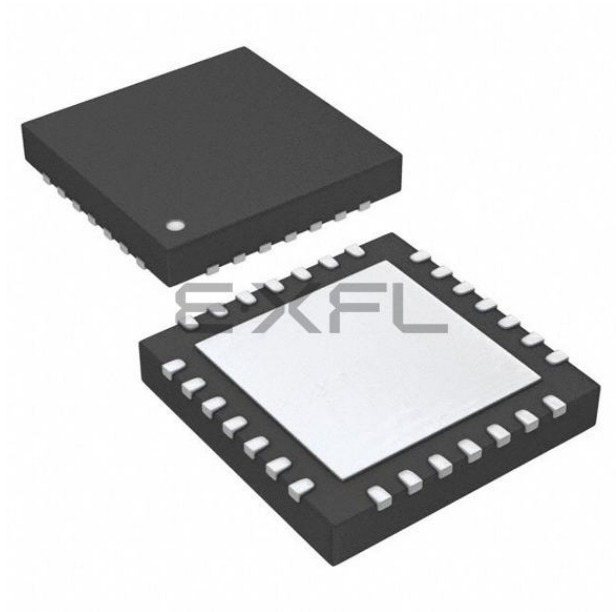


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

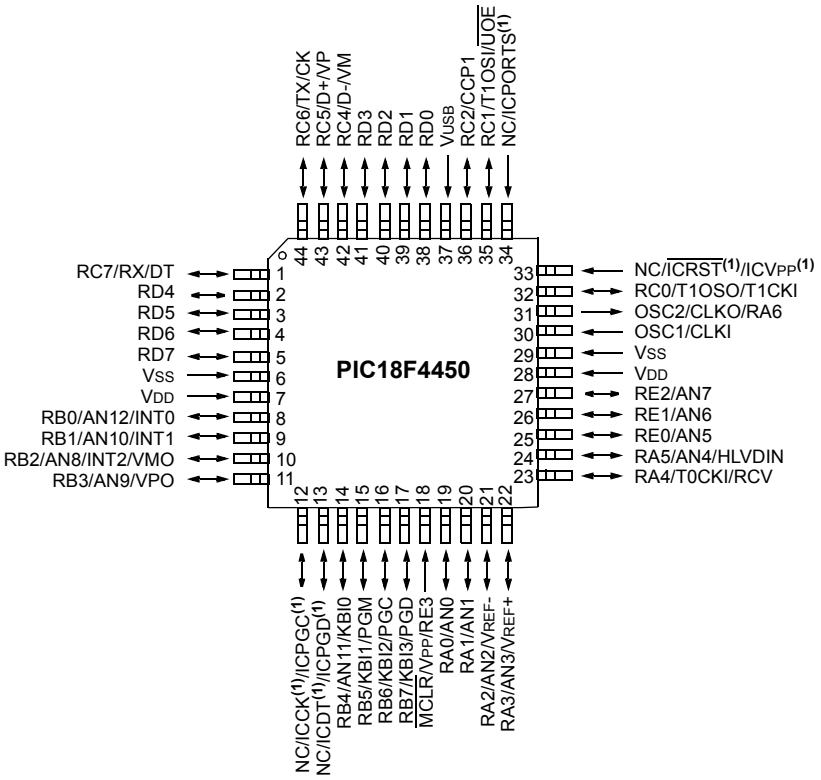
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 48MHz |
| Connectivity | UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 23 |
| Program Memory Size | 16KB (8K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 768 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 10x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-VQFN Exposed Pad |
| Supplier Device Package | 28-QFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf2450-i-ml |

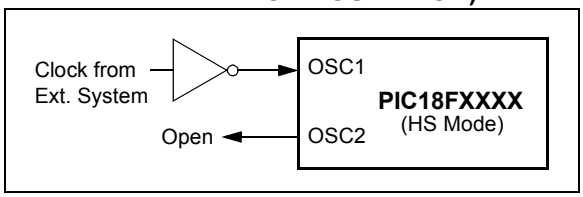
## Pin Diagrams (Continued)

**44-Pin TQFP**




**Note 1:** Special ICPORT features are available in select circumstances. For more information, see **Section 18.9 "Special ICPORT Features (Designated Packages Only)"**.

**FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)**

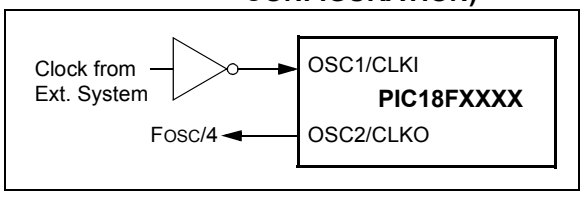



### 2.2.3 EXTERNAL CLOCK INPUT

The EC, ECIO, ECPLL and ECPIO Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

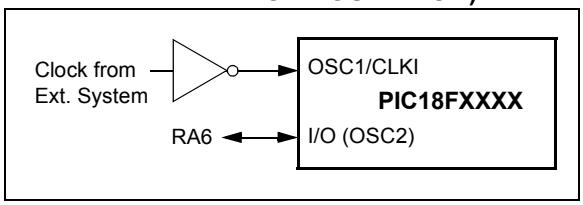
In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

**FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC AND ECPLL CONFIGURATION)**




The ECIO and ECPIO Oscillator modes function like the EC and ECPLL modes, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

**FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO AND ECPIO CONFIGURATION)**




The internal postscaler for reducing clock frequency in XT and HS modes is also available in EC and ECIO modes.

