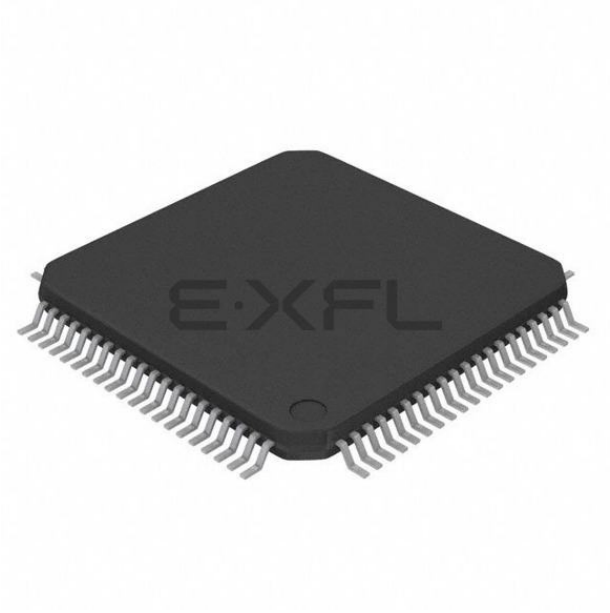


Welcome to E-XFL.COM

## 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 | 25MHz |
| Connectivity | EBI/EMI, I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 70 |
| Program Memory Size | 96KB (48K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 3.8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.2V ~ 5.5V |
| Data Converters | A/D 16x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-TQFP |
| Supplier Device Package | 80-TQFP (12x12) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f8627-e-pt |

## 1.3 Details on Individual Family Members

Devices in the PIC18F8722 family are available in 64-pin and 80-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2.

The devices are differentiated from each other in five ways:

1. Flash program memory (48 Kbytes for PIC18F6527/8527 devices, 64 Kbytes for PIC18F6622/8622 devices, 96 Kbytes for PIC18F6627/8627 devices and 128 Kbytes for PIC18F6722/8722).
2. A/D channels (12 for 64-pin devices, 16 for 80-pin devices).
3. I/O ports (7 bidirectional ports on 64-pin devices, 9 bidirectional ports on 80-pin devices).
4. External Memory Bus, configurable for 8 and 16-bit operation, is available on PIC18F8527/8622/8627/8722 devices.

All other features for devices in this family are identical. These are summarized in Table 1-2 and Table 1-2.

The pinouts for all devices are listed in Table 1-3 and Table 1-4.

Like all Microchip PIC18 devices, members of the PIC18F8722 family are available as both standard and low-voltage devices. Standard devices with Enhanced Flash memory, designated with an “F” in the part number (such as PIC18**F**6627), accommodate an operating $V_{DD}$ range of 4.2V to 5.5V. Low-voltage parts, designated by “LF” (such as PIC18**LF**6627), function over an extended $V_{DD}$ range of 2.0V to 5.5V.

**TABLE 1-1: DEVICE FEATURES (PIC18F6527/6622/6627/6722)**

| Features | PIC18F6527 | PIC18F6622 | PIC18F6627 | PIC18F6722 |
|---|---|---|---|---|
| Operating Frequency | DC – 40 MHz | DC – 40 MHz | DC – 40 MHz | DC – 40 MHz |
| Program Memory (Bytes) | 48K | 64K | 96K | 128K |
| Program Memory (Instructions) | 24576 | 32768 | 49152 | 65536 |
| Data Memory (Bytes) | 3936 | 3936 | 3936 | 3936 |
| Data EEPROM Memory (Bytes) | 1024 | 1024 | 1024 | 1024 |
| Interrupt Sources | 28 | 28 | 28 | 28 |
| I/O Ports | Ports A, B, C, D, E, F, G | Ports A, B, C, D, E, F, G | Ports A, B, C, D, E, F, G | Ports A, B, C, D, E, F, G |
| Timers | 5 | 5 | 5 | 5 |
| Capture/Compare/PWM Modules | 2 | 2 | 2 | 2 |
| Enhanced Capture/Compare/ PWM Modules | 3 | 3 | 3 | 3 |
| Enhanced USART | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, Enhanced USART | MSSP, Enhanced USART | MSSP, Enhanced USART | MSSP, Enhanced USART |
| Parallel Communications (PSP) | Yes | Yes | Yes | Yes |
| 10-bit Analog-to-Digital Module | 12 Input Channels | 12 Input Channels | 12 Input Channels | 12 Input Channels |
| Resets (and Delays) | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT |
| Programmable High/Low-Voltage Detect | Yes | Yes | Yes | Yes |
| Programmable Brown-out Reset | Yes | Yes | Yes | Yes |
| Instruction Set | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled |
| Packages | 64-pin TQFP | 64-pin TQFP | 64-pin TQFP | 64-pin TQFP |

