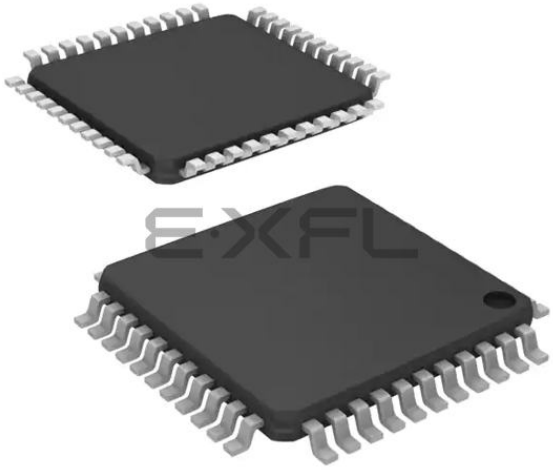


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic18lf44k22t-i-pt

TABLE 1-2: PIC18(L)F2XK22 PINOUT I/O DESCRIPTIONS

Pin Number		Pin Name	Pin Type	Buffer Type	Description
PDIP, SOIC	QFN, UQFN				
2	27	RA0/C12IN0-/AN0			
		RA0	I/O	TTL	Digital I/O.
		C12IN0-	I	Analog	Comparators C1 and C2 inverting input.
		AN0	I	Analog	Analog input 0.
3	28	RA1/C12IN1-/AN1			
		RA1	I/O	TTL	Digital I/O.
		C12IN1-	I	Analog	Comparators C1 and C2 inverting input.
		AN1	I	Analog	Analog input 1.
4	1	RA2/C2IN+/AN2/DACOUT/V_{REF-}			
		RA2	I/O	TTL	Digital I/O.
		C2IN+	I	Analog	Comparator C2 non-inverting input.
		AN2	I	Analog	Analog input 2.
		DACOUT	O	Analog	DAC Reference output.
		V_{REF-}	I	Analog	A/D reference voltage (low) input.
5	2	RA3/C1IN+/AN3/V_{REF+}			
		RA3	I/O	TTL	Digital I/O.
		C1IN+	I	Analog	Comparator C1 non-inverting input.
		AN3	I	Analog	Analog input 3.
		V_{REF+}	I	Analog	A/D reference voltage (high) input.
6	3	RA4/CCP5/C1OUT/SRQ/T0CKI			
		RA4	I/O	ST	Digital I/O.
		CCP5	I/O	ST	Capture 5 input/Compare 5 output/PWM 5 output.
		C1OUT	O	CMOS	Comparator C1 output.
		SRQ	O	TTL	SR latch Q output.
		T0CKI	I	ST	Timer0 external clock input.
7	4	RA5/C2OUT/SRNQ/$\overline{SS1}$/HLVDIN/AN4			
		RA5	I/O	TTL	Digital I/O.
		C2OUT	O	CMOS	Comparator C2 output.
		SRNQ	O	TTL	SR latch $\overline{Q}$ output.
		$\overline{SS1}$	I	TTL	SPI slave select input (MSSP).
		HLVDIN	I	Analog	High/Low-Voltage Detect input.
		AN4	I	Analog	Analog input 4.
10	7	RA6/CLKO/OSC2			
		RA6	I/O	TTL	Digital I/O.
		CLKO	O		In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
		OSC2	O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

Note 1: Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

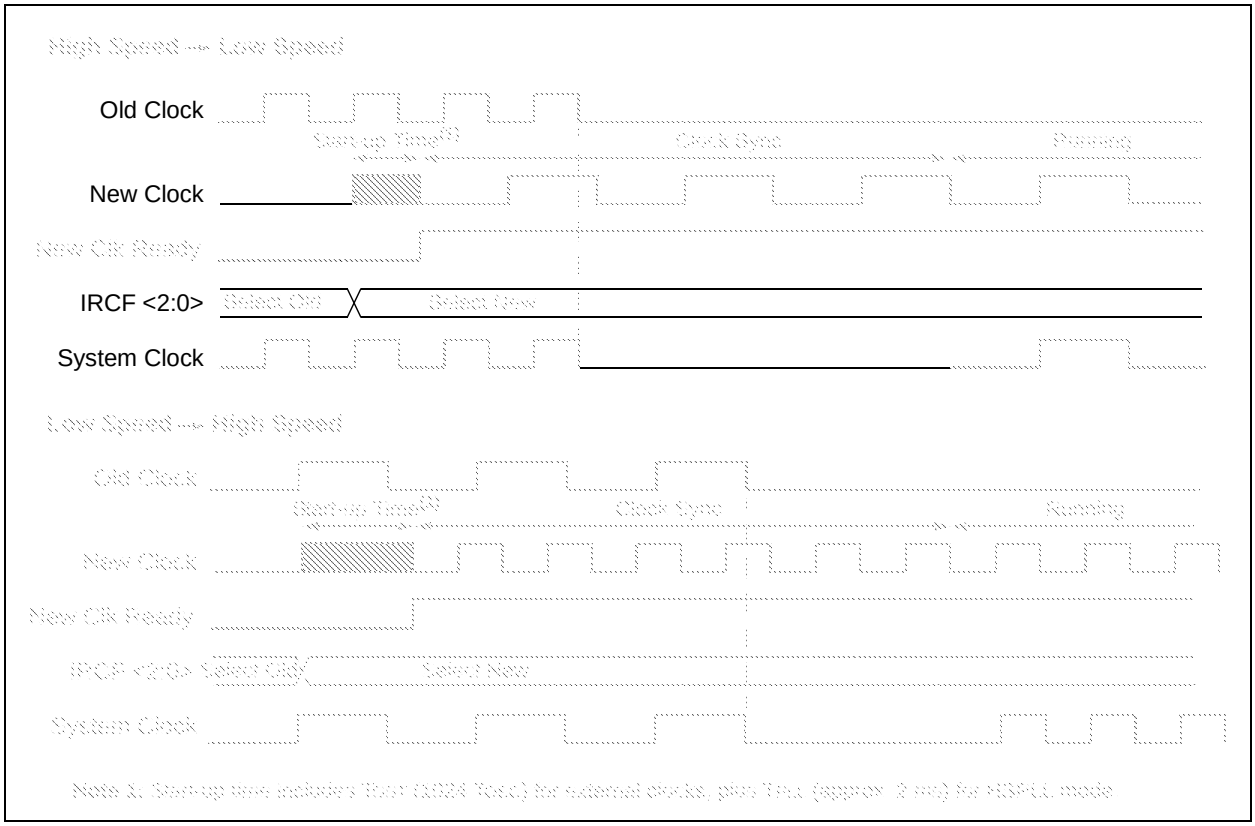
2.12.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin executing by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 external clock cycles.
4. OST timed out. External clock is ready.
5. OSTS is set.
6. Clock switch finishes according to Figure 2-9

