HF-INTOSC postscaler is configured for 8 MHz or 16 MHz.

## 3.6 Internal Oscillator Block

The PIC18F87K90 family of devices includes an internal oscillator block which generates two different clock signals. Either clock can be used as the microcontroller's clock source, which may eliminate the need for an external oscillator circuit on the OSC1 and/or OSC2 pins.

The internal oscillator consists of three blocks, depending on the frequency of operation. They are HF-INTOSC, MF-INTOSC and LF-INTRC.

In HF-INTOSC mode, the internal oscillator can provide a frequency, ranging from 31 kHz to 16 MHz, with the postscaler deciding the selected frequency (IRCF<2:0>).

The INTSRC bit (OSCTUNE<7>) and MFIOSEL bit (OSCCON2<0>) also decide which INTOSC provides the lower frequency (500 kHz to 31 kHz). For the HF-INTOSC to provide these frequencies, INTSRC = 1 and MFI0SEL = 0.

In HF-INTOSC, the postscaler (IRCF<2:0>) provides the frequency range of 31 kHz to 16 MHz. If HF-INTOSC is used with the PLL, the input frequency to the PLL should be 8 MHz or 16 MHz (IRCF<2:0> = 111, 110).

For MF-INTOSC mode to provide a frequency range of 500 kHz to 31 kHz, INTSRC = 1 and MFIOSEL = 1. The postscaler (IRCF<2:0>), in this mode, provides the frequency range of 31 kHz to 500 kHz.

The LF-INTRC can provide only 31 kHz if INTSRC = 0.

The LF-INTRC provides 31 kHz and is enabled if it is selected as the device clock source. The mode is enabled automatically when any of the following is enabled:

- Power-up Timer
- Fail-Safe Clock Monitor
- Watchdog Timer
- Two-Speed Start-up

These features are discussed in greater detail in **Section 28.0 "Special Features of the CPU"**.

The clock source frequency (HF-INTOSC, MF-INTOSC or LF-INTRC direct) is selected by configuring the IRCF bits of the OSCCON register, as well the INTSRC and MFIOSEL bits. The default frequency on device Resets is 8 MHz.

### 3.6.1 INTIO MODES

Using the internal oscillator as the clock source eliminates the need for up to two external oscillator pins, which can then be used for digital I/O. Two distinct oscillator configurations, which are determined by the OSC Configuration bits, are available:

- In INTIO1 mode, the OSC2 pin (RA6) outputs FOSC/4, while OSC1 functions as RA7 (see Figure 3-8) for digital input and output.
- In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6 (see Figure 3-9). Both are available as digital input and output ports.

**FIGURE 3-8: INTIO1 OSCILLATOR MODE**

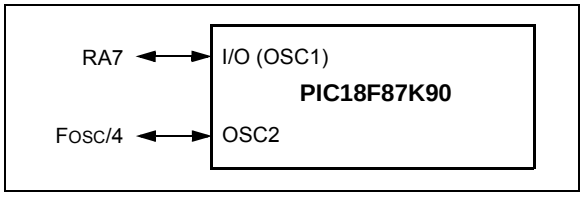



**FIGURE 3-9: INTIO2 OSCILLATOR MODE**

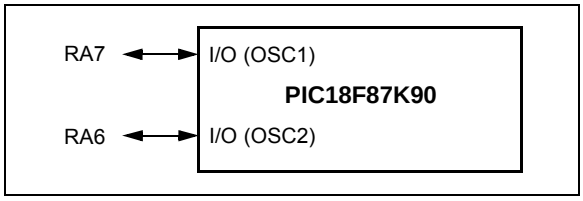


**TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page: |
|---|---|---|---|---|---|---|---|---|---|
| PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | 78 |
| LATD | LATD7 | LATD6 | LATD5 | LATD4 | LATD3 | LATD2 | LATD1 | LATD0 | 78 |
| TRISD | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 78 |
| LCDSE0 | SE07 | SE06 | SE05 | SE04 | SE03 | SE02 | SE01 | SE00 | 83 |
| PADCFG1 | RDPU | REPU | RJPU(1) | — | — | RTSECSEL1 | RTSECSEL0 | — | 80 |

**Legend:** Shaded cells are not used by PORTD.

**Note 1:** This bit is not available in 64-pin devices.


© 2009-2011 Microchip Technology Inc.

**NOTES:**


© 2009-2011 Microchip Technology Inc.

## 15.3 Timer3/5/7 16-Bit Read/Write Mode

Timer3/5/7 can be configured for 16-bit reads and writes (see Figure 15.3). When the RD16 control bit (TxCON<1>) is set, the address for TMRxH is mapped to a buffer register for the high byte of Timer3/5/7. A read from TMRxL will load the contents of the high byte of Timer3/5/7 into the Timerx High Byte Buffer register. This provides users with the ability to accurately read all 16 bits of Timer3/5/7 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer3/5/7 must also take place through the TMRxH Buffer register. The Timer3/5/7 high byte is updated with the contents of TMRxH when a write occurs to TMRxL. This allows users to write all 16 bits to both the high and low bytes of Timer3/5/7 at once.

