



Welcome to [E-XFL.COM](https://www.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)	1.8V ~ 3.6V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf43k22-e-ml

PIC18(L)F2X/4XK22

FIGURE 2-2: INTERNAL OSCILLATOR MUX BLOCK DIAGRAM

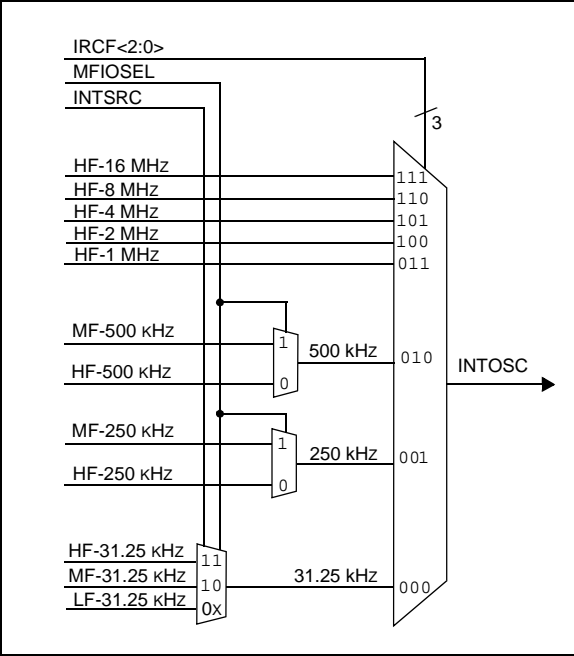


FIGURE 2-3: PLL_SELECT BLOCK DIAGRAM

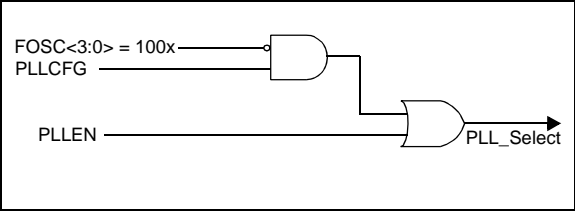
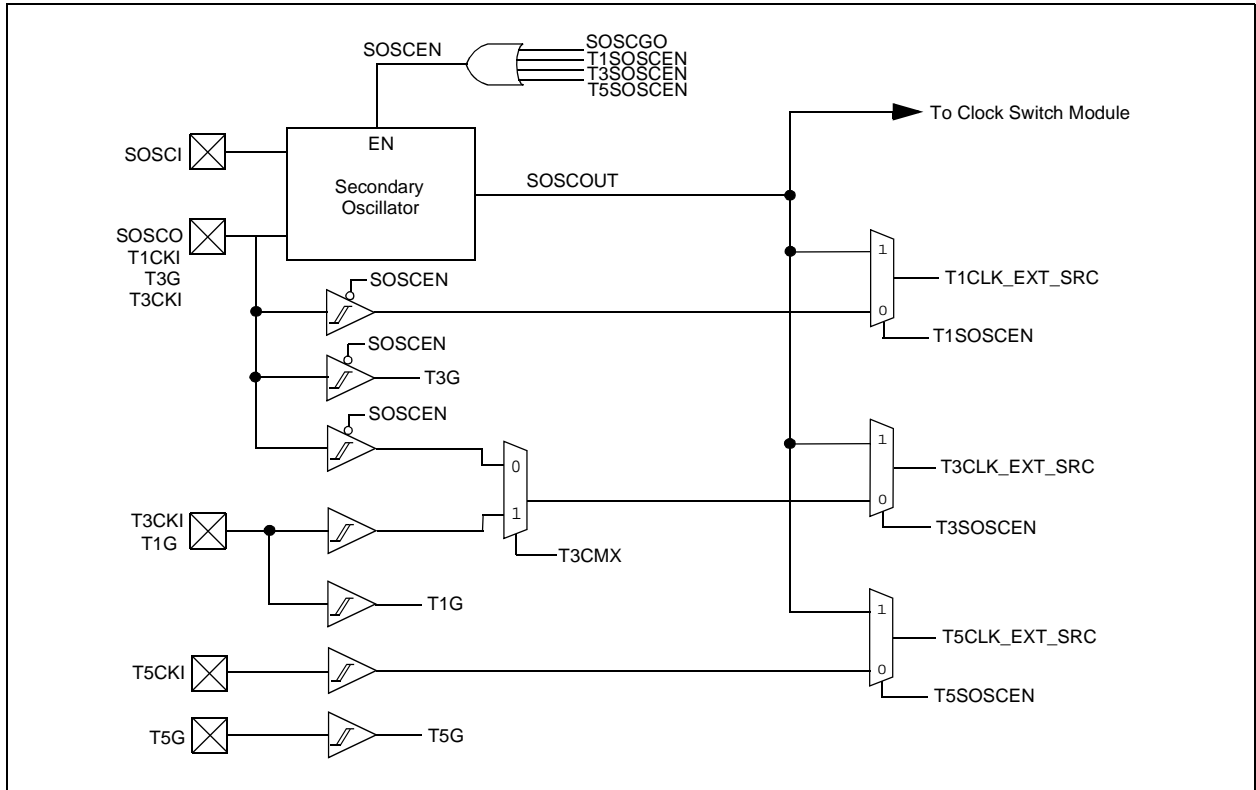


TABLE 2-1: PLL_SELECT TRUTH TABLE

Primary Clock MUX Source	FOSC<3:0>	PLLCFG	PLLEN	PLL_Select
FOSC (any source)	0000-1111	0	0	0
OSC1/OSC2 (external source)	0000-0111 1010-1111	1	x	1
		0	1	1
INTOSC (internal source)	1000-1001	x	0	0
		x	1	1

FIGURE 2-4: SECONDARY OSCILLATOR AND EXTERNAL CLOCK INPUTS



3.5 Exiting Idle and Sleep Modes

An exit from Sleep mode or any of the Idle modes is triggered by any one of the following:

- an interrupt
- a Reset
- a Watchdog Time-out

This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed modes (see **Section 3.2 “Run Modes”**, **Section 3.3 “Sleep Mode”** and **Section 3.4 “Idle Modes”**).