2.12.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in CONFIG1H Configuration register, or the internal oscillator. OSTS = 0 when the external oscillator is not ready, which indicates that the system is running from the internal oscillator.

FIGURE 2-9: CLOCK SWITCH TIMING



7.6 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

The microcontroller itself can both read and write to the internal data EEPROM, regardless of the state of the code-protect Configuration bit. Refer to **Section 24.0 "Special Features of the CPU"** for additional information.

7.7 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (TPWRT).The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

7.8 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM without exceeding the total number of write cycles to a single byte. Refer to the Data EEPROM Memory parameters in **Section 27.0 "Electrical Specifications"** for write cycle limits. If this is the case, then an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

Note:	If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification.

EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR               ; Start at address 0
        CLRF    EEADRH              ; if > 256 bytes EEPROM
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   0AAh                ;
        MOVWF   EECON2              ; Write 0AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     LOOP                ; Not zero, do it again
        INCFSZ  EEADRH, F           ; if > 256 bytes, Increment address
        BRA     LOOP                ; if > 256 bytes, Not zero, do it again

        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

TABLE 10-8: PORTC I/O SUMMARY

Pin Name	Function	TRIS Setting	ANSEL setting	Pin Type	Buffer Type	Description
RC0/P2B/T3CKI/T3G/ T1CKI/SOSCO	RC0	0	—	O	DIG	LATC<0> data output; not affected by analog input.
		1	—	I	ST	PORTC<0> data input; disabled when analog input enabled.
	P2B(2)	0	—	O	DIG	Enhanced CCP2 PWM output 2.
	T3CKI(1)	1	—	I	ST	Timer3 clock input.
	T3G	1	—	I	ST	Timer3 external clock gate input.
	T1CKI	1	—	I	ST	Timer1 clock input.
	SOSCO	x	—	O	XTAL	Secondary oscillator output.
RC1/P2A/CCP2/SOSCI	RC1	0	—	O	DIG	LATC<1> data output; not affected by analog input.
		1	—	I	ST	PORTC<1> data input; disabled when analog input enabled.
	P2A	0	—	O	DIG	Enhanced CCP2 PWM output 1.
	CCP2(1)	0	—	O	DIG	Compare 2 output/PWM 2 output.
		1	—	I	ST	Capture 2 input.
	SOSCI	x	—	I	XTAL	Secondary oscillator input.
RC2/CTPLS/P1A/ CCP1/T5CKI/AN14	RC2	0	0	O	DIG	LATC<2> data output; not affected by analog input.
		1	0	I	ST	PORTC<2> data input; disabled when analog input enabled.
	CTPLS	0	0	O	DIG	CTMU pulse generator output.
	P1A	0	0	O	DIG	Enhanced CCP1 PWM output 1.
	CCP1	0	0	O	DIG	Compare 1 output/PWM 1 output.
		1	0	I	ST	Capture 1 input.
	T5CKI	1	0	I	ST	Timer5 clock input.
	AN14	1	1	I	AN	Analog input 14.
RC3/SCK1/SCL1/AN15	RC3	0	0	O	DIG	LATC<3> data output; not affected by analog input.
		1	0	I	ST	PORTC<3> data input; disabled when analog input enabled.
	SCK1	0	0	O	DIG	MSSP1 SPI Clock output.
		1	0	I	ST	MSSP1 SPI Clock input.
	SCL1	0	0	O	DIG	MSSP1 I^2C Clock output.
		1	0	I	I^2C	MSSP1 I^2C Clock input.
	AN15	1	1	I	AN	Analog input 15.
RC4/SDI1/SDA1/AN16	RC4	0	0	O	DIG	LATC<4> data output; not affected by analog input.
		1	0	I	ST	PORTC<4> data input; disabled when analog input enabled.
	SDI1	1	0	I	ST	MSSP1 SPI data input.
	SDA1	0	0	O	DIG	MSSP1 I^2C data output.
		1	0	I	I^2C	MSSP1 I^2C data input.
	AN16	1	1	I	AN	Analog input 16.

