

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	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f43k22-e-pt

PIC18(L)F2X/4XK22

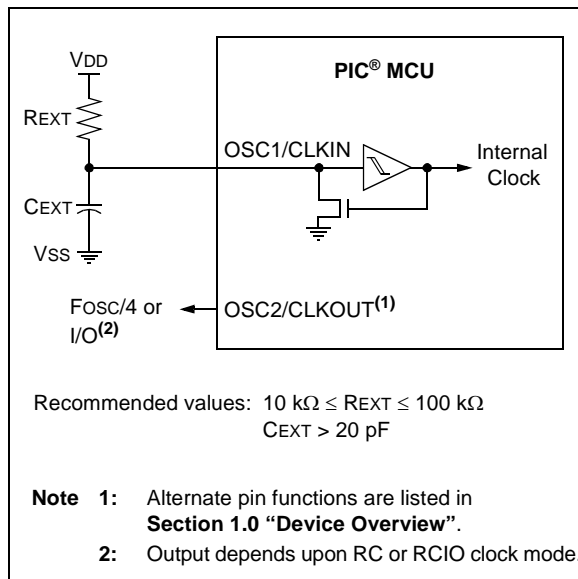
2.5.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

2.5.4.1 RC Mode

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by four. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 2-8 shows the external RC mode connections.

FIGURE 2-8: EXTERNAL RC MODES



2.5.4.2 RCIO Mode

In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes a general purpose I/O pin.

The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}) values and the operating temperature. Other factors affecting the oscillator frequency are:

- input threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

2.6 Internal Clock Modes

The oscillator module has three independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 2-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory calibrated and operates at 31.25 kHz. The LFINTOSC cannot be user-adjusted, but is designed to be stable over temperature and voltage.

The system clock speed can be selected via software using the Internal Oscillator Frequency select bits $IRCF<2:0>$ of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection ($SCS<1:0>$) bits of the OSCCON register. See Section 2.11 “Clock Switching” for more information.

2.6.1 INTOSC WITH I/O OR CLOCKOUT

Two of the clock modes selectable with the $FOSC<3:0>$ bits of the CONFIG1H Configuration register configure the internal oscillator block as the primary oscillator. Mode selection determines whether the OSC2/CLKOUT pin will be configured as general purpose I/O or $FOSC/4$ (CLKOUT). In both modes, the OSC1/CLKIN pin is configured as general purpose I/O. See Section 24.0 “Special Features of the CPU” for more information.

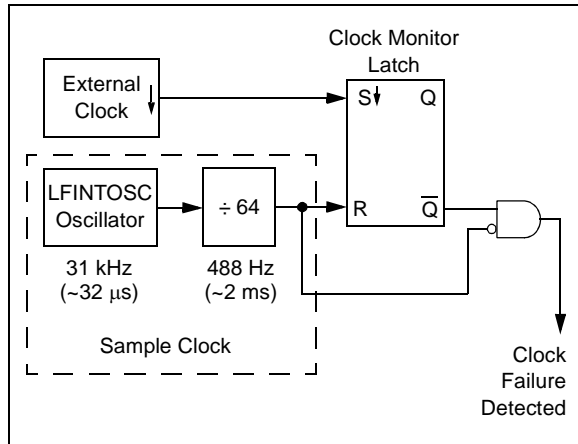
The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

PIC18(L)F2X/4XK22

2.13 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the CONFIG1H Configuration register. The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC, RC and RCIO).

FIGURE 2-10: FSCM BLOCK DIAGRAM



2.13.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64 (see Figure 2-10). Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

2.13.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSCFIF of the PIR2 register. The OSCFIF flag will generate an interrupt if the OSCFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation. An automatic transition back to the failed clock source will not occur.

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

2.13.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared by either one of the following:

- Any Reset
- By toggling the SCS1 bit of the OSCCON register

Both of these conditions restart the OST. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device automatically switches over to the external clock source. The Fail-Safe condition need not be cleared before the OSCFIF flag is cleared.

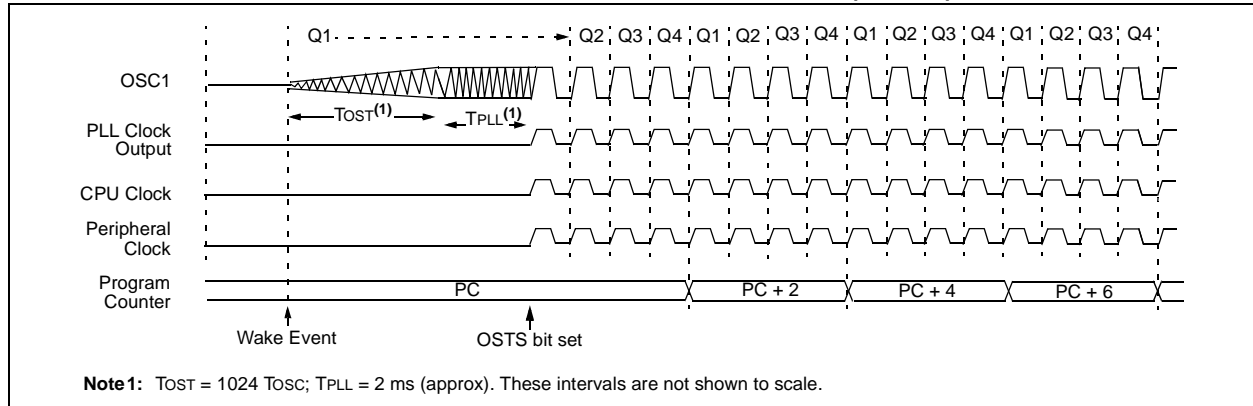
2.13.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit of the OSCCON register to verify the oscillator start-up and that the system clock switchover has successfully completed.