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode or the Sleep mode to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

The instruction immediately following the `SLEEP` instruction is executed on all exits by interrupt from Idle or Sleep modes. Code execution then branches to the interrupt vector if the GIE/GIEH bit of the INTCON register is set, otherwise code execution continues without branching (see **Section 9.0 “Interrupts”**).

A fixed delay of interval T_{CSD} following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 3.2 “Run Modes”** and **Section 3.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 24.3 “Watchdog Timer (WDT)”**).

The WDT timer and postscaler are cleared by any one of the following:

- executing a `SLEEP` instruction
- executing a `CLRWDT` instruction
- the loss of the currently selected clock source when the Fail-Safe Clock Monitor is enabled
- modifying the IRCF bits in the OSCCON register when the internal oscillator block is the device clock source

3.5.3 EXIT BY RESET

Exiting Sleep and Idle modes by Reset causes code execution to restart at address '0'. See **Section 4.0 “Reset”** for more details.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up and the type of oscillator.

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- `PRI_IDLE` mode, where the primary clock source is not stopped and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay since it is already running (`PRI_IDLE`), or normally does not require an oscillator start-up delay (`RC`, `EC`, `INTOSC`, and `INTOSCIO` modes). However, a fixed delay of interval T_{CSD} following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

PIC18(L)F2X/4XK22

4.2 Register Definitions: Reset Control

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0/0	R/W-q/u	U-0	R/W-1/q	R-1/q	R-1/q	R/W-q/u	R/W-0/q
IPEN	SBOREN ⁽¹⁾	—	RI	TO	PD	POR ⁽²⁾	BOR
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

'1' = Bit is set

'0' = Bit is cleared

-n/n = Value at POR and BOR/Value at all other Resets

x = Bit is unknown

u = unchanged

q = depends on condition

- bit 7 **IPEN:** Interrupt Priority Enable bit
 1 = Enable priority levels on interrupts
 0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6 **SBOREN:** BOR Software Enable bit⁽¹⁾
If BOREN<1:0> = 01:
 1 = BOR is enabled
 0 = BOR is disabled
If BOREN<1:0> = 00, 10 or 11:
 Bit is disabled and read as '0'.
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **RI:** RESET Instruction Flag bit
 1 = The RESET instruction was not executed (set by firmware or Power-on Reset)
 0 = The RESET instruction was executed causing a device Reset (must be set in firmware after a code-executed Reset occurs)
- bit 3 **TO:** Watchdog Time-out Flag bit
 1 = Set by power-up, CLRWDT instruction or SLEEP instruction
 0 = A WDT time-out occurred
- bit 2 **PD:** Power-down Detection Flag bit
 1 = Set by power-up or by the CLRWDT instruction
 0 = Set by execution of the SLEEP instruction
- bit 1 **POR:** Power-on Reset Status bit⁽²⁾
 1 = No Power-on Reset occurred
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
- bit 0 **BOR:** Brown-out Reset Status bit⁽³⁾
 1 = A Brown-out Reset has not occurred (set by firmware only)
 0 = A Brown-out Reset occurred (must be set by firmware after a POR or Brown-out Reset occurs)

- Note 1:** When CONFIG2L[2:1] = 01, then the SBOREN Reset state is '1'; otherwise, it is '0'.
- Note 2:** The actual Reset value of POR is determined by the type of device Reset. See the notes following this register and **Section 4.7 "Reset State of Registers"** for additional information.
- Note 3:** See Table 4-1.

Note 1: Brown-out Reset is indicated when BOR is '0' and POR is '1' (assuming that both POR and BOR were set to '1' by firmware immediately after POR).

Note 2: It is recommended that the POR bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

5.4.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the “Access RAM” and is composed of GPRs. This upper half is also where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figures 5-5 through 5-7).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle, without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 5.7.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

5.4.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM, which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

5.4.4 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. SFRs start at the top of data memory (FFFh) and extend downward to occupy the top portion of Bank 15 (F38h 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.