## 4.2 Master Clear ($\overline{\text{MCLR}}$)

The $\overline{\text{MCLR}}$ pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path which detects and ignores small pulses.

The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

In the PIC18F8722 family of devices, the $\overline{\text{MCLR}}$ input can be disabled with the MCLRE Configuration bit. When $\overline{\text{MCLR}}$ is disabled, the pin becomes a digital input. See **Section 11.5 "PORTE, TRISE and LATE Registers"** for more information.

## 4.3 Power-on Reset (POR)

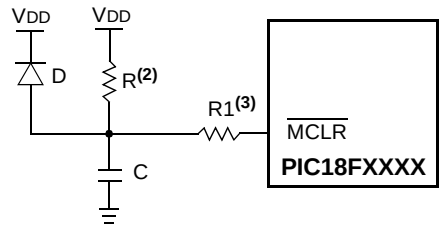
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the $\overline{\text{MCLR}}$ pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004, "Section **28.2 "DC Characteristics: Power-Down and Supply Current"**). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{\text{POR}}$ bit (RCON<1>). The state of the bit is set to '0' whenever a POR occurs; it does not change for any other Reset event. $\overline{\text{POR}}$ is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any POR.

**FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)(1)**




**Note 1:** External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

**2:** R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

**3:** R1 ≥ 1 kΩ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor C, in the event of $\overline{\text{MCLR}}$/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

# 5.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 Enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0 "Flash Program Memory"**. Data EEPROM is discussed separately in **Section 8.0 "Data EEPROM Memory"**.

## 5.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The PIC18F6527 and PIC18F8527 each have 48 Kbytes of Flash memory and can store up to 24,576 single-word instructions.

The PIC18F6622 and PIC18F8622 each have 64 Kbytes of Flash memory and can store up to 32,768 single-word instructions.

The PIC18F6627 and PIC18F8627 each have 96 Kbytes of Flash memory and can store up to 49,152 single-word instructions.

The PIC18F6722 and PIC18F8722 each have 128 Kbytes of Flash memory and can store up to 65,536 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory map for the PIC18F8722 family of devices is shown in Figure 5-1.

### 5.1.1 PIC18F8527/8622/8627/8722 PROGRAM MEMORY MODES

PIC18F8527/8622/8627/8722 devices differ significantly from their PIC18 predecessors in their utilization of program memory. In addition to available on-chip Flash program memory, these controllers can also address up to 2 Mbytes of external program memory through the external memory interface. There are four distinct operating modes available to the controllers:

- Microprocessor (MP)
- Microprocessor with Boot Block (MPBB)
- Extended Microcontroller (EMC)
- Microcontroller (MC)

The program memory mode is determined by setting the two Least Significant bits of the Configuration Register 3L (CONFIG3L) as shown in Register 25-4 (see **Section 25.1 "Configuration Bits"** for additional details on the device Configuration bits).