Note: When the device is configured for Fail-Safe clock monitoring in either HS, XT, or LS Oscillator modes then the IESO configuration bit should also be set so that the clock will automatically switch from the internal clock to the external oscillator when the OST times out.

FIGURE 3-5: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



3.4.1 PRI_IDLE MODE

This mode is unique among the three low-power Idle modes, in that it does not disable the primary device clock. For timing sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to “warm-up” or transition from another oscillator.

PRI_IDLE mode is entered from PRI_RUN mode by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set IDLEN first, then clear the SCS bits and execute SLEEP. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC<3:0> Configuration bits. The OST bit remains set (see Figure 3-6).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of interval T_{CSD} is required between the wake event and when code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OST bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 3-7).

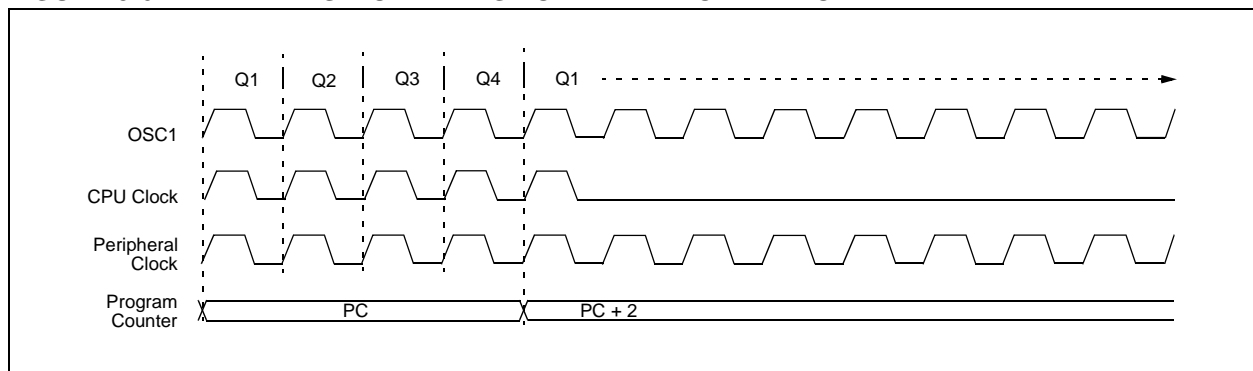
3.4.2 SEC_IDLE MODE

In SEC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the SOSC oscillator. This mode is entered from SEC_RUN by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set the IDLEN bit first, then set the SCS<1:0> bits to ‘01’ and execute SLEEP. When the clock source is switched to the SOSC oscillator, the primary oscillator is shut down, the OST bit is cleared and the SOSCRUN bit is set.

When a wake event occurs, the peripherals continue to be clocked from the SOSC oscillator. After an interval of T_{CSD} following the wake event, the CPU begins executing code being clocked by the SOSC oscillator. The IDLEN and SCS bits are not affected by the wake-up; the SOSC oscillator continues to run (see Figure 3-7).

Note: The SOSC oscillator should already be running prior to entering SEC_IDLE mode. At least one of the secondary oscillator enable bits (SOSCGO, T1SOSCEN, T3SOSCEN or T5SOSCEN) must be set when the SLEEP instruction is executed. Otherwise, the main system clock will continue to operate in the previously selected mode and the corresponding IDLE mode will be entered (i.e., PRI_IDLE or RC_IDLE).

FIGURE 3-6: TRANSITION TIMING FOR ENTRY TO IDLE MODE



9.8 Register Definitions: Interrupt Control

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
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 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked interrupts
 0 = Disables all interrupts including peripherals
When IPEN = 1:
 1 = Enables all high priority interrupts
 0 = Disables all interrupts including low priority
- bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked peripheral interrupts
 0 = Disables all peripheral interrupts
When IPEN = 1:
 1 = Enables all low priority interrupts
 0 = Disables all low priority interrupts
- bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
 1 = Enables the TMR0 overflow interrupt
 0 = Disables the TMR0 overflow interrupt
- bit 4 **INT0IE:** INT0 External Interrupt Enable bit
 1 = Enables the INT0 external interrupt
 0 = Disables the INT0 external interrupt
- bit 3 **RBIE:** Port B Interrupt-On-Change (IOCx) Interrupt Enable bit⁽²⁾
 1 = Enables the IOCx port change interrupt
 0 = Disables the IOCx port change interrupt
- bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared by software)
 0 = TMR0 register did not overflow
- bit 1 **INT0IF:** INT0 External Interrupt Flag bit
 1 = The INT0 external interrupt occurred (must be cleared by software)
 0 = The INT0 external interrupt did not occur
- bit 0 **RBIF:** Port B Interrupt-On-Change (IOCx) Interrupt Flag bit⁽¹⁾
 1 = At least one of the IOC<3:0> (RB<7:4>) pins changed state (must be cleared by software)
 0 = None of the IOC<3:0> (RB<7:4>) pins have changed state

- Note 1:** A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.
- Note 2:** RB port change interrupts also require the individual pin IOCB enables.