PIC18(L)F2X/4XK22

TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FA2h	IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	1111 1111
FA1h	PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	0000 0000
FA0h	PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	0000 0000
F9Fh	IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	-111 1111
F9Eh	PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	-000 0000
F9Dh	PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	-000 0000
F9Ch	HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL<3:0>				0000 0000
F9Bh	OSCTUNE	INTSRC	PLLEN	TUN<5:0>						00xx xxxxx
F96h	TRISE	WPUE3	—	—	—	—	TRISE2 ⁽¹⁾	TRISE1 ⁽¹⁾	TRISE0 ⁽¹⁾	1--- -111
F95h	TRISD ⁽¹⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111
F94h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111
F93h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111
F92h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111
F8Dh	LATE ⁽¹⁾	—	—	—	—	—	LATE2	LATE1	LATE0	---- -xxxx
F8Ch	LATD ⁽¹⁾	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxxx
F8Bh	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxxx
F8Ah	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxxx
F89h	LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxxx
F84h	PORTE ⁽²⁾	—	—	—	—	RE3	—	—	—	---- x---
	PORTE ⁽¹⁾	—	—	—	—	RE3	RE2	RE1	RE0	---- x000
F83h	PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	0000 0000
F82h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	0000 00xx
F81h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxx0 0000
F80h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000
F7Fh	IPR5	—	—	—	—	—	TMR6IP	TMR5IP	TMR4IP	---- -111
F7Eh	PIR5	—	—	—	—	—	TMR6IF	TMR5IF	TMR4IF	---- -111
F7Dh	PIE5	—	—	—	—	—	TMR6IE	TMR5IE	TMR4IE	---- -000
F7Ch	IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	---- -000
F7Bh	PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	---- -000
F7Ah	PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	---- -000
F79h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	C1SP	C1R	C1CH<1:0>		0000 1000
F78h	CM2CON0	C2ON	C2OUT	C2OE	C2POL	C2SP	C2R	C2CH<1:0>		0000 1000
F77h	CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	C1HYS	C2HYS	C1SYNC	C2SYNC	0000 0000
F76h	SPBRGH2	EUSART2 Baud Rate Generator, High Byte								0000 0000
F75h	SPBRG2	EUSART2 Baud Rate Generator, Low Byte								0000 0000
F74h	RCREG2	EUSART2 Receive Register								0000 0000
F73h	TXREG2	EUSART2 Transmit Register								0000 0000
F72h	TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010
F71h	RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
F70h	BAUDCON2	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	01x0 0-00
F6Fh	SSP2BUF	SSP2 Receive Buffer/Transmit Register								xxxx xxxxx
F6Eh	SSP2ADD	SSP2 Address Register in I ² C Slave Mode. SSP2 Baud Rate Reload Register in I ² C Master Mode								0000 0000
F6Dh	SSP2STAT	SMP	CKE	D \overline{A}	P	S	R \overline{W}	UA	BF	0000 0000
F6Ch	SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000
F6Bh	SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
F6Ah	SSP2MSK	SSP1 MASK Register bits								1111 1111
F69h	SSP2CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

- Note**
- 1: PIC18(L)F4XK22 devices only.
 - 2: PIC18(L)F2XK22 devices only.
 - 3: PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.
 - 4: PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

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
RBP \overline{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 **RBP \overline{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

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.

PIC18(L)F2X/4XK22

TABLE 10-8: PORTC I/O SUMMARY (CONTINUED)

Pin Name	Function	TRIS Setting	ANSEL setting	Pin Type	Buffer Type	Description
RC5/SDO1/AN17	RC5	0	0	O	DIG	LATC<5> data output; not affected by analog input.
		1	0	I	ST	PORTC<5> data input; disabled when analog input enabled.
	SDO1	0	0	O	DIG	MSSP1 SPI data output.
	AN17	1	1	I	AN	Analog input 17.
RC6/P3A/CCP3/TX1/CK1/AN18	RC6	0	0	O	DIG	LATC<6> data output; not affected by analog input.
		1	0	I	ST	PORTC<6> data input; disabled when analog input enabled.
	P3A ^{(2), (3)}	0	0	O	CMOS	Enhanced CCP3 PWM output 1.
	CCP3 ^{(2), (3)}	0	0	O	DIG	Compare 3 output/PWM 3 output.
		1	0	I	ST	Capture 3 input.
	TX1	1	0	O	DIG	EUSART asynchronous transmit data output.
	CK1	1	0	O	DIG	EUSART synchronous serial clock output.
		1	0	I	ST	EUSART synchronous serial clock input.
	AN18	1	1	I	AN	Analog input 18.
RC7/P3B/RX1/DT1/AN19	RC7	0	0	O	DIG	LATC<7> data output; not affected by analog input.
		1	0	I	ST	PORTC<7> data input; disabled when analog input enabled.
	P3B	0	0	O	CMOS	Enhanced CCP3 PWM output 2.
	RX1	1	0	I	ST	EUSART asynchronous receive data in.
	DT1	1	0	O	DIG	EUSART synchronous serial data output.
		1	0	I	ST	EUSART synchronous serial data input.
	AN19	1	1	I	AN	Analog input 19.

Legend: AN = Analog input or output; TTL = TTL compatible input; HV = High Voltage; OD = Open Drain; XTAL = Crystal; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I²C = Schmitt Trigger input with I²C.

- Note 1:** Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.
- Note 2:** Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.
- Note 3:** Function on PORTD and PORTE for PIC18FXXK22 devices.

15.5.3 SLAVE TRANSMISSION

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an \overline{ACK} pulse is sent by the slave on the ninth bit.

Following the \overline{ACK} , slave hardware clears the CKP bit and the SCLx pin is held low (see **Section 15.5.6 “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 the SCLx pin should be released by setting the CKP bit of the SSPxCON1 register. 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.

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. This \overline{ACK} value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not \overline{ACK}), then the data transfer is complete. In this case, when the not \overline{ACK} is latched by the slave, the slave goes Idle and waits 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, the SCLx pin must be released by setting bit CKP.