### 2.2.4 PLL FREQUENCY MULTIPLIER

PIC18F2450/4450 devices include a Phase Locked Loop (PLL) circuit. This is provided specifically for USB applications with lower speed oscillators and can also be used as a microcontroller clock source.

The PLL is enabled in HSPLL, XTPLL, ECPLL and ECPIO Oscillator modes. It is designed to produce a fixed 96 MHz reference clock from a fixed 4 MHz input. The output can then be divided and used for both the USB and the microcontroller core clock. Because the PLL has a fixed frequency input and output, there are eight prescaling options to match the oscillator input frequency to the PLL.

There is also a separate postscaler option for deriving the microcontroller clock from the PLL. This allows the USB peripheral and microcontroller to use the same oscillator input and still operate at different clock speeds. In contrast to the postscaler for XT, HS and EC modes, the available options are 1/2, 1/3, 1/4 and 1/6 of the PLL output.

The HSPLL, ECPLL and ECPIO modes make use of the HS mode oscillator for frequencies up to 48 MHz. The prescaler divides the oscillator input by up to 12 to produce the 4 MHz drive for the PLL. The XTPLL mode can only use an input frequency of 4 MHz which drives the PLL directly.

**FIGURE 2-6: PLL BLOCK DIAGRAM (HS MODE)**

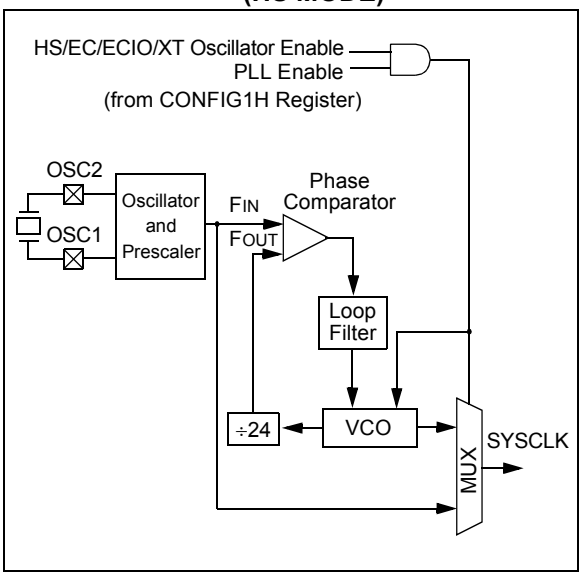


**TABLE 2-3: OSCILLATOR CONFIGURATION OPTIONS FOR USB OPERATION (CONTINUED)**

| Input Oscillator Frequency | PLL Division (PLLDIV2:PLLDIV0) | Clock Mode (FOSC3:FOSC0) | MCU Clock Division (CPUDIV1:CPUDIV0) | Microcontroller Clock Frequency |
|---|---|---|---|---|
| 12 MHz | ÷3 (010) | HS, EC, ECIO | None (00) | 12 MHz |
| | | | ÷2 (01) | 6 MHz |
| | | | ÷3 (10) | 4 MHz |
| | | | ÷4 (11) | 3 MHz |
| | | HSPLL, ECPLL, ECPIO | ÷2 (00) | 48 MHz |
| | | | ÷3 (01) | 32 MHz |
| | | | ÷4 (10) | **24 MHz** |
| | | | ÷6 (11) | 16 MHz |
| 8 MHz | ÷2 (001) | HS, EC, ECIO | None (00) | 8 MHz |
| | | | ÷2 (01) | 4 MHz |
| | | | ÷3 (10) | 2.67 MHz |
| | | | ÷4 (11) | 2 MHz |
| | | HSPLL, ECPLL, ECPIO | ÷2 (00) | 48 MHz |
| | | | ÷3 (01) | 32 MHz |
| | | | ÷4 (10) | **24 MHz** |
| | | | ÷6 (11) | 16 MHz |
| 4 MHz | ÷1 (000) | XT, HS, EC, ECIO | None (00) | 4 MHz |
| | | | ÷2 (01) | 2 MHz |
| | | | ÷3 (10) | 1.33 MHz |
| | | | ÷4 (11) | 1 MHz |
| | | HSPLL, ECPLL, XTPLL, ECPIO | ÷2 (00) | 48 MHz |
| | | | ÷3 (01) | 32 MHz |
| | | | ÷4 (10) | **24 MHz** |
| | | | ÷6 (11) | 16 MHz |

**Legend:** All clock frequencies, except 24 MHz, are exclusively associated with full-speed USB operation (USB clock of 48 MHz). **Bold** is used to highlight clock selections that are compatible with low-speed USB operation (system clock of 24 MHz, USB clock of 6 MHz).

**Note 1:** Only valid when the USBDIV Configuration bit is cleared.

## 3.0 POWER-MANAGED MODES

PIC18F2450/4450 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power-managed modes:

- Run modes
- Idle modes
- Sleep mode

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator); the Sleep mode does not use a clock source.

The power-managed modes include several power-saving features offered on previous PIC® microcontrollers. One is the clock switching feature, offered in other PIC18 devices, allowing the controller to use the Timer1 oscillator in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC microcontrollers, where all device clocks are stopped.