The high byte of Timer3/5/7 is not directly readable or writable in this mode. All reads and writes must take place through the Timerx High Byte Buffer register.

Writes to TMRxH do not clear the Timer3/5/7 prescaler. The prescaler is only cleared on writes to TMRxL.

## 15.4 Using the SOSC Oscillator as the Timer3/5/7 Clock Source

The SOSC internal oscillator may be used as the clock source for Timer3/5/7. The SOSC oscillator is enabled by setting one of five bits: any of the four SOSCEN bits in the TxCON registers (TxCON<3>) or the SOSCGO bit in the OSCCON2 register (OSCCON2<3>). To use it as the Timer3/5/7 clock source, the TMRxCS bit must also be set. As previously noted, this also configures Timer3/5/7 to increment on every rising edge of the oscillator source.

The SOSC oscillator is described in **Section 13.0 "Timer1 Module"**.

## 15.5 Timer3/5/7 Gates

Timer3/5/7 can be configured to count freely or the count can be enabled and disabled using the Timer3/5/7 gate circuitry. This is also referred to as the Timer3/5/7 gate count enable.

The Timer3/5/7 gate can also be driven by multiple selectable sources.

### 15.5.1 TIMER3/5/7 GATE COUNT ENABLE

The Timerx Gate Enable mode is enabled by setting the TMRxGE bit (TxGCON<7>). The polarity of the Timerx Gate Enable mode is configured using the TxGPOL bit (TxGCON<6>).

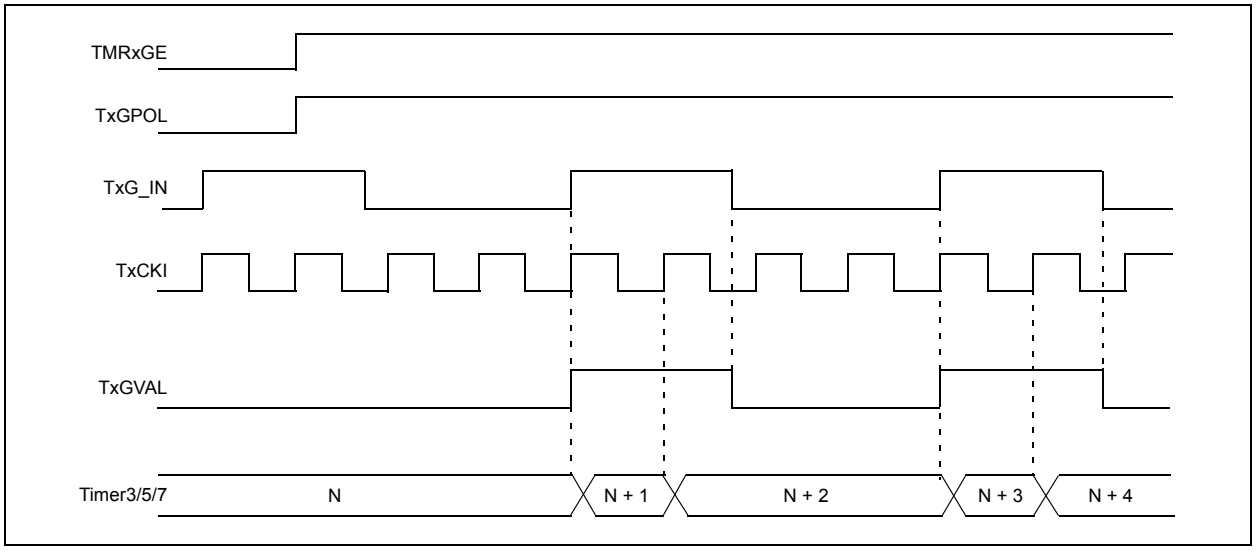
When Timerx Gate Enable mode is enabled, Timer3/5/7 will increment on the rising edge of the Timer3/5/7 clock source. When Timerx Gate Enable mode is disabled, no incrementing will occur and Timer3/5/7 will hold the current count. See Figure 15-2 for timing details.

**TABLE 15-1: TIMER3/5/7 GATE ENABLE SELECTIONS**

| TxCLK(†) | TxGPOL (TxGCON<6>) | TxG Pin | Timerx Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| | 1 | 1 | Counts |

† The clock on which TMR3/5/7 is running. For more information, see TxCLK in Figure 15-1.

**FIGURE 15-2: TIMER3/5/7 GATE COUNT ENABLE MODE**





© 2009-2011 Microchip Technology Inc.

### 20.3.2 INTERNAL RESISTOR BIASING

This mode does not use external resistors, but rather internal resistor ladders that are configured to generate the bias voltage.

The internal reference ladder actually consists of three separate ladders. Disabling the internal reference ladder disconnects all of the ladders, allowing external voltages to be supplied.