The program memory modes operate as follows:

- The **Microprocessor Mode** permits access only to external program memory; the contents of the on-chip Flash memory are ignored. The 21-bit program counter permits access to a 2-Mbyte linear program memory space.
- The **Microprocessor with Boot Block Mode** accesses on-chip Flash memory from the boot block. Above this, external program memory is accessed all the way up to the 2-Mbyte limit. Program execution automatically switches between the two memories as required. The boot block is configurable to 1, 2 or 4 Kbytes.
- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the physical limit of the on-chip Flash (0BFFFh for the PIC18F8527, 0FFFFh for the PIC18F8622, 17FFFh for the PIC18F8627, 1FFFFh for the PIC18F8722) causes a read of all '0's (a NOP instruction).
  The Microcontroller mode is also the only operating mode available to PIC18F6527/6622/6627/6722 devices.
- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip Flash memory; above this, the device accesses external program memory up to the 2-Mbyte program space limit. As with Boot Block mode, execution automatically switches between the two memories as required.

In all modes, the microcontroller has complete access to data RAM and EEPROM.

Figure 5-2 compares the memory maps of the different program memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 5-1.

## 5.3 Data Memory Organization

| | |
|---|---|
| **Note:** | The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 5.5 "Data Memory and the Extended Instruction Set"** for more information. |

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each; the PIC18F8722 family of devices implements all 16 banks. Figure 5-6 shows the data memory organization for the PIC18F8722 family of devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this subsection.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 5.3.2 "Access Bank"** provides a detailed description of the Access RAM.

### 5.3.1 BANK SELECT REGISTER (BSR)

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a 4-bit Bank Pointer.

Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the 4 Most Significant bits of a location's address; the instruction itself includes the 8 Least Significant bits. Only the four lower bits of the BSR are implemented (BSR<3:0>). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the MOVLB instruction.

The value of the BSR indicates the bank in data memory; the 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-7.

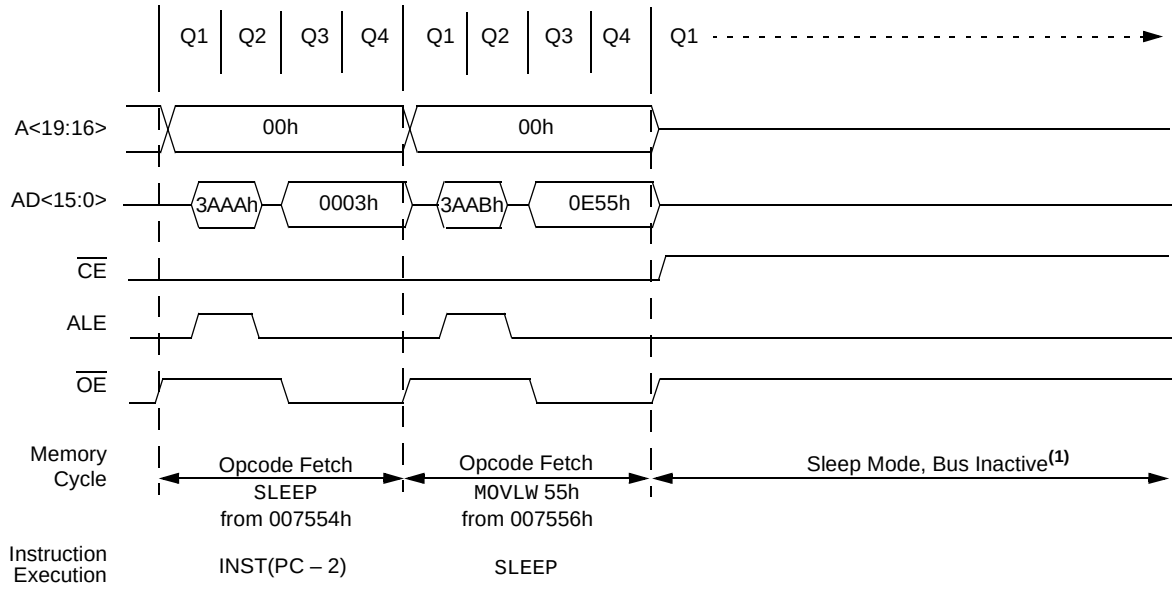
Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh will end up resetting the program counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 5-6 indicates which banks are implemented.