### 3.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions: if the CPU is to be clocked or not and the selection of a clock source. The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS1:SCS0 bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

#### 3.1.1 CLOCK SOURCES

The SCS1:SCS0 bits allow the selection of one of three clock sources for power-managed modes. They are:

- The primary clock, as defined by the FOSC3:FOSC0 Configuration bits
- The secondary clock (the Timer1 oscillator)
- The internal oscillator (for RC modes)

#### 3.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS1:SCS0 bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These are discussed in **Section 3.1.3 "Clock Transitions and Status Indicators"** and subsequent sections.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a `SLEEP` instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a `SLEEP` instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a `SLEEP` instruction to switch to the desired mode.

**TABLE 3-1: POWER-MANAGED MODES**

| Mode | OSCCON Bits | | Module Clocking | | Available Clock and Oscillator Source |
|---|---|---|---|---|---|
| | IDLEN(1) | SCS1:SCS0 | CPU | Peripherals | |
| Sleep | 0 | N/A | Off | Off | None – all clocks are disabled |
| PRI_RUN | N/A | 00 | Clocked | Clocked | Primary – all oscillator modes. This is the normal full-power execution mode. |
| SEC_RUN | N/A | 01 | Clocked | Clocked | Secondary – Timer1 oscillator |
| RC_RUN | N/A | 1x | Clocked | Clocked | Internal oscillator(2) |
| PRI_IDLE | 1 | 00 | Off | Clocked | Primary – all oscillator modes |
| SEC_IDLE | 1 | 01 | Off | Clocked | Secondary – Timer1 oscillator |
| RC_IDLE | 1 | 1x | Off | Clocked | Internal oscillator(2) |

**Note 1:** IDLEN reflects its value when the `SLEEP` instruction is executed.

**2:** Clock is INTRC source.

## 3.3 Sleep Mode

The power-managed Sleep mode in the PIC18F2450/4450 devices is identical to the legacy Sleep mode offered in all other PIC microcontrollers. It is entered by clearing the IDLEN bit (the default state on device Reset) and executing the SLEEP instruction. This shuts down the selected oscillator (Figure 3-5). All clock source status bits are cleared.

Entering the Sleep mode from any other mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source selected by the SCS1:SCS0 bits becomes ready (see Figure 3-6), or it will be clocked from the internal oscillator if either the Two-Speed Start-up or the Fail-Safe Clock Monitor are enabled (see **Section 18.0 "Special Features of the CPU"**). In either case, the OSTS bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

## 3.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

If the IDLEN bit is set to '1' when a SLEEP instruction is executed, the peripherals will be clocked from the clock source selected using the SCS1:SCS0 bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a SLEEP instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of TCSD (parameter 38, Table 21-10) while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC_IDLE mode, the internal oscillator will clock the CPU and peripherals (in other words, RC_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS1:SCS0 bits.

**FIGURE 3-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE**

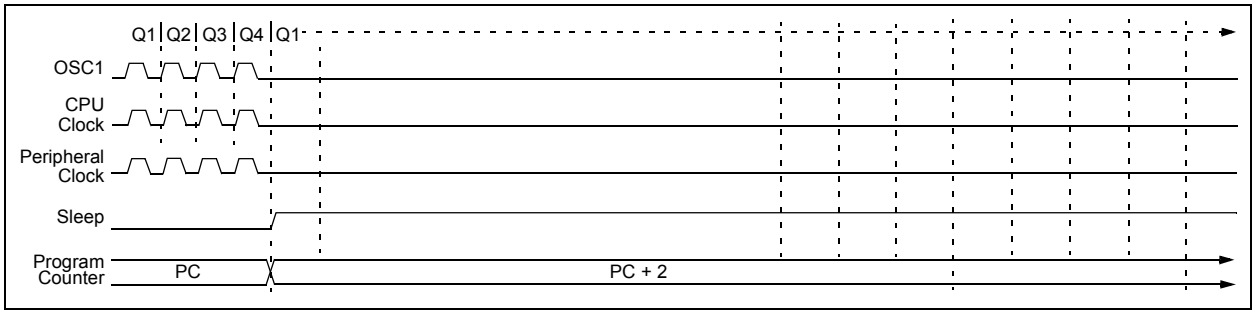
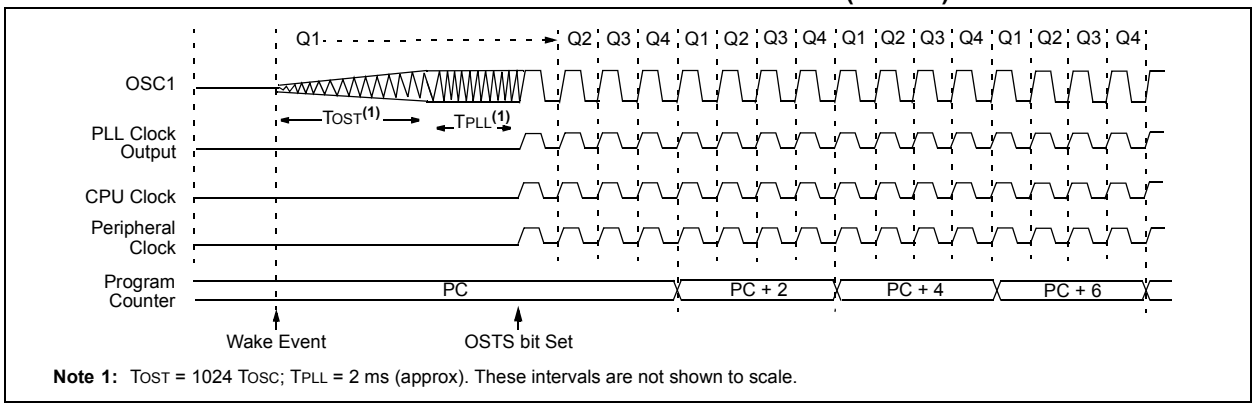



**FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)**




**Note 1:** TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

## 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 5.6 "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. PIC18F2450/4450 devices implement three complete banks, for a total of 768 bytes. Figure 5-5 shows the data memory organization for the 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 5.3.3 "Access Bank"** provides a detailed description of the Access RAM.