An MSSPx interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by 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.

15.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDAx line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCLxIF bit of the PIRx register is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

15.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 15-18 can be used as a reference to this list.

1. Master sends a Start condition on SDAx and SCLx.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with $\overline{R/W}$ bit set is received by the slave setting SSPxIF bit.
4. Slave hardware generates an \overline{ACK} and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. $\overline{R/W}$ is set so CKP was automatically cleared after the \overline{ACK} .
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCLx, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the \overline{ACK} response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master \overline{ACKs} the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCLx (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not \overline{ACK} ; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

PIC18(L)F2X/4XK22

TABLE 15-2: REGISTERS ASSOCIATED WITH I²C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	149
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1 ⁽¹⁾	ANSB0 ⁽¹⁾	150
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	150
ANSELD	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1 ⁽²⁾	ANS0 ⁽²⁾	150
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	122
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	118
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	113
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
SSP1ADD	SSP1 Address Register in I ² C Slave mode. SSP1 Baud Rate Reload Register in I ² C Master mode.								258
SSP1BUF	SSP1 Receive Buffer/Transmit Register								—
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				253
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	255
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	256
SSP1MSK	SSP1 MASK Register bits								257
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	252
SSP2ADD	SSP2 Address Register in I ² C Slave mode. SSP2 Baud Rate Reload Register in I ² C Master mode.								258
SSP2BUF	SSP2 Receive Buffer/Transmit Register								—
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				253
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	255
SSP2CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	256
SSP2MSK	SSP1 MASK Register bits								257
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	252
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1 ⁽¹⁾	TRISB0 ⁽¹⁾	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1 ⁽²⁾	TRISD0 ⁽²⁾	151

Legend: Shaded bits are not used by the MSSPx in I²C mode.

Note 1: PIC18(L)F2XK22 devices.

2: PIC18(L)F4XK22 devices.

FIGURE 16-5: ASYNCHRONOUS RECEPTION

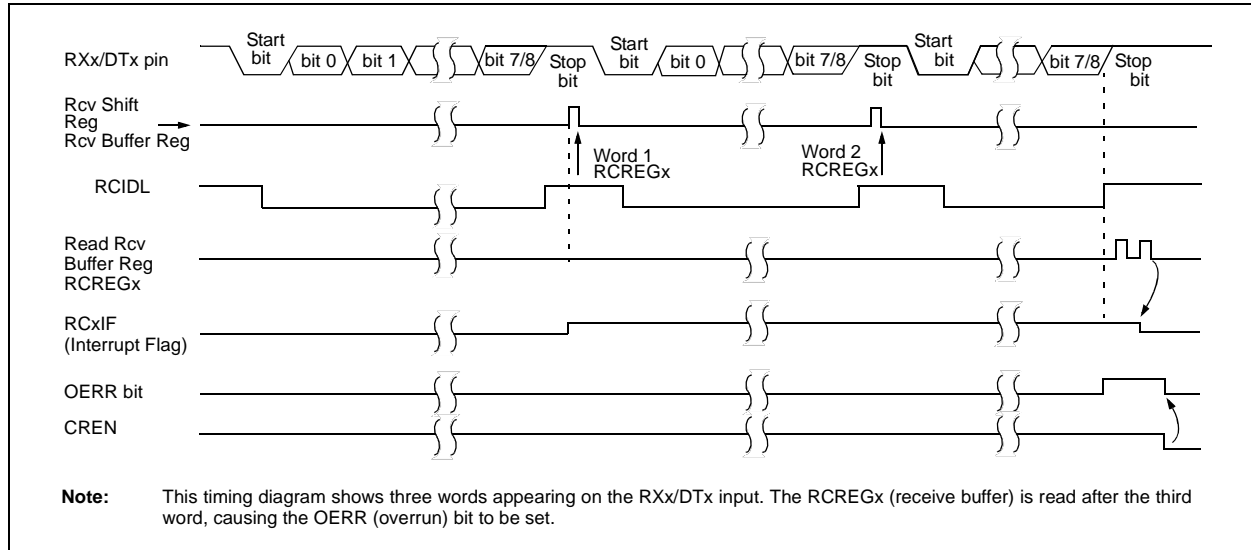


TABLE 16-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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								—
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
ANSEL ⁽¹⁾	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	150
TXSTA1	CSRC	TX9	TXEN	SYNC	SENCB	BRGH	TRMT	TX9D	269
TXSTA2	CSRC	TX9	TXEN	SYNC	SENCB	BRGH	TRMT	TX9D	269

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

Note 1: PIC18(L)F4XK22 devices.
2: PIC18(L)F2XK22 devices.

PIC18(L)F2X/4XK22

REGISTER 16-2: RCSTAx: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RXx/DTx and TXx/CKx pins as serial port pins)
0 = Serial port disabled (held in Reset)
- bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive
- bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care
- bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREGx register and receive next valid byte)
0 = No framing error
- bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error
- bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

PIC18(L)F2X/4XK22

17.2 ADC Operation

17.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will, depending on the ACQT bits of the ADCON2 register, either immediately start the Analog-to-Digital conversion or start an acquisition delay followed by the Analog-to-Digital conversion.

Figure 17-3 shows the operation of the A/D converter after the GO bit has been set and the ACQT<2:0> bits are cleared. A conversion is started after the following instruction to allow entry into SLEEP mode before the conversion begins.