In the core PIC18 instruction set, only the MOVFF instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.


© 2008 Microchip Technology Inc.

**FIGURE 7-6: EXTERNAL MEMORY BUS TIMING FOR SLEEP (MICROPROCESSOR MODE)**




**Note 1:** Bus becomes inactive regardless of power-managed mode entered when SLEEP is executed.

**TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---|---|---|---|---|---|---|---|---|---|
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | 60 |
| LATB | LATB7 | LATB6 | LATB5 | LATB4 | LATB3 | LATB2 | LATB1 | LATB0 | 60 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 60 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 57 |
| INTCON2 | $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | INTEDG3 | TMR0IP | INT3IP | RBIP | 57 |
| INTCON3 | INT2IP | INT1IP | INT3IE | INT2IE | INT1IE | INT3IF | INT2IF | INT1IF | 57 |

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

**TABLE 11-9: PORTE FUNCTIONS**

| Pin Name | Function | TRIS Setting | I/O | I/O Type | Description |
|---|---|---|---|---|---|
| RE0/AD8/ $\overline{\text{RD}}$/P2D | RE0 | 0 | O | DIG | LATE<0> data output. |
| | | 1 | I | ST | PORTE<0> data input. |
| | AD8(2) | x | O | DIG | External memory interface, address/data bit 8 output. Takes priority over ECCP and port data. |
| | | x | I | TTL | External memory interface, data bit 8 input. |
| | $\overline{\text{RD}}$ | 1 | I | TTL | Parallel Slave Port read enable control input. |
| | P2D | 0 | O | DIG | ECCP2 Enhanced PWM output, channel D. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data. |
| RE1/AD9/ $\overline{\text{WR}}$/P2C | RE1 | 0 | O | DIG | LATE<1> data output. |
| | | 1 | I | ST | PORTE<1> data input. |
| | AD9(2) | x | O | DIG | External memory interface, address/data bit 9 output. Takes priority over ECCP and port data. |
| | | x | I | TTL | External memory interface, data bit 9 input. |
| | $\overline{\text{WR}}$ | 1 | I | TTL | Parallel Slave Port write enable control input. |
| | P2C | 0 | O | DIG | ECCP2 Enhanced PWM output, channel C. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data. |
| RE2/AD10/ $\overline{\text{CS}}$/P2B | RE2 | 0 | O | DIG | LATE<2> data output. |
| | | 1 | I | ST | PORTE<2> data input. |
| | AD10(2) | x | O | DIG | External memory interface, address/data bit 10 output. Takes priority over ECCP and port data. |
| | | x | I | TTL | External memory interface, data bit 10 input. |
| | $\overline{\text{CS}}$ | 1 | I | TTL | Parallel Slave Port chip select control input. |
| | P2B | 0 | O | DIG | ECCP2 Enhanced PWM output, channel B. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data. |
| RE3/AD11/P3C | RE3 | 0 | O | DIG | LATE<3> data output. |
| | | 1 | I | ST | PORTE<3> data input. |
| | AD11(2) | x | O | DIG | External memory interface, address/data bit 11 output. Takes priority over ECCP and port data. |
| | | x | I | TTL | External memory interface, data bit 11 input. |
| | P3C | 0 | O | DIG | ECCP3 Enhanced PWM output, channel C. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data. |
| RE4/AD12/P3B | RE4 | 0 | O | DIG | LATE<4> data output. |
| | | 1 | I | ST | PORTE<4> data input. |
| | AD12(2) | x | O | DIG | External memory interface, address/data bit 12 output. Takes priority over ECCP and port data. |
| | | x | I | TTL | External memory interface, data bit 12 input. |
| | P3B | 0 | O | DIG | ECCP3 Enhanced PWM output, channel B. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data. |

**Legend:** PWR = Power Supply, O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

**Note 1:** Alternate assignment for ECCP2 when CCP2MX Configuration bit is cleared (all devices in Microcontroller mode).