### 5.3.1 USB RAM

Bank 4 of the data memory is actually mapped to special dual port RAM. When the USB module is disabled, the GPRs in these banks are used like any other GPR in the data memory space.

When the USB module is enabled, the memory in this bank is allocated as buffer RAM for USB operation. This area is shared between the microcontroller core and the USB Serial Interface Engine (SIE) and is used to transfer data directly between the two.

It is theoretically possible to use this area of USB RAM that is not allocated as USB buffers for normal scratchpad memory or other variable storage. In practice, the dynamic nature of buffer allocation makes this risky at best. Bank 4 is also used for USB buffer management when the module is enabled and should not be used for any other purposes during that time.

Additional information on USB RAM and buffer operation is provided in **Section 14.0 "Universal Serial Bus (USB)"**.

### 5.3.2 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 (BSR3:BSR0). 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.

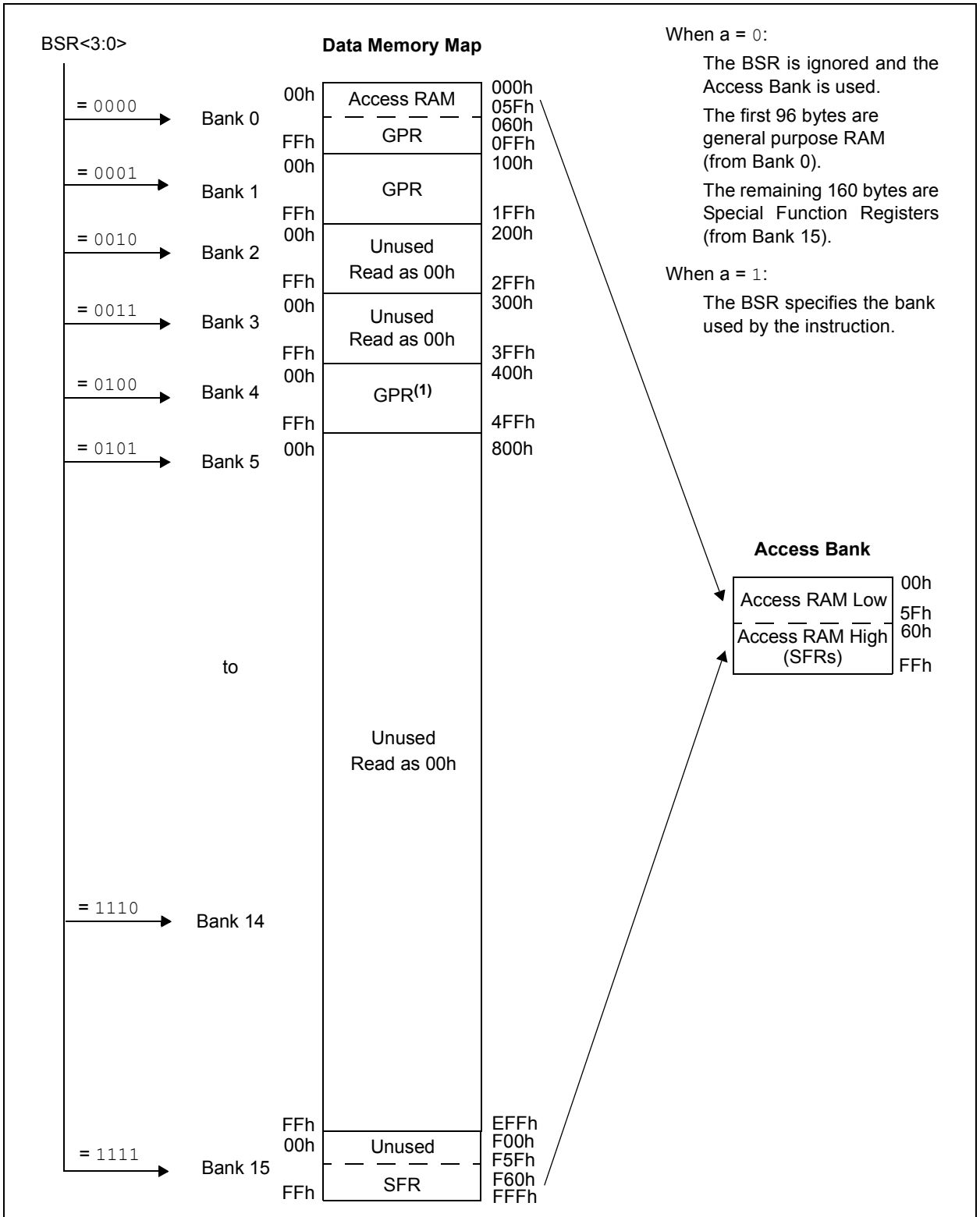
The value of the BSR indicates the bank in data memory. The eight bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-6.

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.