Figure 17-4 shows the operation of the A/D converter after the GO bit has been set and the ACQT<2:0> bits are set to '010' which selects a 4 TAD acquisition time before the conversion starts.

Note: The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 17.2.10 "A/D Conversion Procedure"**.

FIGURE 17-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

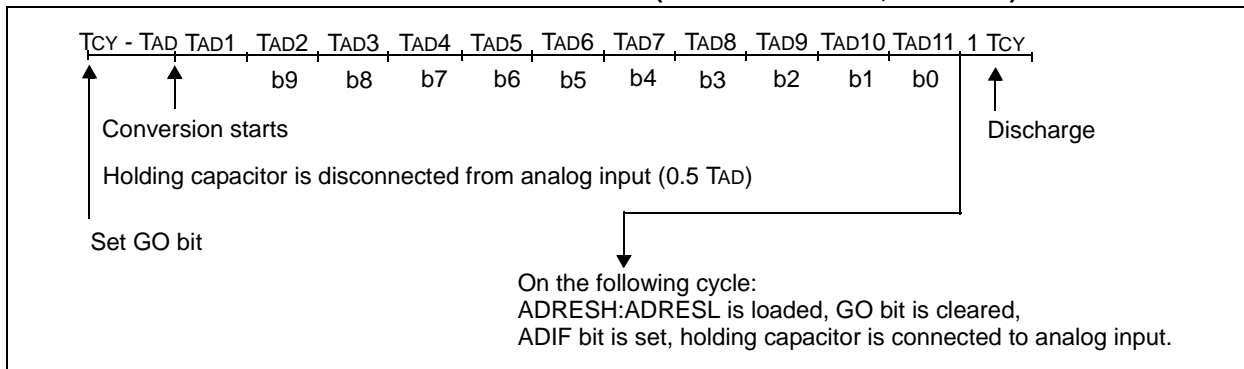
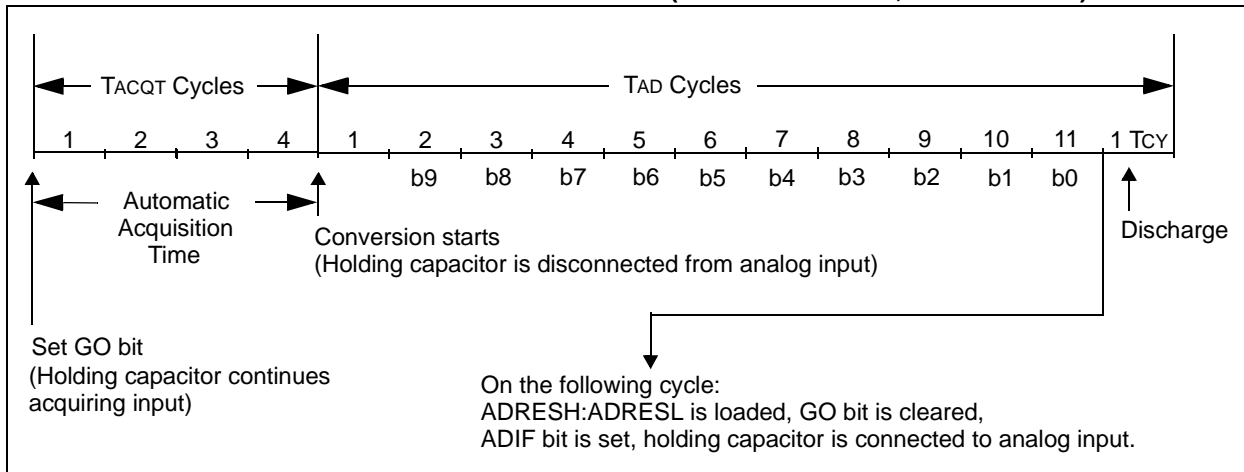


FIGURE 17-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



PIC18(L)F2X/4XK22

REGISTER 24-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	BORV<1:0> ⁽¹⁾		BOREN<1:0> ⁽²⁾		PWRTEN ⁽²⁾
bit 7			bit 0				

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits⁽¹⁾

11 = VBOR set to 1.9V nominal

10 = VBOR set to 2.2V nominal

01 = VBOR set to 2.5V nominal

00 = VBOR set to 2.85V nominal

bit 2-1 **BOREN<1:0>:** Brown-out Reset Enable bits⁽²⁾

11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)

10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode
(SBOREN is disabled)

01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)

00 = Brown-out Reset disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit⁽²⁾

1 = PWRT disabled

0 = PWRT enabled

Note 1: See Section 27.1 “DC Characteristics: Supply Voltage, PIC18(L)F2X/4XK22” for specifications.