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; I^2C = Schmitt Trigger input with I^2C.

Note 1: Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

3: Function on PORTD and PORTE for PIC18(L)F4XK22 devices.

10.7 Port Analog Control

Most port pins are multiplexed with analog functions such as the Analog-to-Digital Converter and comparators. When these I/O pins are to be used as analog inputs it is necessary to disable the digital input buffer to avoid excessive current caused by improper biasing of the digital input. Individual control of the digital input buffers on pins which share analog functions is provided by the ANSELA, ANSELB, ANSELC, ANSELD and ANSELE registers. Setting an ANSx bit high will disable the associated digital input buffer and cause all reads of that pin to return '0' while allowing analog functions of that pin to operate correctly.

The state of the ANSx bits has no affect on digital output functions. A pin with the associated TRISx bit clear and ANSx bit set will still operate as a digital output but the input mode will be analog. This can cause unexpected behavior when performing read-modify-write operations on the affected port.

All ANSEL register bits default to '1' upon POR and BOR, disabling digital inputs for their associated port pins. All TRIS register bits default to '1' upon POR or BOR, disabling digital outputs for their associated port pins. As a result, all port pins that have an ANSEL register will default to analog inputs upon POR or BOR.

10.8 Port Slew Rate Control

The output slew rate of each port is programmable to select either the standard transition rate or a reduced transition rate of approximately 0.1 times the standard to minimize EMI. The reduced transition time is the default slew rate for all ports.

10.9 Register Definitions – Port Control

REGISTER 10-1: PORTX(1): PORTx REGISTER

R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x
Rx7	Rx6	Rx5	Rx4	Rx3	Rx2	Rx1	Rx0
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	x = Bit is unknown
-n/n = Value at POR and BOR/Value at all other Resets		

bit 7-0 **Rx<7:0>:** PORTx I/O bit values(2)

Note 1: Register Description for PORTA, PORTB, PORTC and PORTD.

2: Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

REGISTER 10-13: IOCB: INTERRUPT-ON-CHANGE PORTB CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—
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 7-4 **IOCB<7:4>:** Interrupt-on-Change PORTB control bits
1 = Interrupt-on-change enabled(1)
0 = Interrupt-on-change disabled

Note 1: Interrupt-on-change requires that the RBIE bit (INTCON<3>) is set.

REGISTER 10-14: SLRCON: SLEW RATE CONTROL REGISTER

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SLRE(1)	SLRD(1)	SLRC	SLRB	SLRA
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

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

bit 4 **SLRE:** PORTE Slew Rate Control bit(1)
1 = All outputs on PORTE slew at a limited rate
0 = All outputs on PORTE slew at the standard rate

bit 3 **SLRD:** PORTD Slew Rate Control bit(1)
1 = All outputs on PORTD slew at a limited rate
0 = All outputs on PORTD slew at the standard rate