**FIGURE 5-5: DATA MEMORY MAP FOR PIC18F2450/4450 DEVICES**




**Note 1:** This bank also serve as RAM buffer for USB operation. See **Section 5.3.1 "USB RAM"** for more information.

### 5.3.5 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM in the data memory space. SFRs start at the top of data memory and extend downward to occupy the top segment of Bank 15, from F60h to FFFh. A list of these registers is given in Table 5-1 and Table 5-2.

The SFRs can be classified into two sets: those associated with the "core" device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and interrupt registers are described in their respective chapters, while the ALU's STATUS register is described later in this section. Registers related to the operation of a peripheral feature are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as '0's.

**TABLE 5-1: SPECIAL FUNCTION REGISTER MAP FOR PIC18F2450/4450 DEVICES**

| Address | Name | Address | Name | Address | Name | Address | Name | Address | Name |
|---|---|---|---|---|---|---|---|---|---|
| FFFh | TOSU | FDFh | INDF2(1) | FBFh | CCPR1H | F9Fh | IPR1 | F7Fh | UEP15 |
| FFEh | TOSH | FDEh | POSTINC2(1) | FBEh | CCPR1L | F9Eh | PIR1 | F7Eh | UEP14 |
| FFDh | TOSL | FDDh | POSTDEC2(1) | FBDh | CCP1CON | F9Dh | PIE1 | F7Dh | UEP13 |
| FFCh | STKPTR | FDCh | PREINC2(1) | FBCh | —(2) | F9Ch | —(2) | F7Ch | UEP12 |
| FFBh | PCLATU | FDBh | PLUSW2(1) | FBBh | —(2) | F9Bh | —(2) | F7Bh | UEP11 |
| FFAh | PCLATH | FDAh | FSR2H | FBAh | —(2) | F9Ah | —(2) | F7Ah | UEP10 |
| FF9h | PCL | FD9h | FSR2L | FB9h | —(2) | F99h | —(2) | F79h | UEP9 |
| FF8h | TBLPTRU | FD8h | STATUS | FB8h | BAUDCON | F98h | —(2) | F78h | UEP8 |
| FF7h | TBLPTRH | FD7h | TMR0H | FB7h | —(2) | F97h | —(2) | F77h | UEP7 |
| FF6h | TBLPTRL | FD6h | TMR0L | FB6h | —(2) | F96h | TRISE(3) | F76h | UEP6 |
| FF5h | TABLAT | FD5h | T0CON | FB5h | —(2) | F95h | TRISD(3) | F75h | UEP5 |
| FF4h | PRODH | FD4h | —(2) | FB4h | —(2) | F94h | TRISC | F74h | UEP4 |
| FF3h | PRODL | FD3h | OSCCON | FB3h | —(2) | F93h | TRISB | F73h | UEP3 |
| FF2h | INTCON | FD2h | HLVDCON | FB2h | —(2) | F92h | TRISA | F72h | UEP2 |
| FF1h | INTCON2 | FD1h | WDTCON | FB1h | —(2) | F91h | —(2) | F71h | UEP1 |
| FF0h | INTCON3 | FD0h | RCON | FB0h | SPBRGH | F90h | —(2) | F70h | UEP0 |
| FEFh | INDF0(1) | FCFh | TMR1H | FAFh | SPBRG | F8Fh | —(2) | F6Fh | UCFG |
| FEEh | POSTINC0(1) | FCEh | TMR1L | FAEh | RCREG | F8Eh | —(2) | F6Eh | UADDR |
| FEDh | POSTDEC0(1) | FCDh | T1CON | FADh | TXREG | F8Dh | LATE(3) | F6Dh | UCON |
| FECh | PREINC0(1) | FCCh | TMR2 | FACh | TXSTA | F8Ch | LATD(3) | F6Ch | USTAT |
| FEBh | PLUSW0(1) | FCBh | PR2 | FABh | RCSTA | F8Bh | LATC | F6Bh | UEIE |
| FEAh | FSR0H | FCAh | T2CON | FAAh | —(2) | F8Ah | LATB | F6Ah | UEIR |
| FE9h | FSR0L | FC9h | —(2) | FA9h | —(2) | F89h | LATA | F69h | UIE |
| FE8h | WREG | FC8h | —(2) | FA8h | —(2) | F88h | —(2) | F68h | UIR |
| FE7h | INDF1(1) | FC7h | —(2) | FA7h | EECON2(1) | F87h | —(2) | F67h | UFRMH |
| FE6h | POSTINC1(1) | FC6h | —(2) | FA6h | EECON1 | F86h | —(2) | F66h | UFRML |
| FE5h | POSTDEC1(1) | FC5h | —(2) | FA5h | —(2) | F85h | —(2) | F65h | —(2) |
| FE4h | PREINC1(1) | FC4h | ADRESH | FA4h | —(2) | F84h | PORTE | F64h | —(2) |
| FE3h | PLUSW1(1) | FC3h | ADRESL | FA3h | —(2) | F83h | PORTD(3) | F63h | —(2) |
| FE2h | FSR1H | FC2h | ADCON0 | FA2h | IPR2 | F82h | PORTC | F62h | —(2) |
| FE1h | FSR1L | FC1h | ADCON1 | FA1h | PIR2 | F81h | PORTB | F61h | —(2) |
| FE0h | BSR | FC0h | ADCON2 | FA0h | PIE2 | F80h | PORTA | F60h | —(2) |

**Note 1:** Not a physical register.
**2:** Unimplemented registers are read as '0'.
**3:** These registers are implemented only on 40/44-pin devices.