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

PIC18(L)F2X/4XK22

REGISTER 9-2: INTCON2: INTERRUPT CONTROL 2 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBP}}\text{U}$	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 **$\overline{\text{RBP}}\text{U}$** : PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled provided that the pin is an input and the corresponding WPUB bit is set.
- 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 enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

PIC18(L)F2X/4XK22

REGISTER 9-15: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
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 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 6 **C1IP:** Comparator C1 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 5 **C2IP:** Comparator C2 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 3 **BCL1IP:** MSSP1 Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 2 **HLVDIP:** Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 0 **CCP2IP:** CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

PIC18(L)F2X/4XK22

10.2 PORTB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

EXAMPLE 10-2: INITIALIZING PORTB

```
MOVLB  0xF    ; Set BSR for banked SFRs
CLRF   PORTB  ; Initialize PORTB by
           ; clearing output
           ; data latches
CLRF   LATB   ; Alternate method
           ; to clear output
           ; data latches
MOVLW  0F0h   ; Value for init
MOVWF  ANSELB ; Enable RB<3:0> for
           ; digital input pins
           ; (not required if config bit
           ; PBADEN is clear)
MOVLW  0CFh   ; Value used to
           ; initialize data
           ; direction
MOVWF  TRISB  ; Set RB<3:0> as inputs
           ; RB<5:4> as outputs
           ; RB<7:6> as inputs
```

10.2.1 PORTB OUTPUT PRIORITY

Each PORTB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTB pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

10.3 Additional PORTB Pin Functions

PORTB pins RB<7:4> have an interrupt-on-change option. All PORTB pins have a weak pull-up option.

10.3.1 WEAK PULL-UPS

Each of the PORTB pins has an individually controlled weak internal pull-up. When set, each bit of the WPUB register enables the corresponding pin pull-up. When cleared, the RBPU bit of the INTCON2 register enables pull-ups on all pins which also have their corresponding WPUB bit set. When set, the RBPU bit disables all weak pull-ups. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Note: On a Power-on Reset, RB<5:0> are configured as analog inputs by default and read as '0'; RB<7:6> are configured as digital inputs.

When the PBADEN Configuration bit is set to '1', RB<5:0> will alternatively be configured as digital inputs on POR.

10.3.2 INTERRUPT-ON-CHANGE

Four of the PORTB pins (RB<7:4>) are individually configurable as interrupt-on-change pins. Control bits in the IOCB register enable (when set) or disable (when clear) the interrupt function for each pin.

When set, the RBIE bit of the INTCON register enables interrupts on all pins which also have their corresponding IOCB bit set. When clear, the RBIE bit disables all interrupt-on-changes.

Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt-on-change comparison).

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTB. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RBIF) in the INTCON register.

This interrupt can wake the device from the Sleep mode, or any of the Idle modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB to clear the mismatch condition (except when PORTB is the source or destination of a MOVFF instruction).
- Execute at least one instruction after reading or writing PORTB, then clear the flag bit, RBIF.

10.5 PORTD Registers

Note: PORTD is only available on 40-pin and 44-pin devices.

PORTD is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., disable the output driver). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

All of the PORTD pins are multiplexed with analog and digital peripheral modules. See Table 10-11.

Note: On a Power-on Reset, these pins are configured as analog inputs.

EXAMPLE 10-4: INITIALIZING PORTD

```
MOVLB  0xF      ; Set BSR for banked SFRs
CLRF   PORTD    ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF   LATD     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISD    ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
MOVLW  30h     ; Value used to
                ; enable digital inputs
MOVWF  ANSEL    ; RD<3:0> dig input enable
                ; RC<7:6> dig input enable
```

10.5.1 PORTD OUTPUT PRIORITY

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTD pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

PIC18(L)F2X/4XK22

The I²C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited Multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDAx hold times

Figure 15-2 is a block diagram of the I²C interface module in Master mode. Figure 15-3 is a diagram of the I²C interface module in Slave mode.

The PIC18(L)F2X/4XK22 has two MSSP modules, MSSP1 and MSSP2, each module operating independently from the other.

- Note 1:** In devices with more than one MSSP module, it is very important to pay close attention to SSPxCONx register names. SSP1CON1 and SSP1CON2 registers control different operational aspects of the same module, while SSP1CON1 and SSP2CON1 control the same features for two different modules.
- 2:** Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. Register names, module I/O signals, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required.