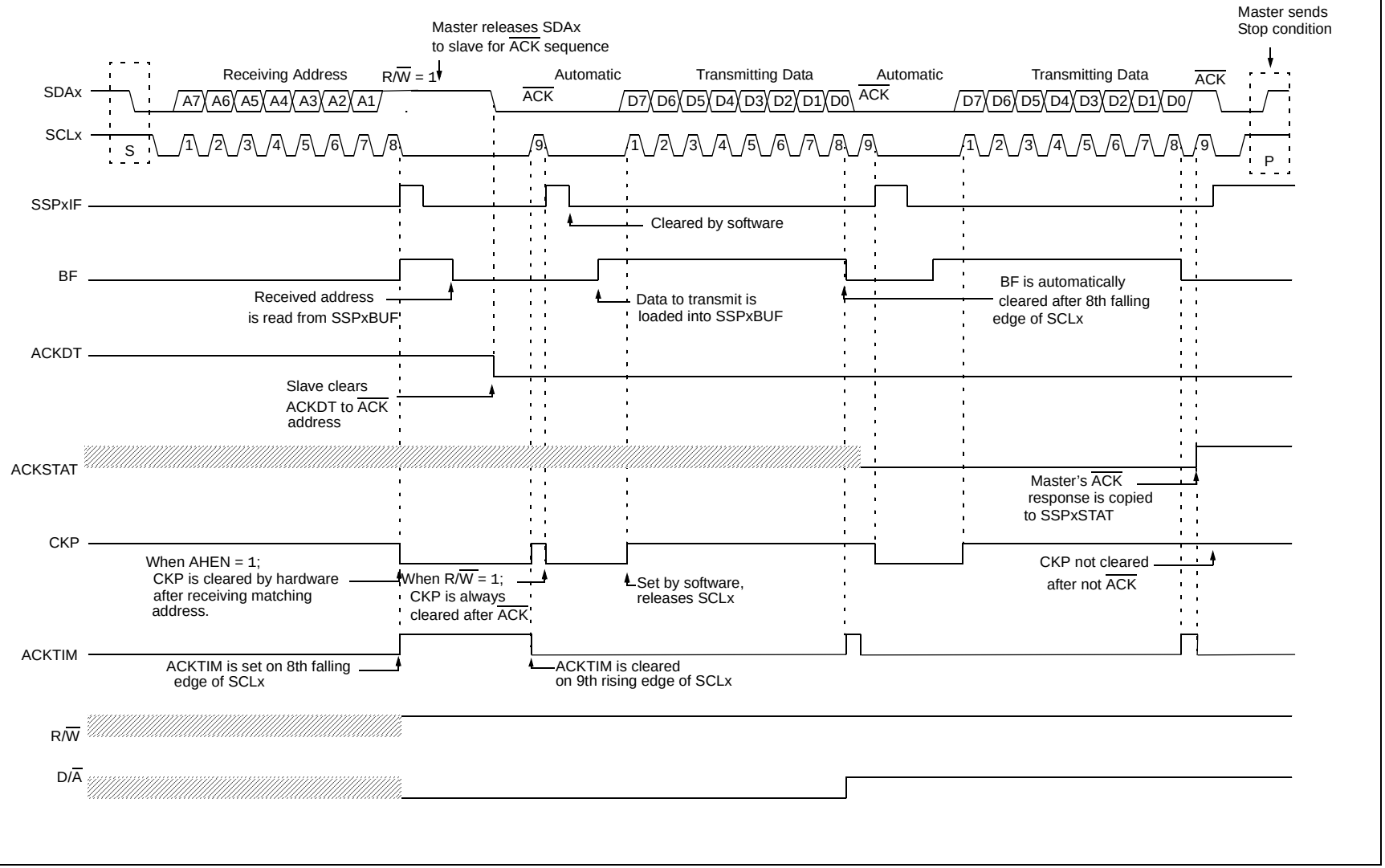
bit 2 **SLRC:** PORTC Slew Rate Control bit
1 = All outputs on PORTC slew at a limited rate
0 = All outputs on PORTC slew at the standard rate

bit 1 **SLRB:** PORTB Slew Rate Control bit
1 = All outputs on PORTB slew at a limited rate
0 = All outputs on PORTB slew at the standard rate

bit 0 **SLRA:** PORTA Slew Rate Control bit
1 = All outputs on PORTA slew at a limited rate(2)
0 = All outputs on PORTA slew at the standard rate

Note 1: These bits are available on PIC18(L)F4XK22 devices.
2: The slew rate of RA6 defaults to standard rate when the pin is used as CLKOUT.

FIGURE 15-19: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 1)



16.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (HFINTOSC). However, the HFINTOSC frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the HFINTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 2.6 "Internal Clock Modes"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 16.4.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

16.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 16.5.1.6 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREGx register. If the RCxIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE/GIEH bit is also set, the program will branch to the interrupt vector.

16.5.2.4 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Set the RXx/DTx and TXx/CKx TRIS controls to '1'.
3. If using interrupts, ensure that the GIE/GIEH and PEIE/GIEL bits of the INTCON register are set and set the RCxIE bit.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCxIF bit will be set when reception is complete. An interrupt will be generated if the RCxIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTAx register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREGx register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

TABLE 16-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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 slave reception.

17.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

17.2.3 DISCHARGE

The discharge phase is used to initialize the value of the capacitor array. The array is discharged after every sample. This feature helps to optimize the unity-gain amplifier, as the circuit always needs to charge the capacitor array, rather than charge/discharge based on previous measure values.

17.2.4 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared by software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

17.2.5 DELAY BETWEEN CONVERSIONS

After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, the currently selected channel is reconnected to the charge holding capacitor commencing the next acquisition.

17.2.6 ADC OPERATION IN POWER-MANAGED MODES

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT<2:0> and ADCS<2:0> bits in ADCON2 should be updated in accordance with the clock source to be used in that mode. After entering the mode, an A/D acquisition or conversion may be started. Once started, the device should continue to be clocked by the same clock source until the conversion has been completed.

If desired, the device may be placed into the corresponding Idle mode during the conversion. If the device clock frequency is less than 1 MHz, the A/D FRC clock source should be selected.

17.2.7 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

17.2.8 SPECIAL EVENT TRIGGER

Two Special Event Triggers are available to start an A/D conversion: CTMU and CCP5. The Special Event Trigger source is selected using the TRIGSEL bit in ADCON1.

When TRIGSEL = 0, the CCP5 module is selected as the Special Event Trigger source. To enable the Special Event Trigger in the CCP module, set CCP5M<3:0> = 1011, in the CCP5CON register.

When TRIGSEL = 1, the CTMU module is selected. The CTMU module requires that the CTTRIG bit in CTMUCONH is set to enable the Special Event Trigger.