### 5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

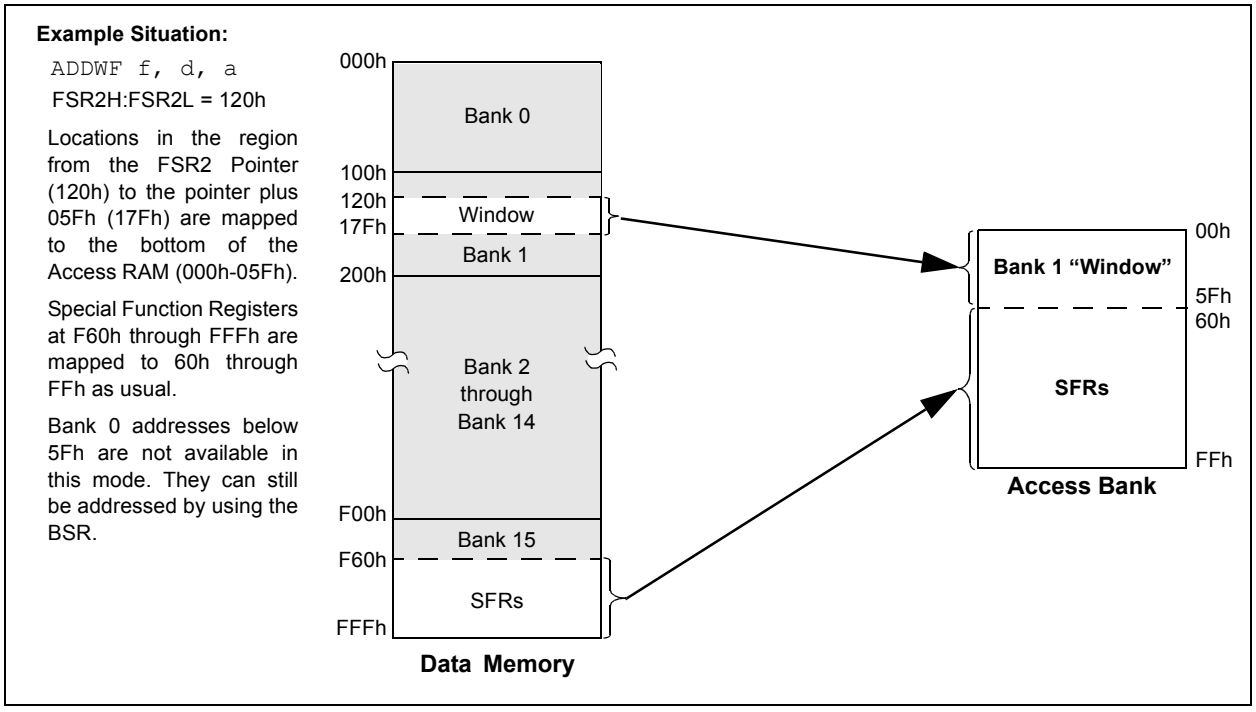
The use of Indexed Literal Offset Addressing mode effectively changes how the lower portion of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom half of Bank 0, this mode maps the contents from Bank 0 and a user-defined "window" that can be located anywhere in the data memory space. The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described (see **Section 5.3.3 "Access Bank"**). An example of Access Bank remapping in this addressing mode is shown in Figure 5-9.

Remapping of the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit is '1') will continue to use Direct Addressing as before. Any indirect or indexed operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

### 5.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

**FIGURE 5-9: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING**



**REGISTER 6-1: EECON1: MEMORY CONTROL REGISTER 1**

| U-0 | R/W-x | U-0 | R/W-0 | R/W-x | R/W-0 | R/S-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | CFGS | — | FREE | WRERR(1) | WREN | WR | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | S = Settable bit | | |
|---|---|---|---|
| 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 **Unimplemented:** Read as '0'

bit 6 **CFGS:** Flash Program or Configuration Select bit
- 1 = Access Configuration registers
- 0 = Access Flash program

bit 5 **Unimplemented:** Read as '0'

bit 4 **FREE:** Flash Row Erase Enable bit
- 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
- 0 = Perform write-only

bit 3 **WRERR:** Flash Program Error Flag bit(1)
- 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation or an improper write attempt)
- 0 = The write operation completed

bit 2 **WREN:** Flash Program Write Enable bit
- 1 = Allows write cycles to Flash program
- 0 = Inhibits write cycles to Flash program

bit 1 **WR:** Write Control bit
- 1 = Initiates a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle complete

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

**Note 1:** When a WRERR occurs, the CFGS bit is not cleared. This allows tracing of the error condition.

**REGISTER 8-2: INTCON2: INTERRUPT CONTROL REGISTER 2**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1 | U-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RBIP |
| 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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

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

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as '0'

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

**Note:** Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

**REGISTER 15-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN(1) | SYNC | SENDB | BRGH | TRMT | TX9D |
| 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 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care.

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-Bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)

1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit

Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed

Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode with the exception that SREN has no effect in Synchronous Slave mode.

## 15.1 Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit (BAUDCON<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free-running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 (BAUDCON<3>) also control the baud rate. In Synchronous mode, BRGH is ignored. Table 15-1 shows the formula for computation of the baud rate for different EUSART modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGH:SPBRG registers can be calculated using the formulas in Table 15-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 15-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 15-2. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGH:SPBRG registers causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 15.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRG register pair.