Note 2: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

POP		Pop Top of Return Stack								
Syntax:	POP									
Operands:	None									
Operation:	(TOS) → bit bucket									
Status Affected:	None									
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0110</td></tr></table>						0000	0000	0000	0110
0000	0000	0000	0110							
Description:	<p>The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack.</p> <p>This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.</p>									
Words:	1									
Cycles:	1									
Q Cycle Activity:										
	Q1	Q2	Q3	Q4						
	Decode	No operation	POP TOS value	No operation						

Example:

POP		
GOTO	NEW	

Before Instruction

TOS	=	0031A2h
Stack (1 level down)	=	014332h

After Instruction

TOS	=	014332h
PC	=	NEW

PUSH		Push Top of Return Stack								
Syntax:	PUSH									
Operands:	None									
Operation:	(PC + 2) → TOS									
Status Affected:	None									
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0101</td></tr></table>						0000	0000	0000	0101
0000	0000	0000	0101							
Description:	<p>The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.</p>									
Words:	1									
Cycles:	1									
Q Cycle Activity:										
	Q1	Q2	Q3	Q4						
	Decode	PUSH PC + 2 onto return stack	No operation	No operation						

Example:

PUSH		
------	--	--

Before Instruction

TOS	=	345Ah
PC	=	0124h

After Instruction

PC	=	0126h
TOS	=	0126h
Stack (1 level down)	=	345Ah

SUBWFB Subtract W from f with Borrow

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

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

Operation: $(f) - (W) - (\overline{C}) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

0101	10da	ffff	ffff
------	------	------	------

Description: Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). 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. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 "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 1: SUBWFB REG, 1, 0

Before Instruction

REG	=	19h	(0001 1001)
W	=	0Dh	(0000 1101)
C	=	1	

After Instruction

REG	=	0Ch	(0000 1100)
W	=	0Dh	(0000 1101)
C	=	1	
Z	=	0	
N	=	0	; result is positive

Example 2: SUBWFB REG, 0, 0

Before Instruction

REG	=	1Bh	(0001 1011)
W	=	1Ah	(0001 1010)
C	=	0	

After Instruction

REG	=	1Bh	(0001 1011)
W	=	00h	
C	=	1	
Z	=	1	; result is zero
N	=	0	

Example 3: SUBWFB REG, 1, 0

Before Instruction

REG	=	03h	(0000 0011)
W	=	0Eh	(0000 1110)
C	=	1	

After Instruction

REG	=	F5h	(1111 0101) ; [2's comp]
W	=	0Eh	(0000 1110)
C	=	0	
Z	=	0	
N	=	1	; result is negative

SWAPF Swap f

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

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

Operation: $(f<3:0>) \rightarrow \text{dest}<7:4>$,
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

Status Affected: None

Encoding:

0011	10da	ffff	ffff
------	------	------	------

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed 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. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 "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: SWAPF REG, 1, 0

Before Instruction

REG	=	53h
-----	---	-----

After Instruction

REG	=	35h
-----	---	-----

FIGURE 27-5: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

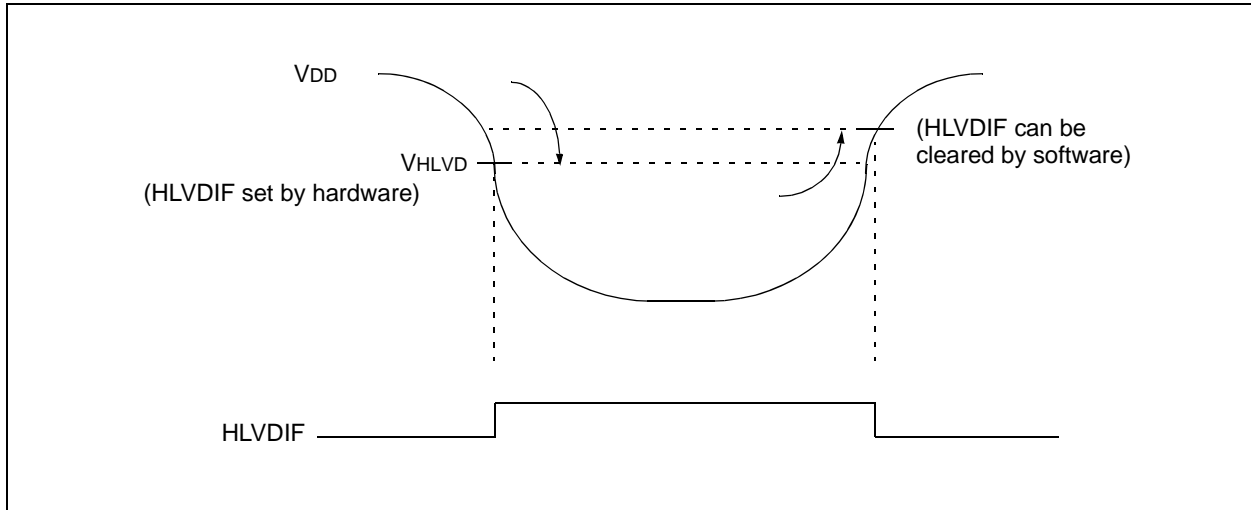


TABLE 27-5: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Symbol	Characteristic	HLVDL<3:0>	Min	Typ†	Max	Units	Conditions
		HLVD Voltage on VDD Transition High-to-Low	0000	1.69	1.84	1.99	V	
			0001	1.92	2.07	2.22	V	
			0010	2.08	2.28	2.48	V	
			0011	2.24	2.44	2.64	V	
			0100	2.34	2.54	2.74	V	
			0101	2.54	2.74	2.94	V	
			0110	2.62	2.87	3.12	V	
			0111	2.76	3.01	3.26	V	
			1000	3.00	3.30	3.60	V	
			1001	3.18	3.48	3.78	V	
			1010	3.44	3.69	3.94	V	
			1011	3.66	3.91	4.16	V	
			1100	3.90	4.15	4.40	V	
			1101	4.11	4.41	4.71	V	
			1110	4.39	4.74	5.09	V	
			1111	V(HLVDIN pin)			v	

† Production tested at $T_{\text{AMB}} = 25^{\circ}\text{C}$. Specifications over temperature limits ensured by characterization.