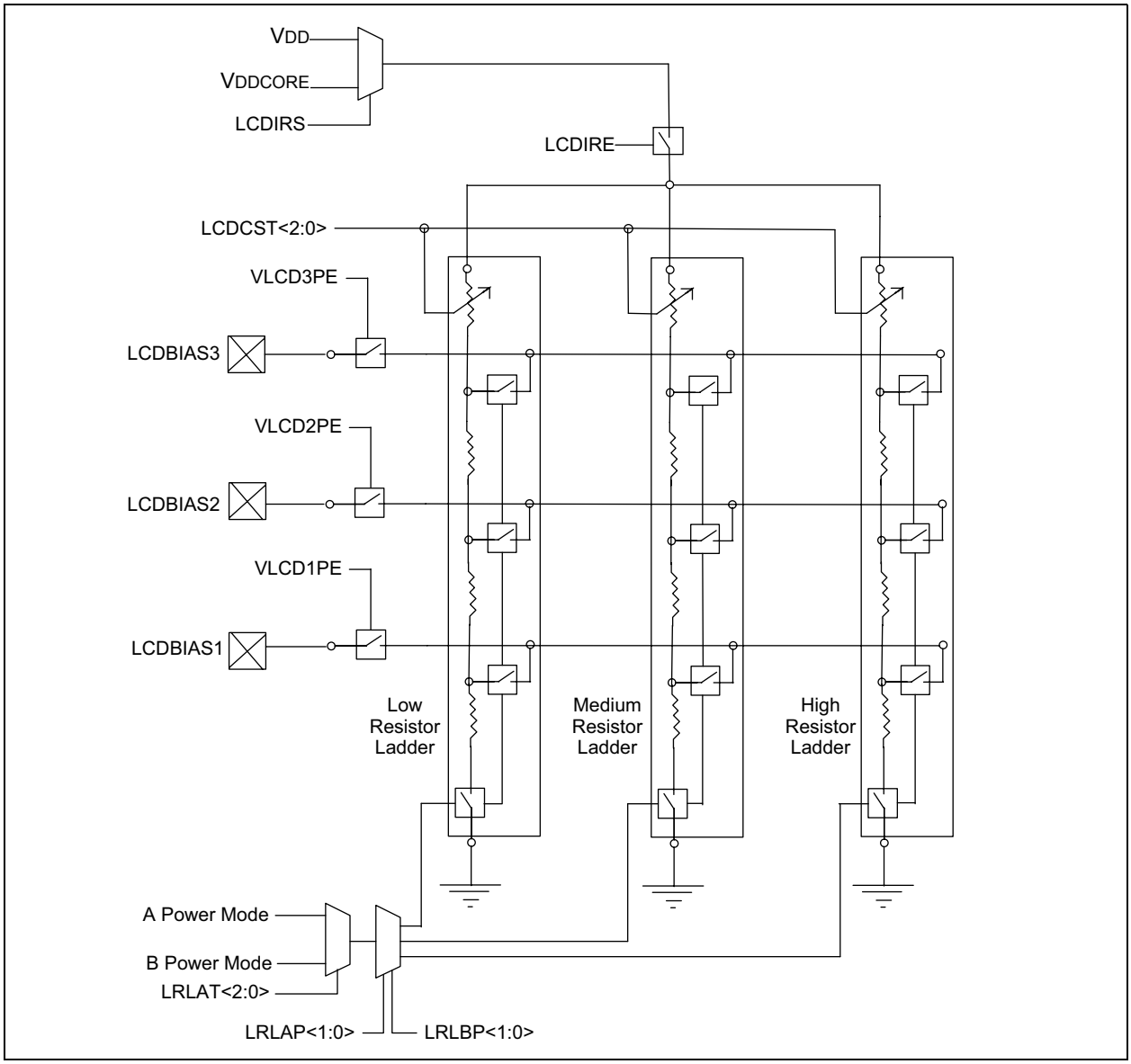
Depending on the total resistance of the resistor ladders, the biasing can be classified as low, medium or high power.

Table 20-3 shows the total resistance of each of the ladders. Figure 20-4 shows the internal resister ladder connections. When the internal resistor ladder is selected, the bias voltage can either be from VDD or from VDDCORE, depending on the LCDIRS setting.

**TABLE 20-3: INTERNAL RESISTANCE LADDER POWER MODES**

| Power Mode | Nominal Resistance of Entire Ladder | IDD |
|---|---|---|
| Low | 3 MΩ | 1 μA |
| Medium | 300 kΩ | 10 μA |
| High | 30 kΩ | 100 μA |

**FIGURE 20-4: LCD BIAS INTERNAL RESISTOR LADDER CONNECTION DIAGRAM**





© 2009-2011 Microchip Technology Inc.

## 20.7 LCD Frame Frequency

The rate at which the COM and SEG outputs change is called the LCD frame frequency.