In addition to TRIGSEL bit, the following steps are required to start an A/D conversion:

- The A/D module must be enabled (ADON = 1)
- The appropriate analog input channel selected
- The minimum acquisition period set one of these ways:
  - Timing provided by the user
  - Selection made of an appropriate TACQ time

With these conditions met, the trigger sets the GO/$\overline{\text{DONE}}$ bit and the A/D acquisition starts.

If the A/D module is not enabled (ADON = 0), the module ignores the Special Event Trigger.

17.2.9 PERIPHERAL MODULE DISABLE

When a peripheral module is not used or inactive, the module can be disabled by setting the Module Disable bit in the PMD registers. This will reduce power consumption to an absolute minimum. Setting the PMD bits holds the module in Reset and disconnects the module's clock source. The Module Disable bit for the ADC module is ADCMD in the PMD2 Register. See **Section 3.0 "Power-Managed Modes"** for more information.

REGISTER 20-2: SRCON1: SR LATCH CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **SRSPE:** SR Latch Peripheral Set Enable bit
1 = SRI pin status sets SR latch
0 = SRI pin status has no effect on SR latch

bit 6 **SRSCKE:** SR Latch Set Clock Enable bit
1 = Set input of SR latch is pulsed with DIVSRCLK
0 = Set input of SR latch is not pulsed with DIVSRCLK

bit 5 **SRSC2E:** SR Latch C2 Set Enable bit
1 = C2 Comparator output sets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 4 **SRSC1E:** SR Latch C1 Set Enable bit
1 = C1 Comparator output sets SR latch
0 = C1 Comparator output has no effect on SR latch

bit 3 **SRRPE:** SR Latch Peripheral Reset Enable bit
1 = SRI pin resets SR latch
0 = SRI pin has no effect on SR latch

bit 2 **SRRCKE:** SR Latch Reset Clock Enable bit
1 = Reset input of SR latch is pulsed with DIVSRCLK
0 = Reset input of SR latch is not pulsed with DIVSRCLK

bit 1 **SRRC2E:** SR Latch C2 Reset Enable bit
1 = C2 Comparator output resets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 0 **SRRC1E:** SR Latch C1 Reset Enable bit
1 = C1 Comparator output resets SR latch
0 = C1 Comparator output has no effect on SR latch

TABLE 20-2: REGISTERS ASSOCIATED WITH THE SR LATCH

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
SRCON0	SRLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR	329
SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E	330
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	152

Legend: Shaded bits are not used with this module.

22.7 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the VREFCON1 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

22.8 Effects of a Reset

A device Reset affects the following:

- DAC is disabled
- DAC output voltage is removed from the DACOUT pin
- The DACR<4:0> range select bits are cleared

22.9 Register Definitions: DAC Control

REGISTER 22-1: VREFCON1: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0
DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS
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	

bit 7 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **DACLPS:** DAC Low-Power Voltage Source Select bit
1 = DAC Positive reference source selected
0 = DAC Negative reference source selected

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC voltage level is also an output on the DACOUT pin
0 = DAC voltage level is disconnected from the DACOUT pin

bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
00 = V_{DD}
01 = $V_{REF}+$
10 = FVR BUF1 output
11 = Reserved, do not use

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **DACNSS:** DAC Negative Source Select bits
1 = $V_{REF}-$
0 = V_{SS}