### 15.1.2 SAMPLING

The data on the RX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 15-1: BAUD RATE FORMULAS**

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|---|---|---|---|---|
| SYNC | BRG16 | BRGH | | |
| 0 | 0 | 0 | 8-Bit/Asynchronous | FOSC/[64 (n + 1)] |
| 0 | 0 | 1 | 8-Bit/Asynchronous | FOSC/[16 (n + 1)] |
| 0 | 1 | 0 | 16-Bit/Asynchronous | |
| 0 | 1 | 1 | 16-Bit/Asynchronous | FOSC/[4 (n + 1)] |
| 1 | 0 | x | 8-Bit/Synchronous | |
| 1 | 1 | x | 16-Bit/Synchronous | |

**Legend:** x = Don't care, n = Value of SPBRGH:SPBRG register pair

**REGISTER 18-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)**

| U-0 | U-0 | R/P-0 | R/P-0 | R/P-0 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| — | — | USBDIV | CPUDIV1 | CPUDIV0 | PLLDIV2 | PLLDIV1 | PLLDIV0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as '0' |
| -n = Value when device is unprogrammed | | u = Unchanged from programmed state |

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **USBDIV:** USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1)
1 = USB clock source comes from the 96 MHz PLL divided by 2
0 = USB clock source comes directly from the primary oscillator block with no postscale

bit 4-3 **CPUDIV1:CPUDIV0:** System Clock Postscaler Selection bits
For XT, HS, EC and ECIO Oscillator modes:
11 = Primary oscillator divided by 4 to derive system clock
10 = Primary oscillator divided by 3 to derive system clock
01 = Primary oscillator divided by 2 to derive system clock
00 = Primary oscillator used directly for system clock (no postscaler)
For XTPLL, HSPLL, ECPLL and ECPIO Oscillator modes:
11 = 96 MHz PLL divided by 6 to derive system clock
10 = 96 MHz PLL divided by 4 to derive system clock
01 = 96 MHz PLL divided by 3 to derive system clock
00 = 96 MHz PLL divided by 2 to derive system clock

bit 2-0 **PLLDIV2:PLLDIV0:** PLL Prescaler Selection bits
111 = Divide by 12 (48 MHz oscillator input)
110 = Divide by 10 (40 MHz oscillator input)
101 = Divide by 6 (24 MHz oscillator input)
100 = Divide by 5 (20 MHz oscillator input)
011 = Divide by 4 (16 MHz oscillator input)
010 = Divide by 3 (12 MHz oscillator input)
001 = Divide by 2 (8 MHz oscillator input)
000 = No prescale (4 MHz oscillator input drives PLL directly)

**ANDWF** **AND W with f**

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

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

| 0001 | 01da | ffff | ffff |
|---|---|---|---|

Description: The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ANDWF REG, 0, 0`

Before Instruction
W = 17h
REG = C2h
After Instruction
W = 02h
REG = C2h

**BC** **Branch if Carry**

Syntax: BC n

Operands: $-128 \le n \le 127$

Operation: if Carry bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

| 1110 | 0010 | nnnn | nnnn |
|---|---|---|---|

Description: If the Carry bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'n' | Process Data | No operation |

Example: `HERE BC 5`

Before Instruction
PC = address `(HERE)`
After Instruction
If Carry = 1;
PC = address `(HERE + 12)`
If Carry = 0;
PC = address `(HERE + 2)`

| XORWF | Exclusive OR W with f |
|---|---|
| Syntax: | XORWF f {,d {,a}} |
| Operands: | $0 \le f \le 255$<br>$d \in [0,1]$<br>$a \in [0,1]$ |
| Operation: | (W) .XOR. (f) $\rightarrow$ dest |
| Status Affected: | N, Z |

Encoding:

| 0001 | 10da | ffff | ffff |
|---|---|---|---|

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 19.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:

```
XORWF   REG, 1, 0
```

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

## 21.2 DC Characteristics: Power-Down and Supply Current PIC18F2450/4450 (Industrial) PIC18LF2450/4450 (Industrial) (Continued)

| PIC18LF2450/4450 (Industrial) | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18F2450/4450** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD)(2)** | | | | | | |
| | PIC18LF2450/4450 | 10 | 32 | μA | -40°C | VDD = 2.0V | FOSC = 31 kHz (**RC_RUN** mode, INTRC source) |
| | | 10 | 30 | μA | +25°C | | |
| | | 12 | 29 | μA | +85°C | | |
| | PIC18LF2450/4450 | 35 | 63 | μA | -40°C | VDD = 3.0V | |
| | | 30 | 60 | μA | +25°C | | |
| | | 25 | 57 | μA | +85°C | | |
| | All devices | 95 | 168 | μA | -40°C | VDD = 5.0V | |
| | | 75 | 160 | μA | +25°C | | |
| | | 65 | 152 | μA | +85°C | | |
| | PIC18LF2450/4450 | 2.3 | 8 | μA | -40°C | VDD = 2.0V | FOSC = 31 kHz (**RC_IDLE** mode, INTRC source) |
| | | 2.5 | 8 | μA | +25°C | | |
| | | 3.3 | 11 | μA | +85°C | | |
| | PIC18LF2450/4450 | 3.3 | 11 | μA | -40°C | VDD = 3.0V | |
| | | 3.6 | 11 | μA | +25°C | | |
| | | 4.0 | 15 | μA | +85°C | | |
| | All devices | 6.5 | 16 | μA | -40°C | VDD = 5.0V | |
| | | 7.0 | 16 | μA | +25°C | | |
| | | 9.0 | 36 | μA | +85°C | | |

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

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD or VSS;
MCLR = VDD; WDT enabled/disabled as specified.