**TABLE 20-5: FRAME FREQUENCY FORMULAS**

| Multiplex | Frame Frequency = |
|---|---|
| Static | Clock Source/(4 x 1 x (LP<3:0> + 1)) |
| 1/2 | Clock Source/(2 x 2 x (LP<3:0> + 1)) |
| 1/3 | Clock Source/(1 x 3 x (LP<3:0> + 1)) |
| 1/4 | Clock Source/(1 x 4 x (LP<3:0> + 1)) |

**Note:** Clock source is (FOSC/4)/8192, Timer1 Osc/32 or INTRC/32.

**TABLE 20-6: APPROXIMATE FRAME FREQUENCY (IN Hz) USING FOSC AT 32 MHz, TIMER1 AT 32.768 kHz OR INTRC OSC**

| LP<3:0> | Static | 1/2 | 1/3 | 1/4 |
|---|---|---|---|---|
| 1 | 125 | 125 | 167 | 125 |
| 2 | 83 | 83 | 111 | 83 |
| 3 | 62 | 62 | 83 | 62 |
| 4 | 50 | 50 | 67 | 50 |
| 5 | 42 | 42 | 56 | 42 |
| 6 | 36 | 36 | 48 | 36 |
| 7 | 31 | 31 | 42 | 31 |

## 20.8 LCD Waveform Generation

LCD waveform generation is based on the philosophy that the net AC voltage across the dark pixel should be maximized and the net AC voltage across the clear pixel should be minimized. The net DC voltage across any pixel should be zero.

The COM signal represents the time slice for each common, while the SEG contains the pixel data.

The pixel signal (COM-SEG) will have no DC component and can take only one of the two rms values. The higher rms value will create a dark pixel and a lower rms value will create a clear pixel.

As the number of commons increases, the delta between the two rms values decreases. The delta represents the maximum contrast that the display can have.

The LCDs can be driven by two types of waveforms: Type-A and Type-B. In a Type-A waveform, the phase changes within each common type, whereas a Type-B waveform's phase changes on each frame boundary. Thus, Type-A waveforms maintain 0 VDC over a single frame, whereas Type-B waveforms take two frames.

**Note 1:** If Sleep has to be executed with LCD Sleep enabled (SLPEN (LCDCON<6>) = 1), care must be taken to execute Sleep only when VDC on all the pixels is '0'.

**2:** When the LCD clock source is (FOSC/4)/8192, if Sleep is executed irrespective of the LCDCON<SLPEN> setting, the LCD goes into Sleep. Thus, take care to see that VDC on all pixels is '0' when Sleep is executed.

Figure 20-7 through Figure 20-17 provide waveforms for static, half-multiplex, one-third multiplex and quarter multiplex drives for Type-A and Type-B waveforms.


© 2009-2011 Microchip Technology Inc.

### 21.3.1 REGISTERS

Each MSSP module has four registers for SPI mode operation. These are:

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Status Register (SSPxSTAT)
- Serial Receive/Transmit Buffer Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible

SSPxCON1 and SSPxSTAT are the control and status registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower 6 bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

SSPxSR is the shift register used for shifting data in or out. SSPxBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPxSR and SSPxBUF together create a double-buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not double-buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

| **Note:** | The SSPxBUF register cannot be used with read-modify-write instructions, such as BCF, COMF, etc.<br>To avoid lost data in Master mode, a read of the SSPxBUF must be performed to clear the Buffer Full (BF) detect bit (SSPxSTAT<0>) between each transmission. |
|---|---|

**REGISTER 21-1: SSPxSTAT: MSSPx STATUS REGISTER (SPI MODE)**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE(1) | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| 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 **SMP:** Sample bit
SPI Master mode:
1 = Input data sampled at the end of data output time
0 = Input data sampled at the middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Select bit(1)
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
Used in I²C™ mode only.

bit 4 **P:** Stop bit
Used in I²C mode only. This bit is cleared when the MSSPx module is disabled; SSPEN is cleared.

bit 3 **S:** Start bit
Used in I²C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit
Used in I²C mode only.

bit 1 **UA:** Update Address bit
Used in I²C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty

**Note 1:** Polarity of the clock state is set by the CKP bit (SSPxCON1<4>).


© 2009-2011 Microchip Technology Inc.

### 21.3.4 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPxCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPxCON registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and $\overline{SS}$x pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDIx must have TRISC<4> or TRISD<5> bit set
- SDOx must have the TRISC<5> or TRISD<4> bit cleared
- SCKx (Master mode) must have the TRISC<3> or TRISD<6>bit cleared
- SCKx (Slave mode) must have the TRISC<3> or TRISD<6> bit set
- $\overline{SS}$x must have the TRISF<7> or TRISD<7> bit set