FIGURE 15-2: MSSPx BLOCK DIAGRAM (I²C MASTER MODE)

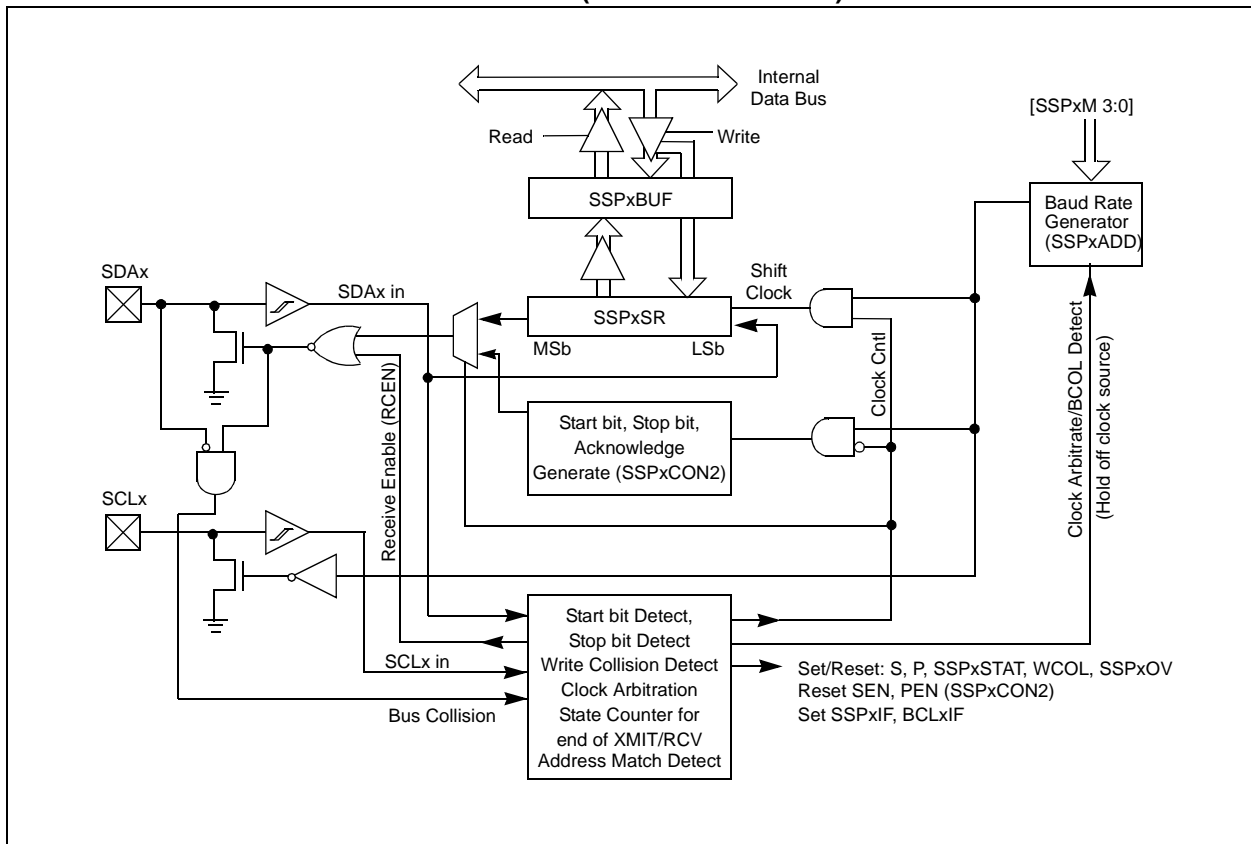
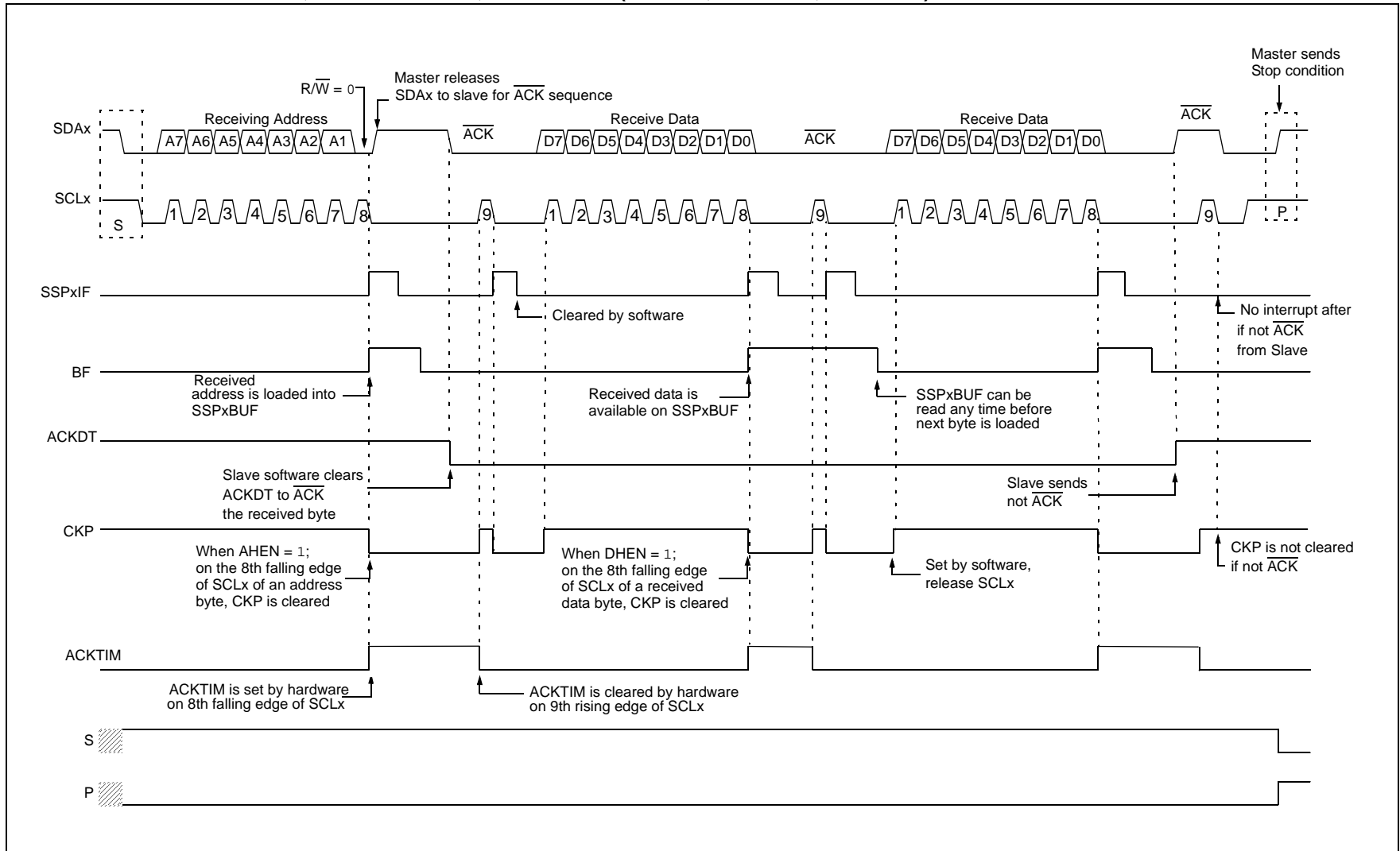