**3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

**4:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

## 21.2 DC Characteristics: Power-Down and Supply Current PIC18F2450/4450 (Industrial) PIC18LF2450/4450 (Industrial) (Continued)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18LF2450/4450** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **PIC18F2450/4450** (Industrial) | | **Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **USB and Related Module Differential Currents (ΔIUSBX, ΔIPLL, ΔIUREG)** | | | | | | |
| ΔIUSBX | **USB Module with On-Chip Transceiver** | 8.0 | 14.5 | mA | +25°C | VDD = 3.3V | |
| | | 12.4 | 20 | mA | +25°C | VDD = 5.0V | |
| ΔIPLL | **96 MHz PLL (Oscillator Module)** | 1.2 | 3.0 | mA | +25°C | VDD = 3.3V | |
| | | 1.2 | 4.8 | mA | +25°C | VDD = 5.0V | |
| ΔIUREG | **USB Internal Voltage Regulator** | 80 | 125 | μA | +25°C | VDD = 5.0V | USB Idle, UCON<SUSPND> = 1 |

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

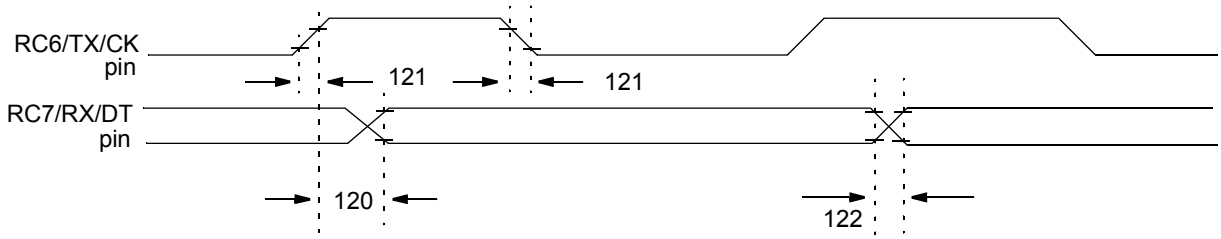
**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD or VSS;
MCLR = VDD; WDT enabled/disabled as specified.

**3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

**4:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

**FIGURE 21-11: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

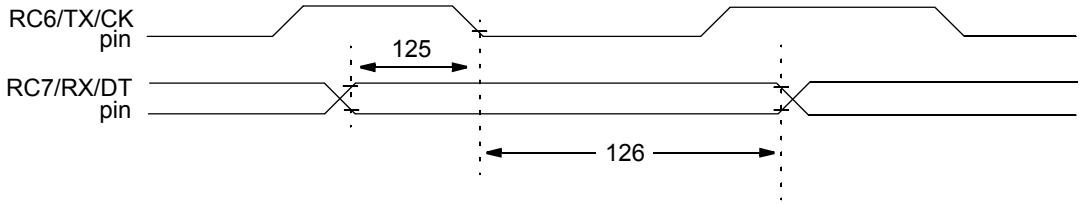



**Note:** Refer to Figure 21-4 for load conditions.

**TABLE 21-13: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Param No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid | PIC18**F**XXXX | — | 40 | ns | |
| | | | PIC18**LF**XXXX | — | 100 | ns | VDD = 2.0V |
| 121 | Tckrf | Clock Out Rise Time and Fall Time (Master mode) | PIC18**F**XXXX | — | 20 | ns | |
| | | | PIC18**LF**XXXX | — | 50 | ns | VDD = 2.0V |
| 122 | Tdtrf | Data Out Rise Time and Fall Time | PIC18**F**XXXX | — | 20 | ns | |
| | | | PIC18**LF**XXXX | — | 50 | ns | VDD = 2.0V |

**FIGURE 21-12: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




**Note:** Refer to Figure 21-4 for load conditions.

**TABLE 21-14: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|
| 125 | TDTV2CKL | SYNC RCV (MASTER & SLAVE) Data Hold before CK ↓ (DT hold time) | 10 | — | ns | |
| 126 | TCKL2DTL | Data Hold after CK ↓ (DT hold time) | 15 | — | ns | |

## 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 | PIC18F2450(1), PIC18F4450(1), PIC18F2450T(2), PIC18F4450T(2); VDD range 4.2V to 5.5V<br>PIC18LF2450(1), PIC18LF4450(1), PIC18LF2450T(2), PIC18LF4450T(2); VDD range 2.0V to 5.5V | |
| Temperature Range | I = -40°C to +85°C (Industrial)<br>E = -40°C to +125°C (Extended) | |
| Package | PT = TQFP (Thin Quad Flatpack)<br>SO = SOIC<br>SP = Skinny Plastic DIP<br>P = PDIP<br>ML = QFN | |
| Pattern | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC18LF4450-I/P 301 = Industrial temp., PDIP package, Extended VDD limits, QTP pattern #301.

b) PIC18LF2450-I/SO = Industrial temp., SOIC package, Extended VDD limits.

c) PIC18F4450-I/P = Industrial temp., PDIP package, normal VDD limits.

**Note 1:** F = Standard Voltage Range
LF = Wide Voltage Range

**2:** T = in tape and reel TQFP packages only.