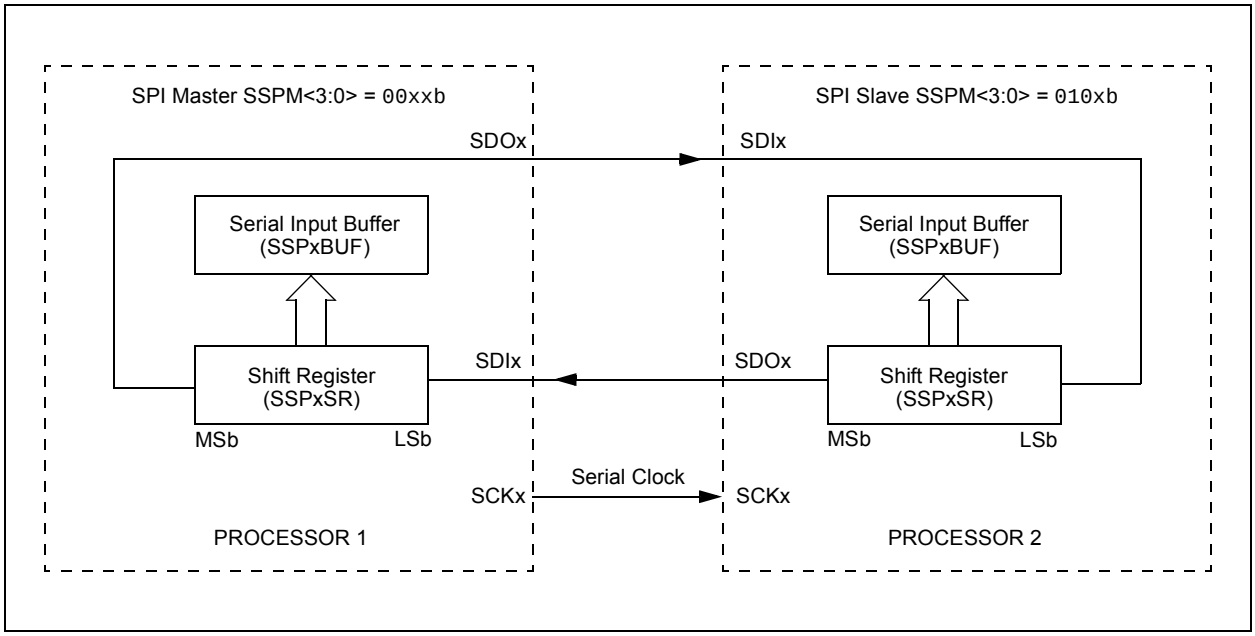
Any serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value.

### 21.3.5 TYPICAL CONNECTION

Figure 21-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 21-2: SPI MASTER/SLAVE CONNECTION**



#### 21.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDAx when SCLx goes from a low level to a high level.

b) SCLx goes low before SDAx is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDAx and the pin is allowed to float high, the BRG is loaded with SSPxADD<6:0> and counts down to 0. The SCLx pin is then deasserted and when sampled high, the SDAx pin is sampled.

If SDAx is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 21-31). If SDAx is sampled high, the BRG is reloaded and begins counting. If SDAx goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDAx at exactly the same time.

If SCLx goes from high-to-low before the BRG times out and SDAx has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (see Figure 21-32).

If, at the end of the BRG time-out, both SCLx and SDAx are still high, the SDAx pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCLx pin, the SCLx pin is driven low and the Repeated Start condition is complete.

**FIGURE 21-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

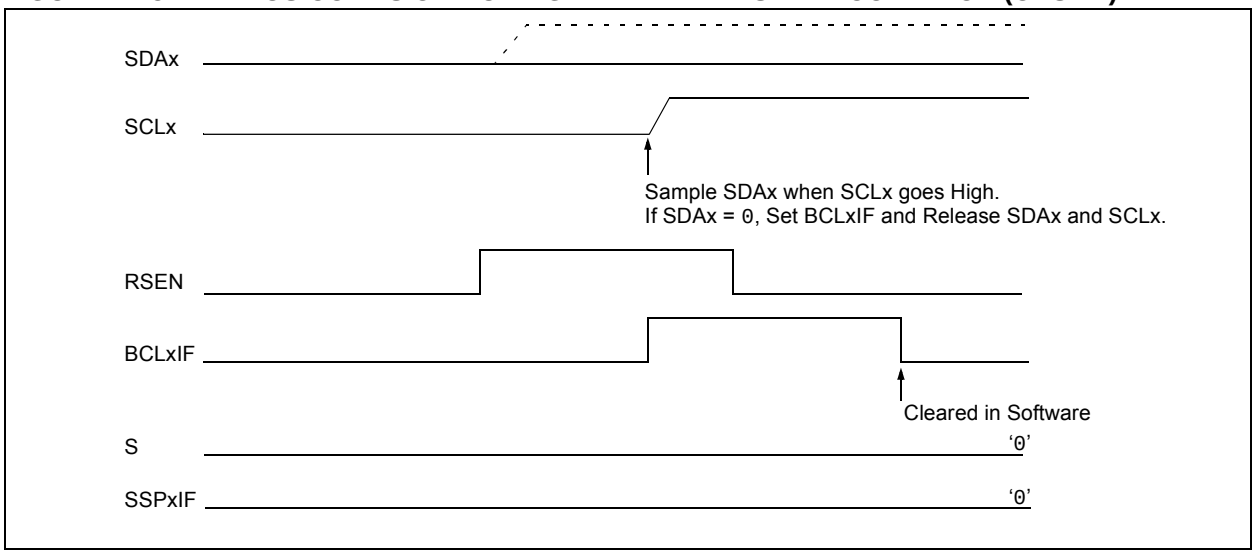
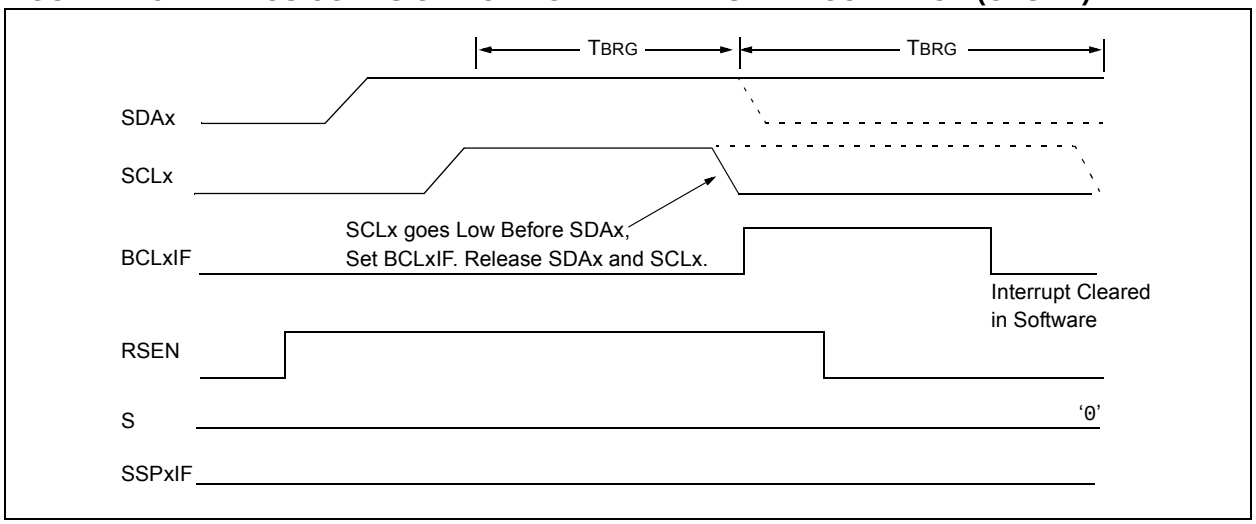