**2:** Implemented on 80-pin devices only.

#### 19.4.3.2 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and the SDAx line is held low ($\overline{ACK}$).

When the address byte overflow condition exists, then the no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPxSTAT<0>) is set, or bit SSPOV (SSPxCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPxIF, must be cleared in software. The SSPxSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPxCON2<0> = 1), SCLx will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPxCON1<4>). See **Section 19.4.4 “Clock Stretching”** for more detail.

#### 19.4.3.3 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin SCLx is held low regardless of SEN (see **Section 19.4.4 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then pin SCLx should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 19-9).

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not $\overline{ACK}$), then the data transfer is complete. In this case, when the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPxSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, pin SCLx must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared in software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.


© 2008 Microchip Technology Inc.

#### 19.4.17.2 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 low level to 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 19-29). 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 19-30).

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 19-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

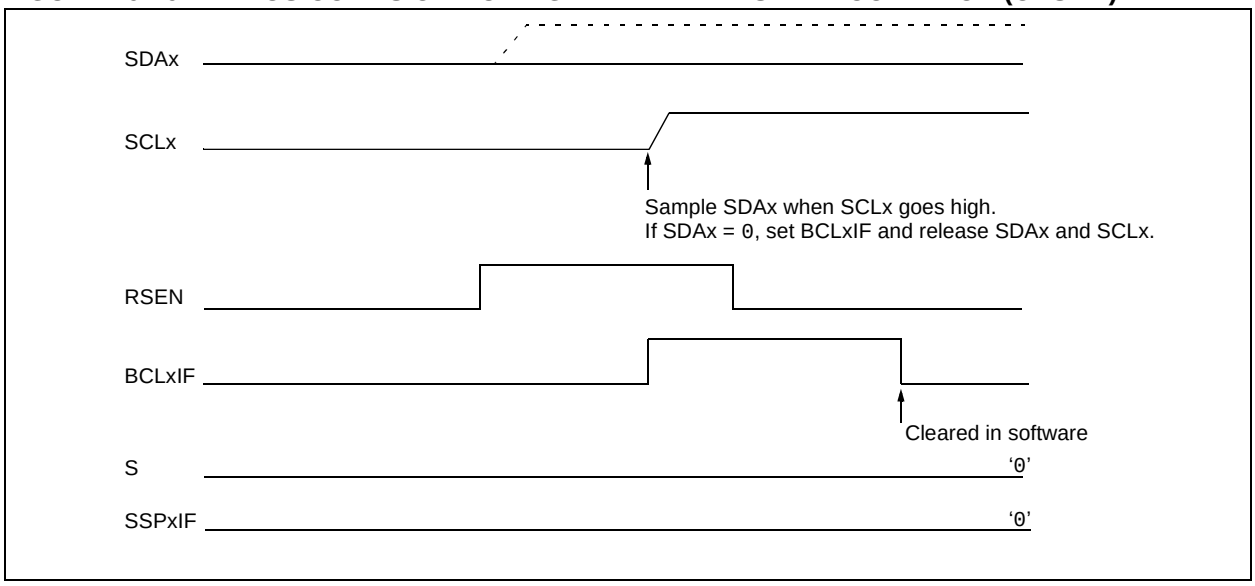
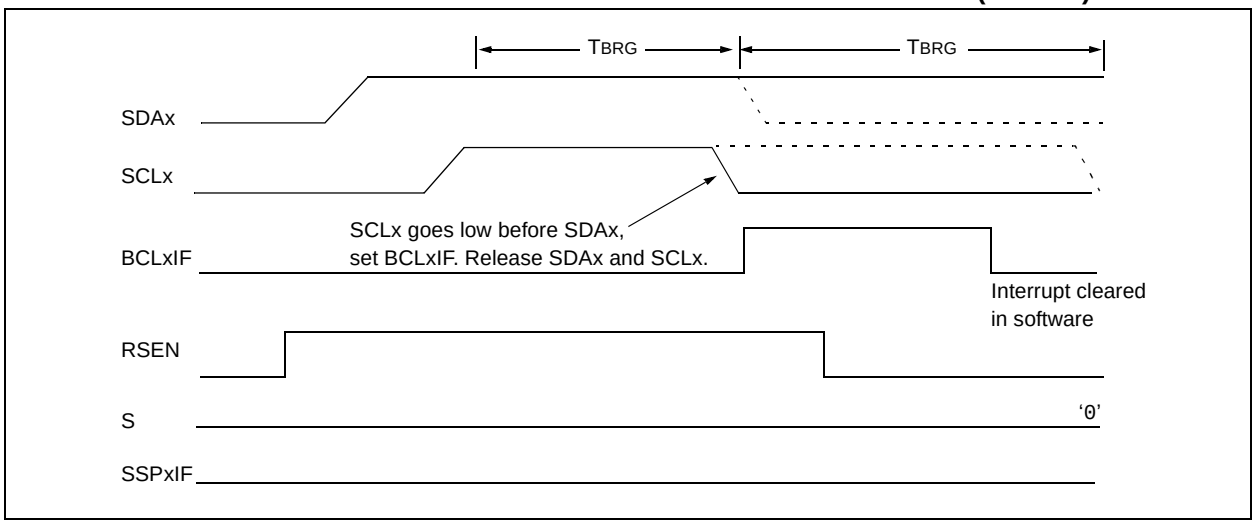