PIC18(L)F2X/4XK22

FIGURE 28-24: PIC18LF2X/4XK22 I_{DD}: RC_RUN MF-INTOSC 500 kHz

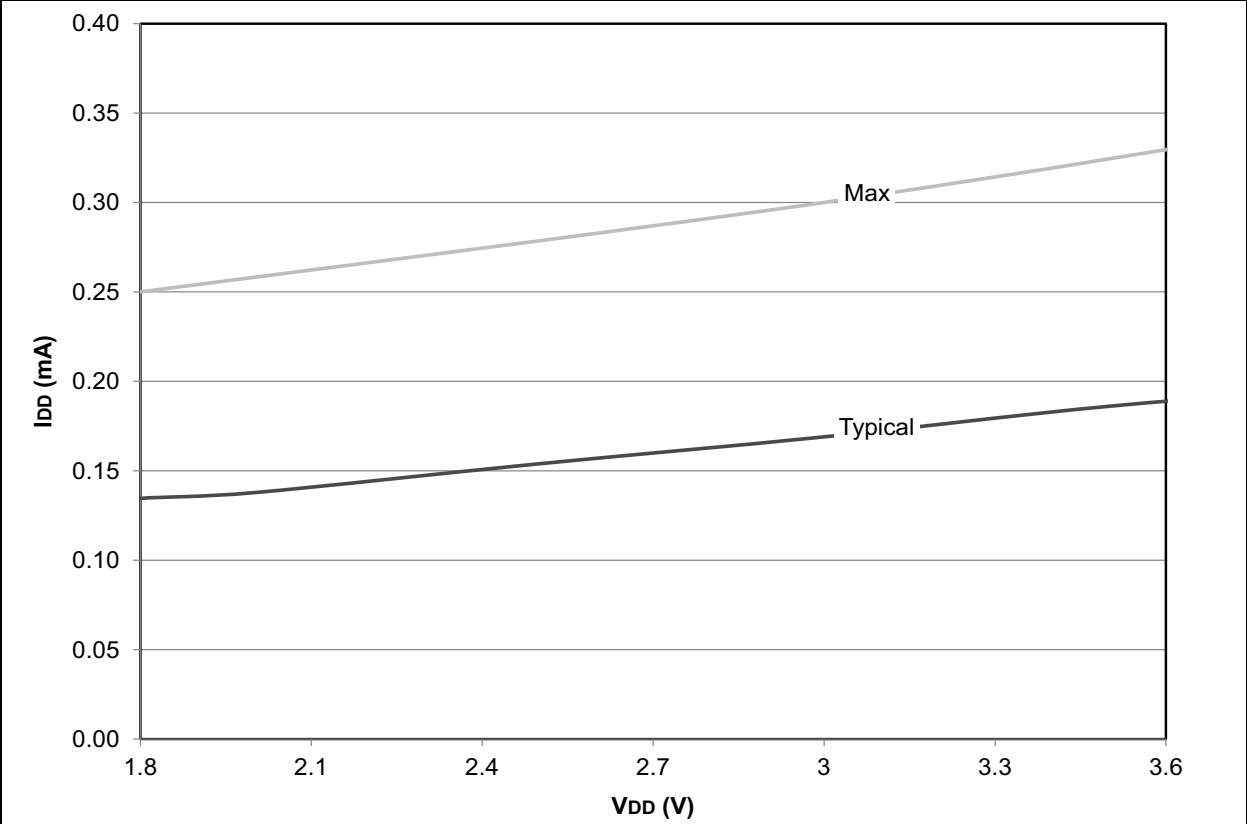
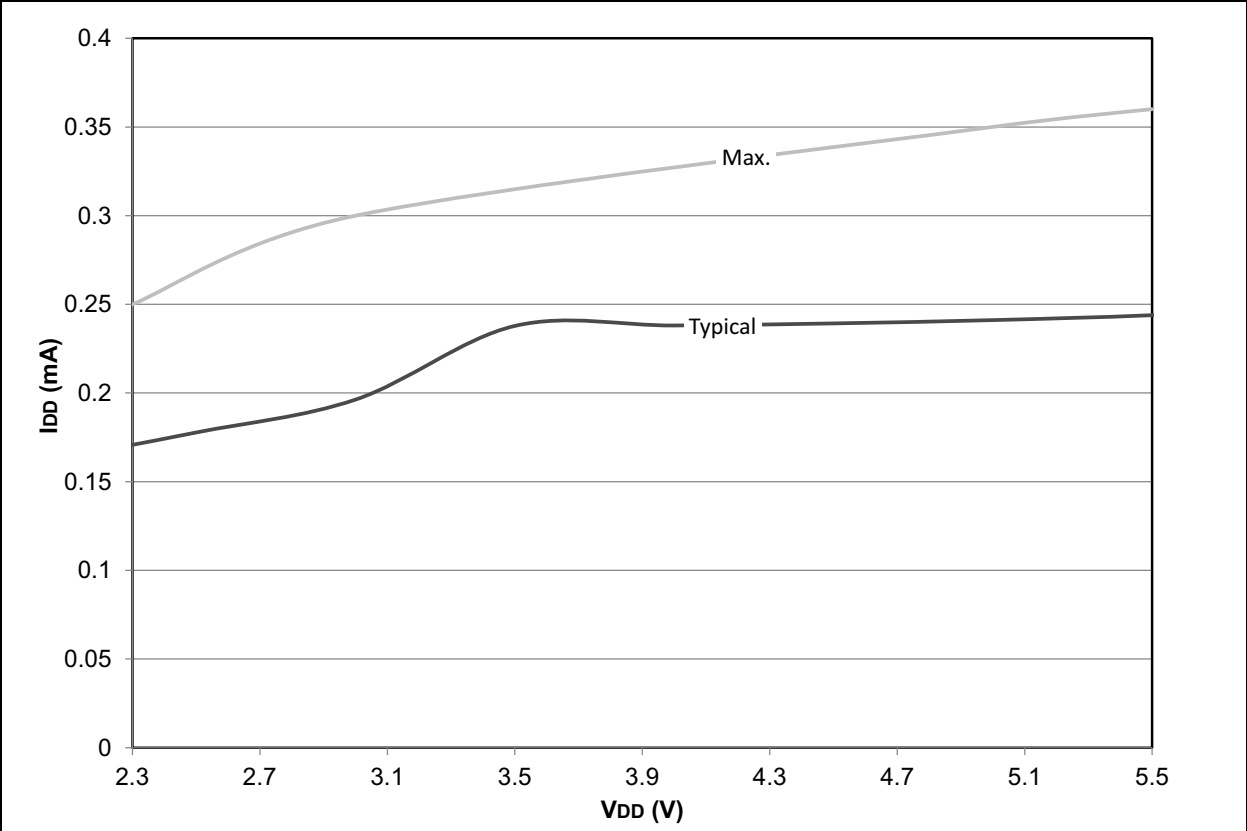