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

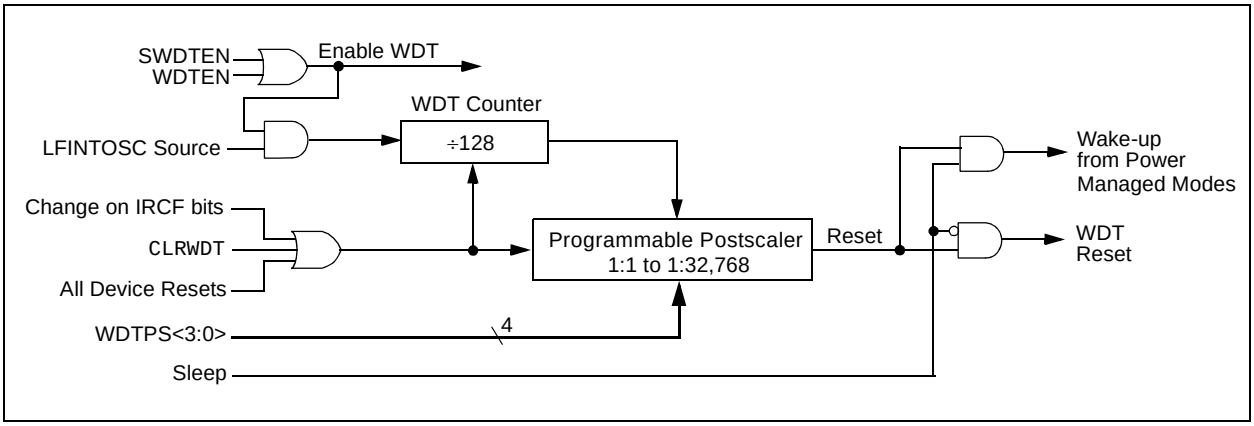
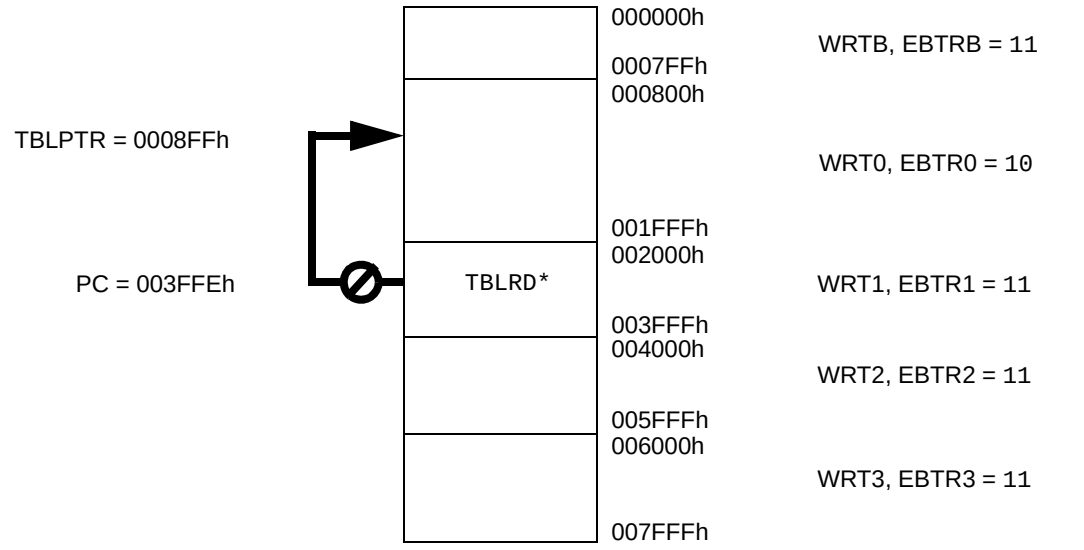
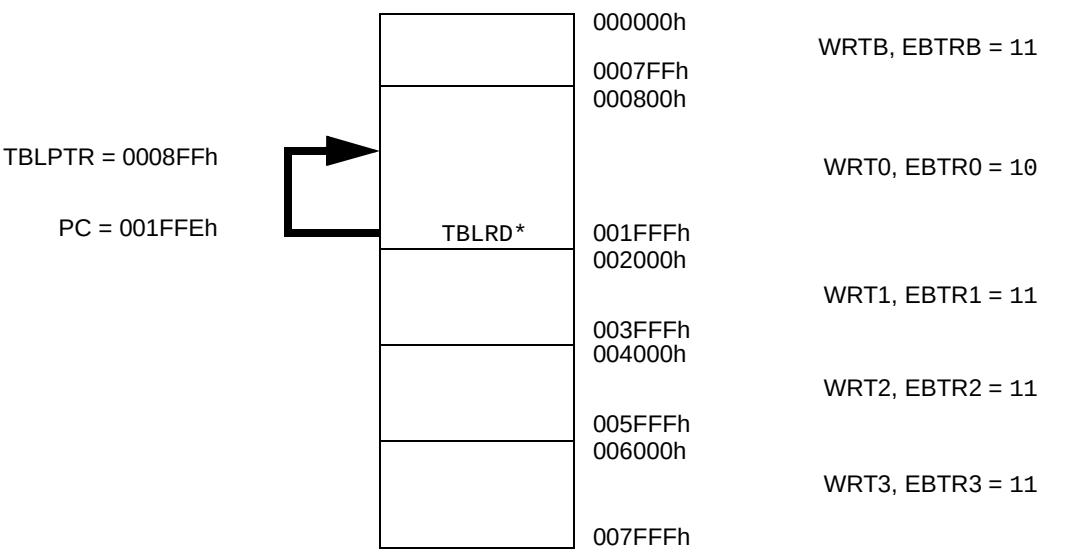


FIGURE 24-4: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED



Results: All table reads from external blocks to Blockn are disabled whenever EBTRn = 0.
TABLAT register returns a value of '0'.

FIGURE 24-5: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



Results: Table reads permitted within Blockn, even when EBTRBn = 0.
TABLAT register returns the value of the data at the location TBLPTR.

TBLWT **Table Write**

Syntax: TBLWT (*; *+; *-; +*)

Operands: None

Operation: if TBLWT*,
(TABLAT) → Holding Register;
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register;
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register;
(TBLPTR) – 1 → TBLPTR;
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register;

Status Affected: None

Encoding:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction uses the three LSBs of TBLPTR to determine which of the eight holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 "Flash Program Memory"** for additional details on programming Flash memory.)
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-MByte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word
TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

TBLWT **Table Write (Continued)**

Example1: TBLWT *+;

Before Instruction
TABLAT = 55h
TBLPTR = 00A356h
HOLDING REGISTER (00A356h) = FFh
After Instructions (table write completion)
TABLAT = 55h
TBLPTR = 00A357h
HOLDING REGISTER (00A356h) = 55h

Example 2: TBLWT +*;