FIGURE 15-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



PIC18(L)F2X/4XK22

REGISTER 15-7: SSPxADD: MSSPx ADDRESS AND BAUD RATE REGISTER (I²C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD<7:0>							
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

Master mode:

bit 7-0 **ADD<7:0>**: Baud Rate Clock Divider bits
SCLx pin clock period = ((ADD<7:0> + 1) * 4) / Fosc

10-Bit Slave mode — Most Significant Address byte:

bit 7-3 **Not used**: Unused for Most Significant Address byte. Bit state of this register is a “don't care”. Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD<2:1>**: Two Most Significant bits of 10-bit address

bit 0 **Not used**: Unused in this mode. Bit state is a “don't care”.

10-Bit Slave mode — Least Significant Address byte:

bit 7-0 **ADD<7:0>**: Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 7-1 **ADD<7:1>**: 7-bit address

bit 0 **Not used**: Unused in this mode. Bit state is a “don't care”.

16.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V_{OH} Mark state which represents a '1' data bit, and a V_{OL} Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 16-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

16.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 16-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREGx register.

16.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTAx register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTAx register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTAx register enables the EUSART and automatically configures the TXx/CKx I/O pin as an output. If the TXx/CKx pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note: The TXxIF transmitter interrupt flag is set when the TXEN enable bit is set.

16.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREGx register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREGx is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREGx until the Stop bit of the previous character has been transmitted. The pending character in the TXREGx is then transferred to the TSR in one T_{cy} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREGx.

16.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the CKTXP bit of the BAUDCONx register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the CKTXP bit to '1' will invert the transmit data resulting in low true idle and data bits. The CKTXP bit controls transmit data polarity only in Asynchronous mode. In Synchronous mode the CKTXP bit has a different function.

16.1.1.4 Transmit Interrupt Flag

The TXxIF interrupt flag bit of the PIR1/PIR3 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREGx. In other words, the TXxIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREGx. The TXxIF flag bit is not cleared immediately upon writing TXREGx. TXxIF becomes valid in the second instruction cycle following the write execution. Polling TXxIF immediately following the TXREGx write will return invalid results. The TXxIF bit is read-only, it cannot be set or cleared by software.

The TXxIF interrupt can be enabled by setting the TXxIE interrupt enable bit of the PIE1/PIE3 register. However, the TXxIF flag bit will be set whenever the TXREGx is empty, regardless of the state of TXxIE enable bit.

To use interrupts when transmitting data, set the TXxIE bit only when there is more data to send. Clear the TXxIE interrupt enable bit upon writing the last character of the transmission to the TXREGx.

PIC18(L)F2X/4XK22

REGISTER 16-3: BAUDCONx: BAUD RATE CONTROL REGISTER

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
 1 = Auto-baud timer overflowed
 0 = Auto-baud timer did not overflow
Synchronous mode:
 Don't care
- bit 6 **RCIDL:** Receive Idle Flag bit
Asynchronous mode:
 1 = Receiver is Idle
 0 = Start bit has been detected and the receiver is active
Synchronous mode:
 Don't care
- bit 5 **DTRXP:** Data/Receive Polarity Select bit
Asynchronous mode:
 1 = Receive data (RXx) is inverted (active-low)
 0 = Receive data (RXx) is not inverted (active-high)
Synchronous mode:
 1 = Data (DTx) is inverted (active-low)
 0 = Data (DTx) is not inverted (active-high)
- bit 4 **CKTXP:** Clock/Transmit Polarity Select bit
Asynchronous mode:
 1 = Idle state for transmit (TXx) is low
 0 = Idle state for transmit (TXx) is high
Synchronous mode:
 1 = Data changes on the falling edge of the clock and is sampled on the rising edge of the clock
 0 = Data changes on the rising edge of the clock and is sampled on the falling edge of the clock
- bit 3 **BRG16:** 16-bit Baud Rate Generator bit
 1 = 16-bit Baud Rate Generator is used (SPBRGHx:SPBRGx)
 0 = 8-bit Baud Rate Generator is used (SPBRGx)
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
 1 = Receiver is waiting for a falling edge. No character will be received but RCxIF will be set on the falling edge. WUE will automatically clear on the rising edge.
 0 = Receiver is operating normally
Synchronous mode:
 Don't care
- bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
 1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
 0 = Auto-Baud Detect mode is disabled
Synchronous mode:
 Don't care