FIGURE 28-25: PIC18F2X/4XK22 I_{DD}: RC_RUN MF-INTOSC 500 kHz



PIC18(L)F2X/4XK22

FIGURE 28-68: PIC18LF2X/4XK22 TYPICAL I_{DD} : PRI_IDLE EC with PLL

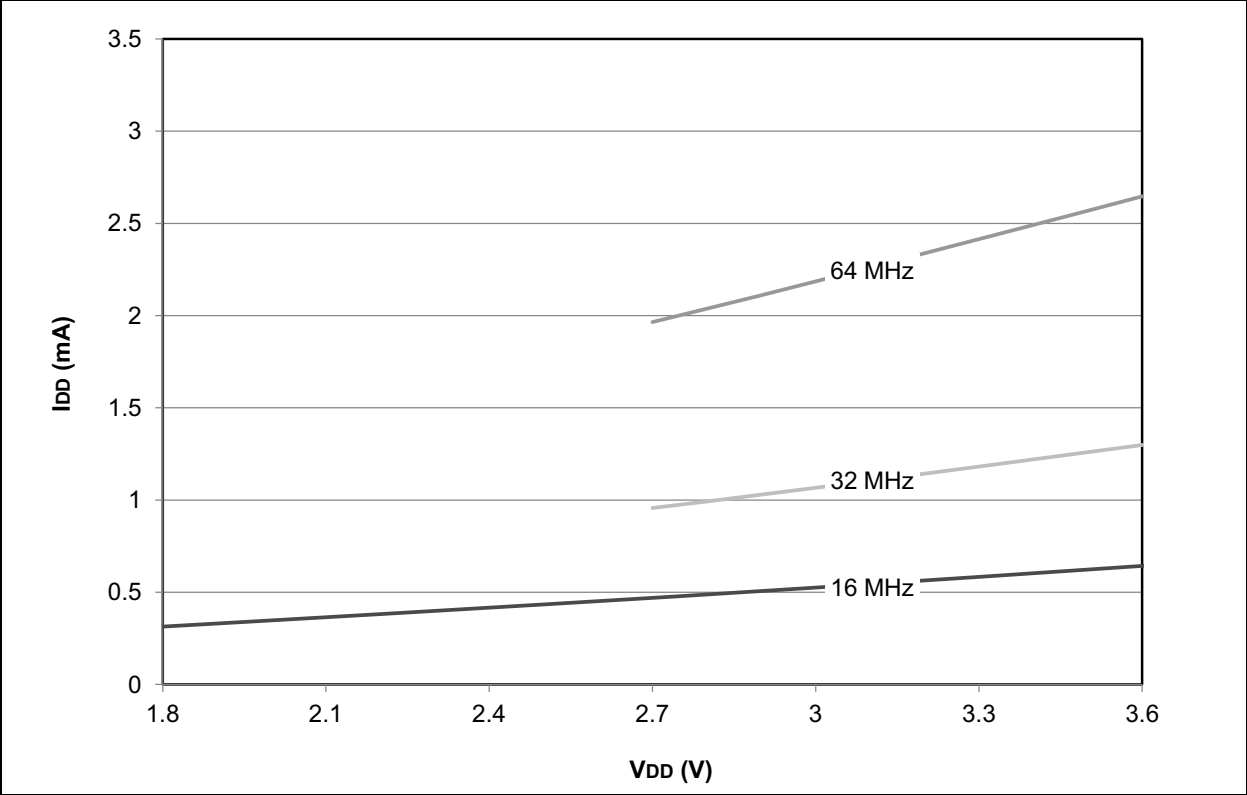


FIGURE 28-69: PIC18LF2X/4XK22 MAXIMUM I_{DD} : PRI_IDLE EC with PLL