Before Instruction
TABLAT = 34h
TBLPTR = 01389Ah
HOLDING REGISTER (01389Ah) = FFh
HOLDING REGISTER (01389Bh) = FFh
After Instruction (table write completion)
TABLAT = 34h
TBLPTR = 01389Bh
HOLDING REGISTER (01389Ah) = FFh
HOLDING REGISTER (01389Bh) = 34h

26.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

26.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

FIGURE 27-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)

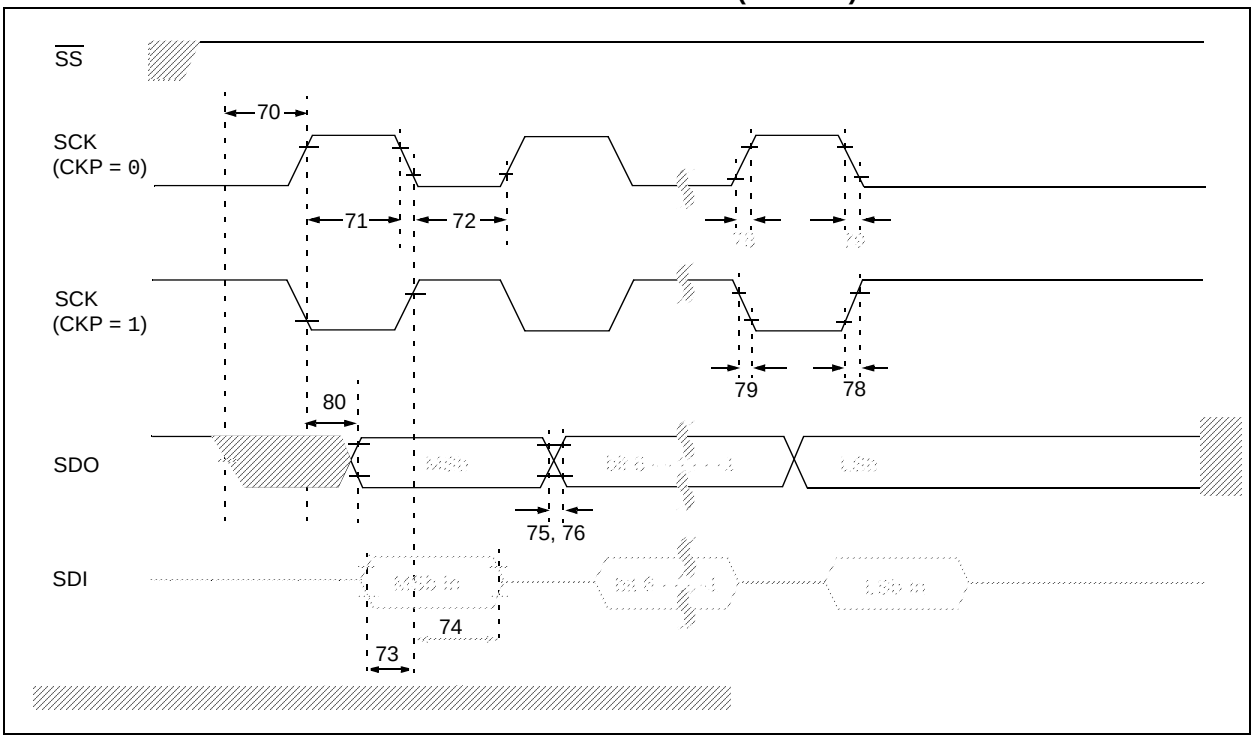
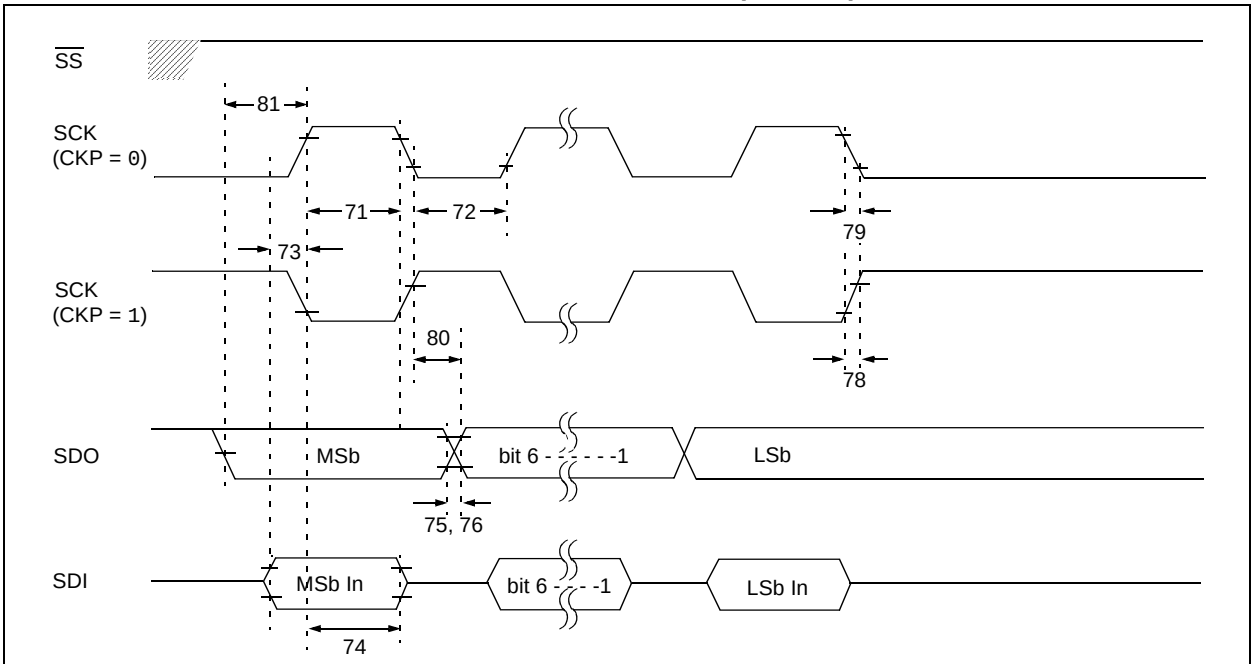