PIC18(L)F2X/4XK22

16.5 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

16.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTAx register configures the device for synchronous operation. Setting the CSRC bit of the TXSTAx register configures the device as a master. Clearing the SREN and CREN bits of the RCSTAx register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTAx register enables the EUSART. If the RXx/DTx or TXx/CKx pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

The TRIS bits corresponding to the RXx/DTx and TXx/CKx pins should be set.

16.5.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TXx/CKx line. The TXx/CKx pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

16.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the CKTXP bit of the BAUDCONx register. Setting the CKTXP bit sets the clock Idle state as high. When the CKTXP bit is set, the data changes on the falling edge of each clock and is sampled on the rising edge of each clock. Clearing the CKTXP bit sets the Idle state as low. When the CKTXP bit is cleared, the data changes on the rising edge of each clock and is sampled on the falling edge of each clock.

16.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RXx/DTx pin. The RXx/DTx and TXx/CKx pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREGx register. If the TSR still contains all or part of a previous character the new character data is held in the TXREGx until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREGx is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREGx.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note: The TSR register is not mapped in data memory, so it is not available to the user.

16.5.1.4 Data Polarity

The polarity of the transmit and receive data can be controlled with the DTRXP bit of the BAUDCONx register. The default state of this bit is '0' which selects high true transmit and receive data. Setting the DTRXP bit to '1' will invert the data resulting in low true transmit and receive data.

PIC18(L)F2X/4XK22

TABLE 16-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON1	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
BAUDCON2	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
SPBRG1	EUSART1 Baud Rate Generator, Low Byte								—
SPBRGH1	EUSART1 Baud Rate Generator, High Byte								—
SPBRG2	EUSART2 Baud Rate Generator, Low Byte								—
SPBRGH2	EUSART2 Baud Rate Generator, High Byte								—
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD ⁽¹⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
ANSEL	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	150
ANSELD ⁽¹⁾	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	150
TXREG1	EUSART1 Transmit Register								—
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	269
TXREG2	EUSART2 Transmit Register								—
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	269

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous master transmission.

Note 1: PIC18(L)F4XK22 devices.

Note 2: PIC18(L)F2XK22 devices.

PIC18(L)F2X/4XK22

FIGURE 16-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

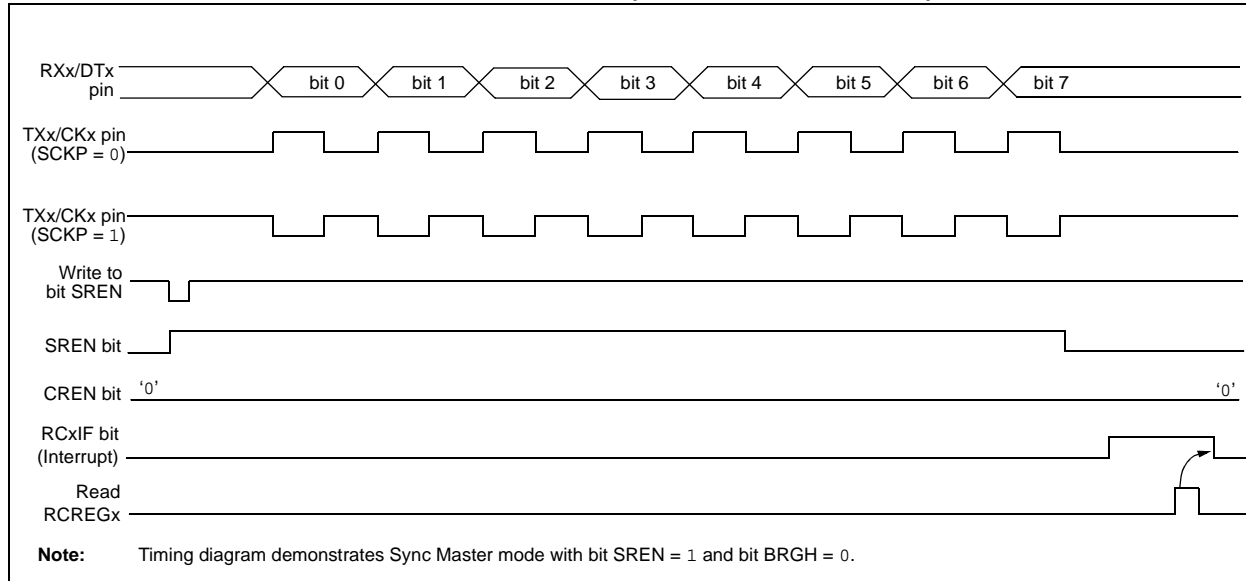


TABLE 16-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON1	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
BAUDCON2	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	271
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
RCREG1	EUSART1 Receive Register								—
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
RCREG2	EUSART2 Receive Register								—
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	270
SPBRG1	EUSART1 Baud Rate Generator, Low Byte								—
SPBRGH1	EUSART1 Baud Rate Generator, High Byte								—
SPBRG2	EUSART2 Baud Rate Generator, Low Byte								—
SPBRGH2	EUSART2 Baud Rate Generator, High Byte								—
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	269
TXSTA2	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	269

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous master reception.