**FIGURE 21-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**





© 2009-2011 Microchip Technology Inc.

**EXAMPLE 22-1: CALCULATING BAUD RATE ERROR**

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, and 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGHx:SPBRGx] + 1))

Solving for SPBRGHx:SPBRGx:

X = ((FOSC/Desired Baud Rate)/64) – 1

= ((16000000/9600)/64) – 1

= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))

= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate

= (9615 – 9600)/9600 = 0.16%

**TABLE 22-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page: |
|---|---|---|---|---|---|---|---|---|---|
| TXSTA1 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 77 |
| RCSTA1 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 77 |
| BAUDCON1 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 79 |
| SPBRGH1 | EUSART1 Baud Rate Generator Register High Byte | | | | | | | | 76 |
| SPBRG1 | EUSART1 Baud Rate Generator Register Low Byte | | | | | | | | 77 |
| TXSTA2 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 81 |
| RCSTA2 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 81 |
| BAUDCON2 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 81 |
| SPBRGH2 | EUSART2 Baud Rate Generator Register High Byte | | | | | | | | 82 |
| SPBRG2 | EUSART2 Baud Rate Generator Register Low Byte | | | | | | | | 82 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the BRG.


© 2009-2011 Microchip Technology Inc.

### 22.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode, and bit, SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREGx register. If the RCxIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RCxIE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RCxIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCxIE, was set.
6. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREGx register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 22-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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 | 75 |
| PIR1 | — | ADIF | RC1IF | TX1IF | SSP1IF | TMR1GIF | TMR2IF | TMR1IF | 77 |
| PIE1 | — | ADIE | RC1IE | TX1IE | SSP1IE | TMR1GIE | TMR2IE | TMR1IE | 77 |
| IPR1 | — | ADIP | RC1IP | TX1IP | SSP1IP | TMR1GIP | TMR2IP | TMR1IP | 77 |
| PIR3 | TMR5GIF | LCDIF | RC2IF | TX2IF | CTMUIF | CCP2IF | CCP1IF | RTCCIF | 77 |
| PIE3 | TMR5GIE | LCDIE | RC2IE | TX2IE | CTMUIE | CCP2IE | CCP1IE | RTCCIE | 77 |
| IPR3 | TMR5GIP | LCDIP | RC2IP | TX2IP | CTMUIP | CCP2IP | CCP1IP | RTCCIP | 77 |
| RCSTA1 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 77 |
| RCREG1 | EUSART1 Receive Register | | | | | | | | 77 |
| TXSTA1 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 77 |
| BAUDCON1 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 79 |
| SPBRGH1 | EUSART1 Baud Rate Generator Register High Byte | | | | | | | | 76 |
| SPBRG1 | EUSART1 Baud Rate Generator Register Low Byte | | | | | | | | 77 |
| RCSTA2 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 81 |
| RCREG2 | EUSART2 Receive Register | | | | | | | | 82 |
| TXSTA2 | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 81 |
| BAUDCON2 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 81 |
| SPBRGH2 | EUSART2 Baud Rate Generator Register High Byte | | | | | | | | 82 |
| SPBRG2 | EUSART2 Baud Rate Generator Register Low Byte | | | | | | | | 82 |

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


© 2009-2011 Microchip Technology Inc.

## 26.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

The PIC18F87K90 family of devices has a High/Low-Voltage Detect module (HLVD). This is a programmable circuit that sets both a device voltage trip point and the direction of change from that point. If the device experiences an excursion past the trip point in that direction, an interrupt flag is set. If the interrupt is enabled, the program execution branches to the interrupt vector address and the software responds to the interrupt.