**FIGURE 19-30: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**



**NOTES:**


© 2008 Microchip Technology Inc.

### 20.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 20-10 for the timing of the Break character sequence.

#### 20.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

### 20.2.6 RECEIVING A BREAK CHARACTER

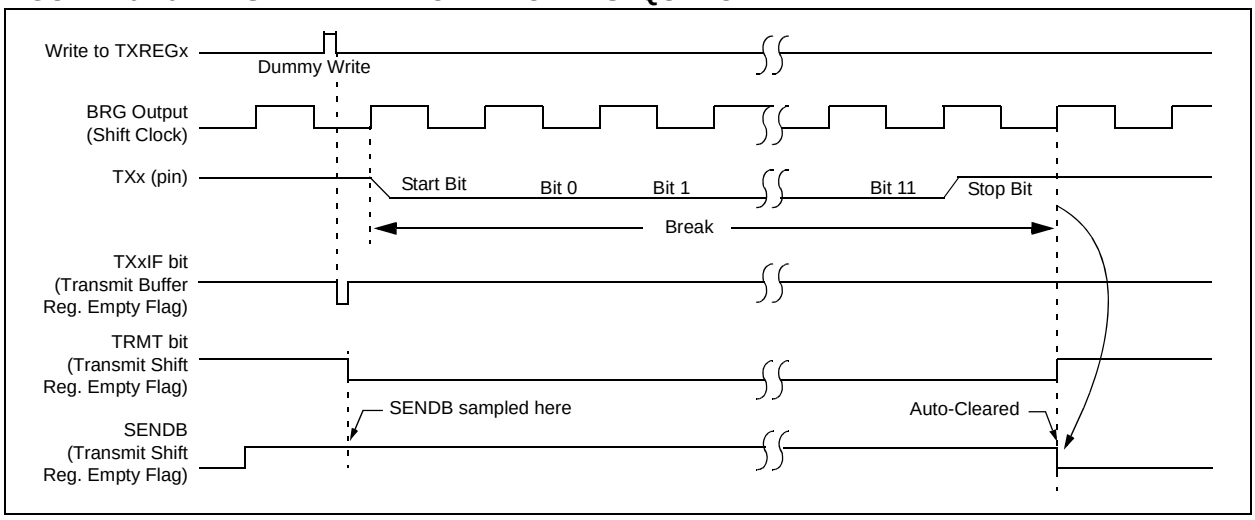
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 20.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABD bit once the TXxIF interrupt is observed.