PIC18(L)F2X/4XK22

24.3 Watchdog Timer (WDT)

For PIC18(L)F2X/4XK22 devices, the WDT is driven by the LFINTOSC source. When the WDT is enabled, the clock source is also enabled. The nominal WDT period is 4 ms and has the same stability as the LFINTOSC oscillator.

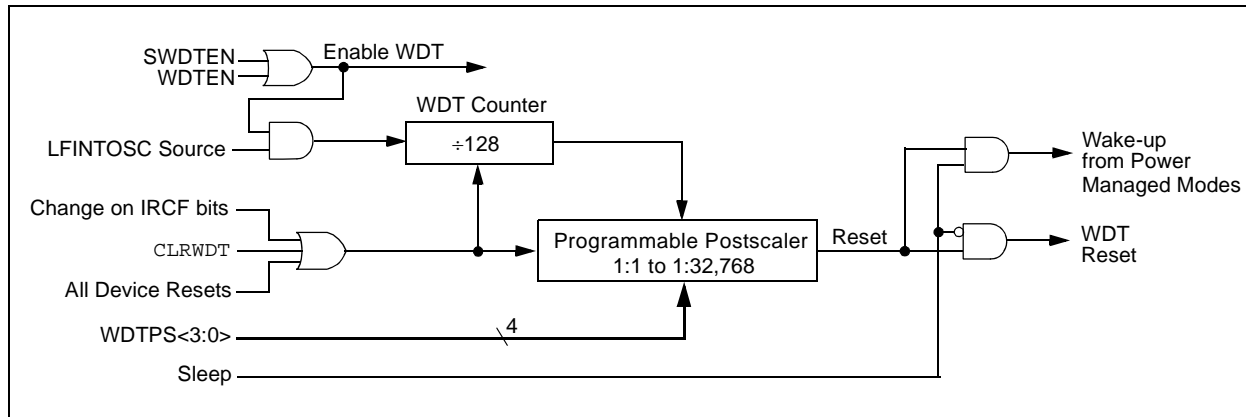
The 4 ms period of the WDT is multiplied by a 16-bit postscaler. Any output of the WDT postscaler is selected by a multiplexer, controlled by bits in Configuration Register 2H. Available periods range from 4 ms to 131.072 seconds (2.18 minutes). The WDT and postscaler are cleared when any of the following events occur: a SLEEP or CLRWDT instruction is executed, the IRCF bits of the OSCCON register are changed or a clock failure has occurred.

Note 1: The CLRWDT and SLEEP instructions clear the WDT and postscaler counts when executed.

2: Changing the setting of the IRCF bits of the OSCCON register clears the WDT and postscaler counts.

3: When a CLRWDT instruction is executed, the postscaler count will be cleared.