The High/Low-Voltage Detect Control register (Register 26-1) completely controls the operation of the HLVD module. This allows the circuitry to be "turned off" by the user under software control, which minimizes the current consumption for the device.

The module's block diagram is shown in Figure 26-1.

**REGISTER 26-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER**

| R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| VDIRMAG | BGVST | IRVST | HLVDEN | HLVDL3(1) | HLVDL2(1) | HLVDL1(1) | HLVDL0(1) |
| 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 **VDIRMAG:** Voltage Direction Magnitude Select bit
1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Reference Voltages Stable Status Flag bit
1 = Internal band gap voltage references are stable
0 = Internal band gap voltage references are not stable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the voltage detect logic will generate the interrupt flag at the specified voltage range
0 = Indicates that the voltage detect logic will not generate the interrupt flag at the specified voltage range and the HLVD interrupt should not be enabled

bit 4 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
1 = HLVD is enabled
0 = HLVD is disabled

bit 3-0 **HLVDL<3:0>:** Voltage Detection Limit bits(1)
1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Maximum setting
.
.
.
0000 = Minimum setting

**Note 1:** For the electrical specifications, see Parameter D420.

**REGISTER 28-14: DEVID1: DEVICE ID REGISTER 1 FOR THE PIC18F87K90 FAMILY**

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV2 | DEV1 | DEV0 | REV4 | REV3 | REV2 | REV1 | REV0 |
| 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 **DEV<2:0>:** Device ID bits

Devices with DEV<10:3> of 0101 0010 (see DEVID2):
010 = PIC18F65K90
000 = PIC18F66K90
101 = PIC18F85K90
011 = PIC18F86K90

Devices with DEV<10:3> of 0101 0001:
000 = PIC18F67K90
010 = PIC18F87K90

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

**REGISTER 28-15: DEVID2: DEVICE ID REGISTER 2 FOR THE PIC18F87K90 FAMILY**

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV10(1) | DEV9(1) | DEV8(1) | DEV7(1) | DEV6(1) | DEV5(1) | DEV4(1) | DEV3(1) |
| 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 **DEV<10:3>:** Device ID bits(1)

These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.
0101 0010 = PIC18F65K90, PIC18F66K90, PIC18F85K90 and PIC18F86K90
0101 0001 = PIC18F67K90 and PIC18F87K90

**Note 1:** These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

**INCFSZ** **Increment f, Skip if 0**

Syntax: INCFSZ f {,d {,a}}

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

Operation: (f) + 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

| 0011 | 11da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

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 ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

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

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    INCFSZ   CNT, 1, 0
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT + 1
If CNT = 0;
PC = Address (ZERO)
If CNT ≠ 0;
PC = Address (NZERO)

**INFSNZ** **Increment f, Skip if Not 0**

Syntax: INFSNZ f {,d {,a}}

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

Operation: (f) + 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

| 0100 | 10da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

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 ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