FIGURE 27-14: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



Note: Refer to Figure 27-6 for load conditions.

TABLE 27-18: MASTER SSP I^2C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
101	TLOW	Clock Low Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
102	TR	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	300	ns	
103	TF	SDA and SCL Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	100	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode(1)	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	ms	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	ms	
D102	CB	Bus Capacitive Loading		—	400	pF	

Note 1: Maximum pin capacitance = 10 pF for all I^2C pins.

2: A fast mode I^2C bus device can be used in a standard mode I^2C bus system, but parameter 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the Low period of the SCL signal. If such a device does stretch the Low period of the SCL signal, it must output the next data bit to the SDA line, parameter 102 + parameter 107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCL line is released.

FIGURE 28-36: PIC18F2X/4XK22 TYPICAL IDD: RC_IDLE LF-INTOSC 31 kHz

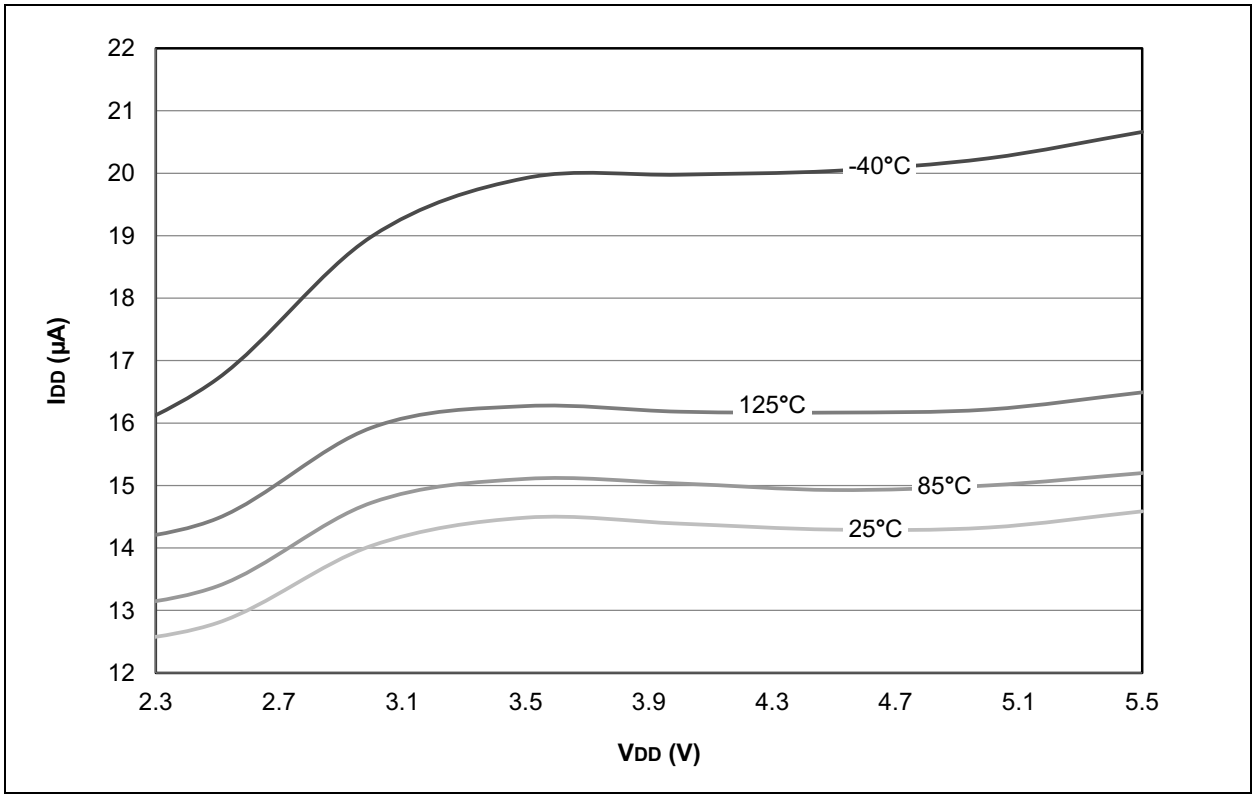


FIGURE 28-37: PIC18F2X/4XK22 MAXIMUM IDD: RC_IDLE LF-INTOSC 31 kHz