FIGURE 24-1: WDT BLOCK DIAGRAM



PIC18(L)F2X/4XK22

26.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

26.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

26.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

26.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

PIC18(L)F2X/4XK22

FIGURE 28-48: PIC18LF2X/4XK22 TYPICAL I_{DD} : PRI_RUN EC MEDIUM POWER

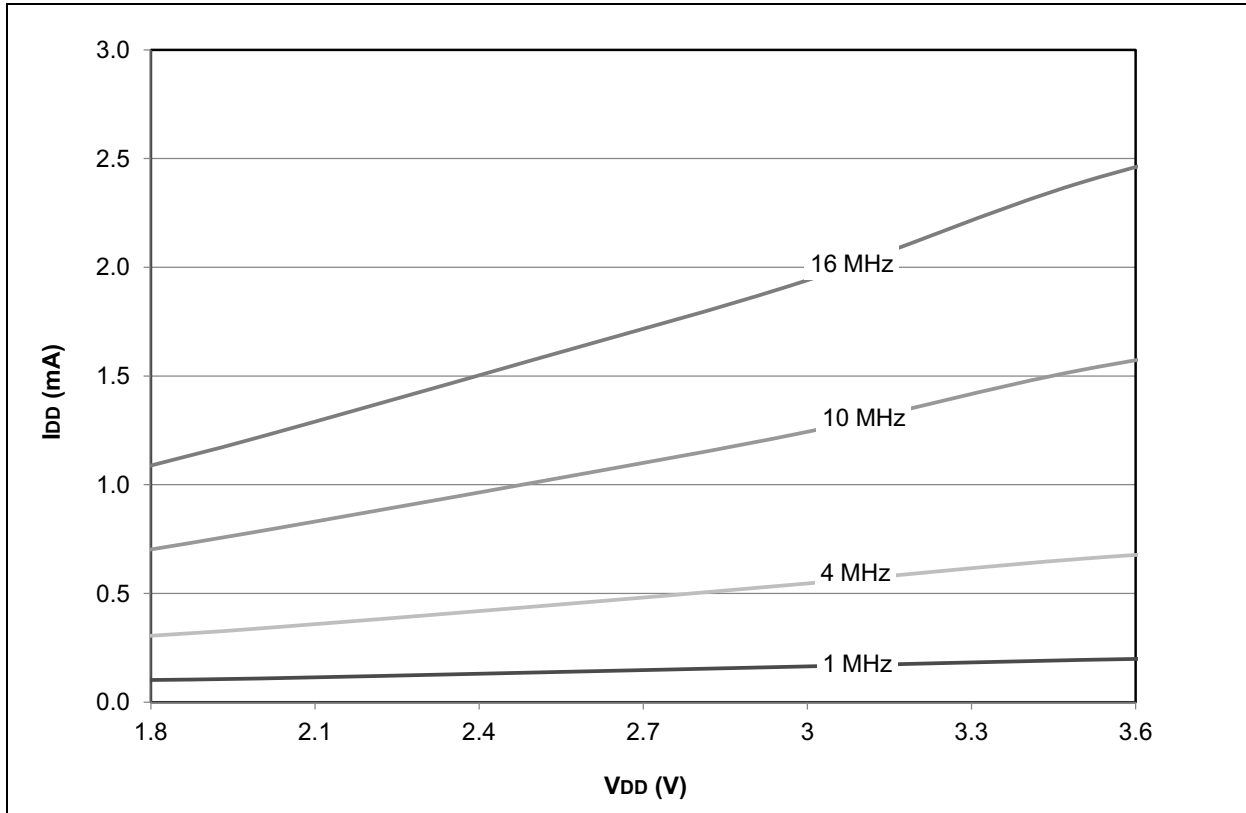
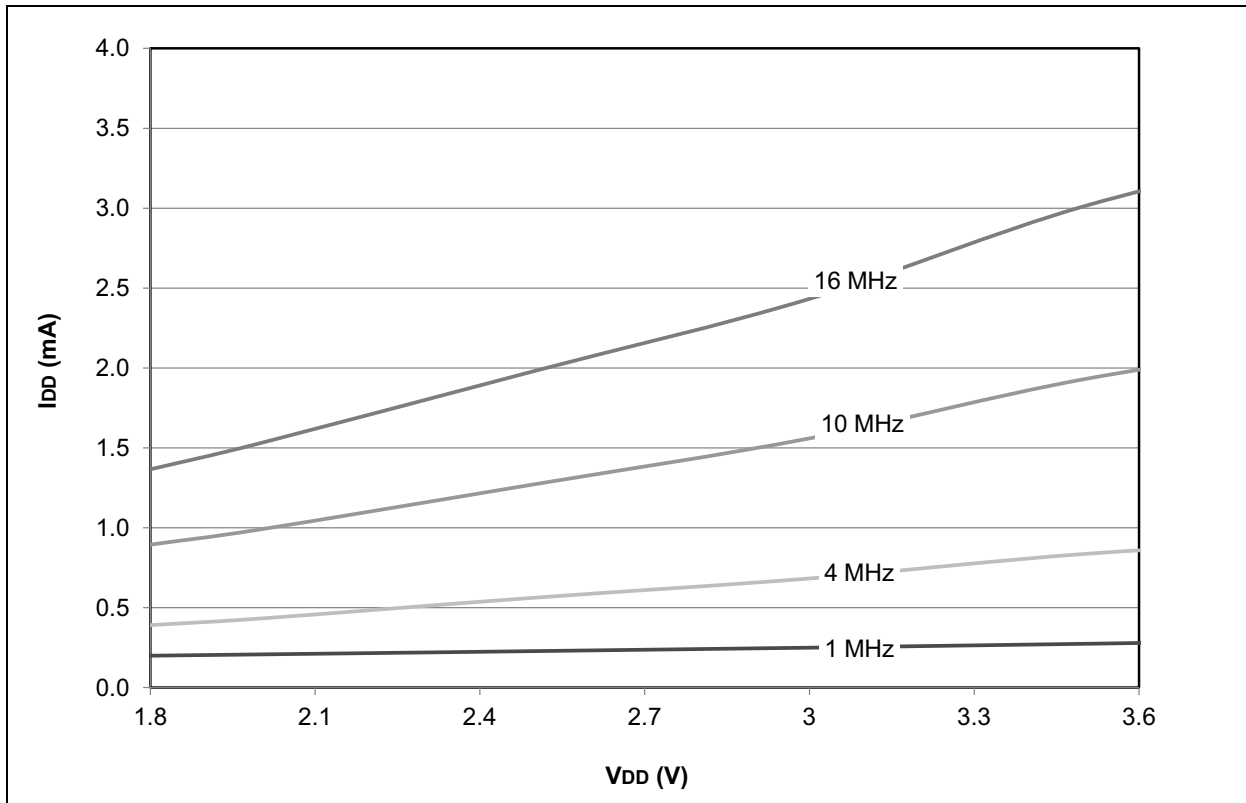


FIGURE 28-49: PIC18LF2X/4XK22 MAXIMUM I_{DD} : PRI_RUN EC MEDIUM POWER



PIC18(L)F2X/4XK22

FIGURE 28-100: PIC18(L)F2X/4XK22 HF-INTOSC FREQUENCY vs. TEMPERATURE at 16 MHz
MIN / MAX: $\pm 2\%$, T = 0°C to +70°C
 $+2\% / -3\%$, T = +70°C to +85°C
 $\pm 5\%$, T = -40°C to 0°C and +85°C to +125°C