**FIGURE 20-10: SEND BREAK CHARACTER SEQUENCE**



### REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| 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 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **VCFG<1:0>:** Voltage Reference Configuration bits

| | A/D VREF+ | A/D VREF- |
|---|---|---|
| 00 | AVDD | AVSS |
| 01 | External VREF+ | AVSS |
| 10 | AVDD | External VREF- |
| 11 | External VREF+ | External VREF- |

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits:

| PCFG<3:0> | AN15(1) | AN14(1) | AN13(1) | AN12(1) | AN11 | AN10 | AN9 | AN8 | AN7 | AN6 | AN5 | AN4 | AN3 | AN2 | AN1 | AN0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000 | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0001 | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0010 | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0011 | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0100 | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0101 | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0110 | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0111 | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 1000 | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 1001 | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 1010 | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 1011 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 1100 | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 1101 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 1110 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 1111 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |

A = Analog input D = Digital I/O

**Note 1:** AN15 through AN12 are available only on 80-pin devices.


© 2008 Microchip Technology Inc.

### REGISTER 25-1: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

| R/P-0 | R/P-0 | U-0 | U-0 | R/P-0 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| IESO | FCMEN | — | — | FOSC3 | FOSC2 | FOSC1 | FOSC0 |
| 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 **IESO:** Internal/External Oscillator Switchover bit
1 = Two-Speed Start-up enabled
0 = Two-Speed Start-up disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3-0 **FOSC<3:0>:** Oscillator Selection bits
11xx = External RC oscillator, CLKO function on RA6
101x = External RC oscillator, CLKO function on RA6
1001 = Internal oscillator block, CLKO function on RA6, port function on RA7
1000 = Internal oscillator block, port function on RA6 and RA7
0111 = External RC oscillator, port function on RA6
0110 = HS oscillator, PLL enabled (Clock Frequency = 4 x FOSC1)
0101 = EC oscillator, port function on RA6
0100 = EC oscillator, CLKO function on RA6
0011 = External RC oscillator, CLKO function on RA6
0010 = HS oscillator
0001 = XT oscillator
0000 = LP oscillator

| INCFSZ | Increment f, Skip if 0 |
|---|---|
| Syntax: | INCFSZ f {,d {,a}} |
| Operands: | 0 df d255<br>d • [0,1]<br>a • [0,1] |
| Operation: | (f) + 1 o dest,<br>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’. (default)<br>If the result is ‘0’, the next instruction which is already fetched is discarded and a NOPis executed instead, making it a two-cycle instruction.<br>If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank (default).<br>If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f d95 (5Fh). See Section 26.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details. |
| Words: | 1 |
| Cycles: | 1(2)<br>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 z 0;
PC = Address (NZERO)

| INFSNZ | Increment f, Skip if not 0 |
|---|---|
| Syntax: | INFSNZ f {,d {,a}} |
| Operands: | 0 df d255<br>d • [0,1]<br>a • [0,1] |
| Operation: | (f) + 1 o dest,<br>skip if result z0 |
| 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’ (default).<br>If the result is not ‘0’, the next instruction which is already fetched is discarded and a NOPis executed instead, making it a two-cycle instruction.<br>If ‘a’ is ‘0 |

| | |
|---|---|
| RRNCF | Rotate Right f (no carry) |
| Syntax: | RRNCF f {,d {,a}} |
| Operands: | 0 df d255<br>d • [0,1]<br>a • [0,1] |
| Operation: | (f<n>) o dest<n – 1>,<br>(f<0>) o dest<7> |
| Status Affected: | N, Z |
| Encoding: | 0100 00da ffff ffff |
| Description: | The contents of register ‘f’ are rotated one bit to the right. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed back in register ‘f’ (default).<br>If ‘a’ is ‘0’, the Access Bank will be |

## G


General Call Address Support ..... 229
GOTO ..... 342


## H


Hardware Multiplier ..... 117
    Introduction ..... 117
    Operation ..... 117
    Performance Comparison ..... 117
High/Low-Voltage Detect ..... 291
    Applications ..... 294
    Associated Registers ..... 295
    Characteristics ..... 395
    Current Consumption ..... 293
    Effects of a Reset ..... 295
    Operation ..... 292
        During Sleep ..... 295
    Setup ..... 293
    Start-up Time ..... 293
    Typical Application ..... 294
HLVD. See High/Low-Voltage Detect. ..... 291