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

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

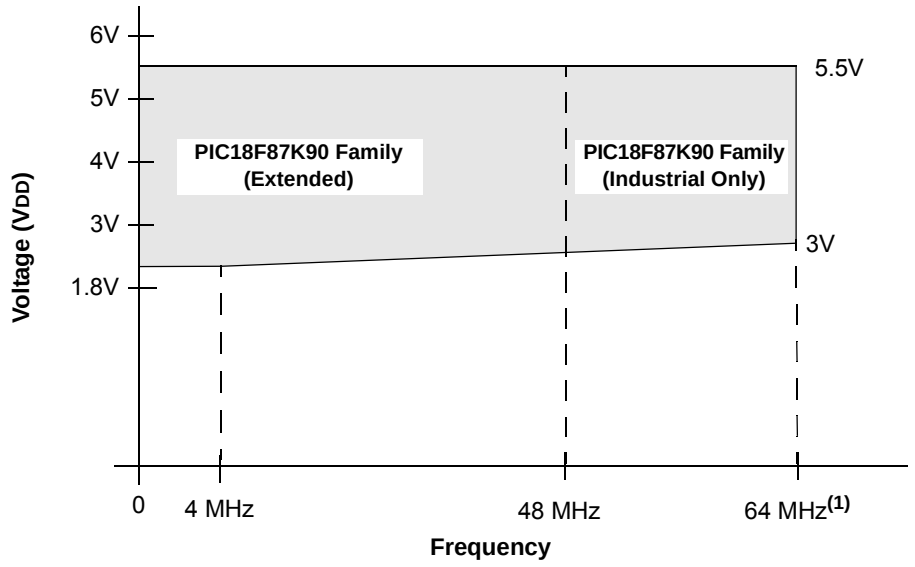
If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    INFSNZ  REG, 1, 0
ZERO
NZERO
```

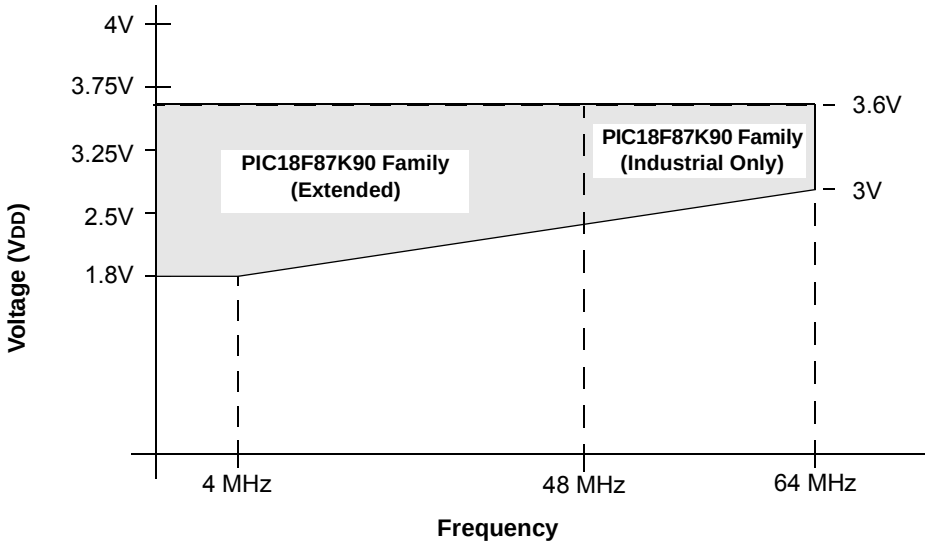
Before Instruction
PC = Address (HERE)
After Instruction
REG = REG + 1
If REG ≠ 0;
PC = Address (NZERO)
If REG = 0;
PC = Address (ZERO)

**FIGURE 31-1: VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL/EXTENDED)(1)**




**Note 1:** FMAX = 64 MHz in all other modes. For VDD values, 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

**FIGURE 31-2: VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL/EXTENDED)(1,2)**




**Note 1:** When the on-chip voltage regulator is disabled, VDD must be maintained so that VDD ≤ 3.6V.

**2:** For VDD values, 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.


© 2009-2011 Microchip Technology Inc.

**TABLE 31-1: MEMORY PROGRAMMING REQUIREMENTS**

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | | **Internal Program Memory Programming Specifications(1)** | | | | | |
| D110 | VPP | Voltage on $\overline{\text{MCLR}}$/VPP/RE5 pin | VDD + 1.5 | — | 9 | V | **(Note 3)** |
| D113 | IDDP | Supply Current during Programming | — | — | 10 | mA | |
| | | **Data EEPROM Memory** | | | | | **(Note 2)** |
| D120 | ED | Byte Endurance | 100K | — | — | E/W | -40°C to +125°C |
| D121 | VDRW | VDD for Read/Write | 1.8 | — | 5.5 | V | Using EECON to read/write ENVREG tied to VDD |
| | | | 1.8 | — | 3.6 | V | Using EECON to read/write ENVREG tied to VSS |
| D122 | TDEW | Erase/Write Cycle Time | — | 4 | — | ms | |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh(2) | 1M | 10M | — | E/W | -40°C to +125°C |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Cell Endurance | 10K | — | — | E/W | -40°C to +125°C |
| D131 | VPR | VDD for Read | 1.8 | — | 5.5 | V | ENVREG tied to VDD |
| | | | 1.8 | — | 3.6 | V | ENVREG tied to VSS |
| D132B | VPEW | Voltage for Self-Timed Erase or Write Operations<br>VDD | 1.8 | — | 5.5 | V | ENVREG tied to VDD |
| D133A | TIW | Self-Timed Write Cycle Time | — | 2 | — | ms | |
| D134 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D135 | IDDP | Supply Current during Programming | — | — | 10 | mA | |
| D140 | TWE | Writes per Erase Cycle | — | — | 1 | | For each physical address |

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** These specifications are for programming the on-chip program memory through the use of table write instructions.

**2:** Refer to **Section 8.8 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

**3:** The MPLAB® ICD 2 does not support variable VPP output. Circuitry to limit the ICD 2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD 2.

## APPENDIX A: REVISION HISTORY

### Revision A (September 2009)

Original data sheet for PIC18F87K90 family devices.

### Revision B (April 2010)

Changes to **Section 32.0 "Packaging Information"**, including new packaging diagrams. Changes to some of the values in **Section 31.0 "Electrical Characteristics"**. The new **Section 2.0 "Guidelines for Getting Started with PIC18FXXKXX Microcontrollers"** has been added. Minor text edits throughout the document.

### Revision C (March 2011)

Updated notes for clamping diodes, updated D080, D090, D121, D131 and D310. Also, updated the absolute maximum specification for the I/O pin and the maximum specification for the input/output clamp current. The 64-lead QFN packaging diagram was updated.

### Revision D (July 2011)

Updated the specification values in **Section 31.0 "Electrical Characteristics"**. Minor text edits throughout the document.

## APPENDIX B: MIGRATION FROM PIC18F85J90 AND PIC18F87J90 TO PIC18F87K90

Devices in the PIC18F87K90, PIC18F85J90 and PIC18F87J90 families are almost similar in their functions and features. Code can be migrated from the 18F85J90 to the PIC18F87K90 without many changes. The differences between the two device families are listed in Table B-1.