## I


I/O Ports ..... 135
$I^2C$ Mode (MSSP)
    Acknowledge Sequence Timing ..... 239
    Associated Registers ..... 245
    Baud Rate Generator ..... 232
    Bus Collision
        During a Repeated Start Condition ..... 243
        During a Stop Condition ..... 244
    Clock Arbitration ..... 233
    Clock Stretching ..... 225
        10-Bit Slave Receive Mode (SEN = 1) ..... 225
        10-Bit Slave Transmit Mode ..... 225
        7-Bit Slave Receive Mode (SEN = 1) ..... 225
        7-Bit Slave Transmit Mode ..... 225
    Clock Synchronization and the CKP bit ..... 226
    Effects of a Reset ..... 240
    General Call Address Support ..... 229
    $I^2C$ Clock Rate w/BRG ..... 232
    Master Mode ..... 230
        Operation ..... 231
        Reception ..... 236
        Repeated Start Condition Timing ..... 235
        Start Condition Timing ..... 234
        Transmission ..... 236
    Multi-Master Communication, Bus Collision
        and Arbitration ..... 240
    Multi-Master Mode ..... 240
    Operation ..... 219
    Read/Write Bit Information (R/$\overline{W}$ Bit) ..... 219, 220
    Registers ..... 215
    Serial Clock (RC3/SCKx/SCLx) ..... 220
    Slave Mode ..... 219
        Addressing ..... 219
        Reception ..... 220
        Transmission ..... 220
    Sleep Operation ..... 240
    Stop Condition Timing ..... 239
ID Locations ..... 297, 320
INCF ..... 342
INCFSZ ..... 343
In-Circuit Debugger ..... 320
In-Circuit Serial Programming (ICSP) ..... 297, 320
Indexed Literal Offset Addressing
    and Standard PIC18 Instructions ..... 368
Indexed Literal Offset Mode ..... 368
Indirect Addressing ..... 82
INFSNZ ..... 343
Initialization Conditions for all Registers ..... 57–61
Instruction Cycle ..... 69
    Clocking Scheme ..... 69
Instruction Flow/Pipelining ..... 69
Instruction Set ..... 321
    ADDLW ..... 327
    ADDWF ..... 327
    ADDWF (Indexed Literal Offset Mode) ..... 369
    ADDWFC ..... 328
    ANDLW ..... 328
    ANDWF ..... 329
    BC ..... 329
    BCF ..... 330
    BN ..... 330
    BNC ..... 331
    BNN ..... 331
    BNOV ..... 332
    BNZ ..... 332
    BOV ..... 335
    BRA ..... 333
    BSF ..... 333
    BSF (Indexed Literal Offset Mode) ..... 369
    BTFSC ..... 334
    BTFSS ..... 334
    BTG ..... 335
    BZ ..... 336
    CALL ..... 336
    CLRF ..... 337
    CLRWDT ..... 337
    COMF ..... 338
    CPFSEQ ..... 338
    CPFSGT ..... 339
    CPFSLT ..... 339
    DAW ..... 340
    DCFSNZ ..... 341
    DECF ..... 340
    DECFSZ ..... 341
    Extended Instructions ..... 363
        Considerations when Enabling ..... 368
        Syntax ..... 363
        Use with MPLAB IDE Tools ..... 370
    General Format ..... 323
    GOTO ..... 342
    INCF ..... 342
    INCFSZ ..... 343
    INFSNZ ..... 343
    IORLW ..... 344
    IORWF ..... 344
    LFSR ..... 345
    MOVF ..... 345
    MOVFF ..... 346
    MOVLB ..... 346
    MOVLW ..... 347
    MOVWF ..... 347
    MULLW ..... 348
    MULWF ..... 348
    NEGF ..... 349
    NOP ..... 349
    POP ..... 350
    PUSH ..... 350



© 2008 Microchip Technology Inc.