



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	40MHz
Connectivity	EBI/EMI, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	70
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f8622t-i-pt

PIC18F8722 FAMILY

TABLE 2-2: CAPACITOR SELECTION FOR QUARTZ CRYSTALS

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
LP	32 kHz	22 pF	22 pF
XT	1 MHz	22 pF	22 pF
	4 MHz	22 pF	22 pF
HS	4 MHz	22 pF	22 pF
	10 MHz	22 pF	22 pF
	20 MHz	22 pF	22 pF
	25 MHz	22 pF	22 pF

Capacitor values are for design guidance only.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. Refer to the following application notes for oscillator specific information:

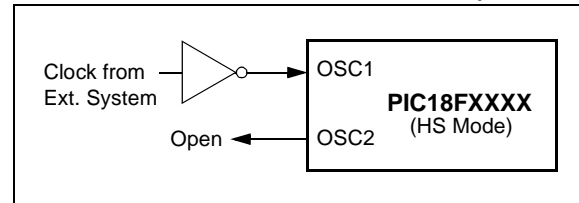
- AN588 – PIC® Microcontroller Oscillator Design Guide
- AN826 – Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices
- AN849 – Basic PIC® Oscillator Design
- AN943 – Practical PIC® Oscillator Analysis and Design
- AN949 – Making Your Oscillator Work

See the notes following this table for additional information.

- Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time.
- 2: When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.
 - 3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
 - 4: Rs may be required to avoid overdriving crystals with low drive level specification.
 - 5: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 2-2. When operated in this mode, parameters D033 and D043 apply.

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

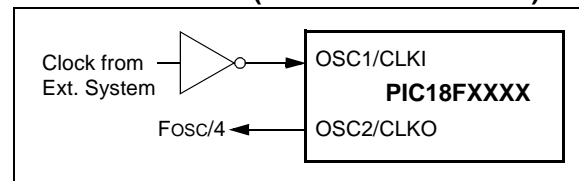


2.3 External Clock Input

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

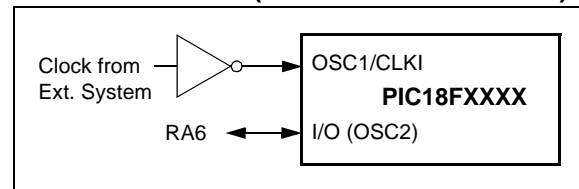
In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-3 shows the pin connections for the EC Oscillator mode.

FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)



The ECIO Oscillator mode functions like the EC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-4 shows the pin connections for the ECIO Oscillator mode. When operated in this mode, parameters D033A and D043A apply.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)



PIC18F8722 FAMILY

5.1.3.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. When STVREN is set, a full or underflow will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

5.1.4 FAST REGISTER STACK

A fast register stack is provided for the STATUS, WREG and BSR registers, to provide a “fast return” option for interrupts. The stack for each register is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the Stack registers. The values in the registers are then loaded back into their associated registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high-priority interrupts are enabled, the stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the fast register stack for returns from interrupt. If no interrupts are used, the fast register stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the fast register stack. A RETURN, FAST instruction is then executed to restore these registers from the fast register stack.

Example 5-1 shows a source code example that uses the fast register stack during a subroutine call and return.

EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                     ;SAVED IN FAST REGISTER
                     ;STACK
    •
    •
SUB1    •
    •
    RETURN, FAST      ;RESTORE VALUES SAVED
                     ;IN FAST REGISTER STACK
```

5.1.5 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

5.1.5.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 5-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions that returns the value ‘nn’ to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

Note: The “ADDWF PCL” instruction does not update the PCLATH and PCLATU registers. A read operation on PCL must be performed to update PCLATH and PCLATU.

EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

```
MAIN:  ORG      0x0000
        MOVLW   0x00
        CALL    TABLE
...
        ORG      0x8000
TABLE  MOVF     PCL, F           ; A simple read of PCL will update PCLATH, PCLATU
        RLNCF   W, W           ; Multiply by 2 to get correct offset in table
        ADDWF   PCL             ; Add the modified offset to force jump into table
        RETLW   'A'
        RETLW   'B'
        RETLW   'C'
        RETLW   'D'
        RETLW   'E'
        END
```

PIC18F8722 FAMILY

5.2.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSb = 0). To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSb will always read '0' (see **Section 5.1.2 "Program Counter"**).

Figure 5-5 shows an example of how instruction words are stored in the program memory.

The **CALL** and **GOTO** instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 5-5 shows how the instruction **GOTO 0006h** is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 26.0 "Instruction Set Summary"** provides further details of the instruction set.

FIGURE 5-5: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations →			Word Address		
			LSB = 1	LSB = 0	↓
Instruction 1: MOVLW	055h				000000h
					000002h
Instruction 2: GOTO	0006h				000004h
					000006h
Instruction 3: MOVFF	123h, 456h		0Fh	55h	000008h
			EFh	03h	00000Ah
			F0h	00h	00000Ch
			C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

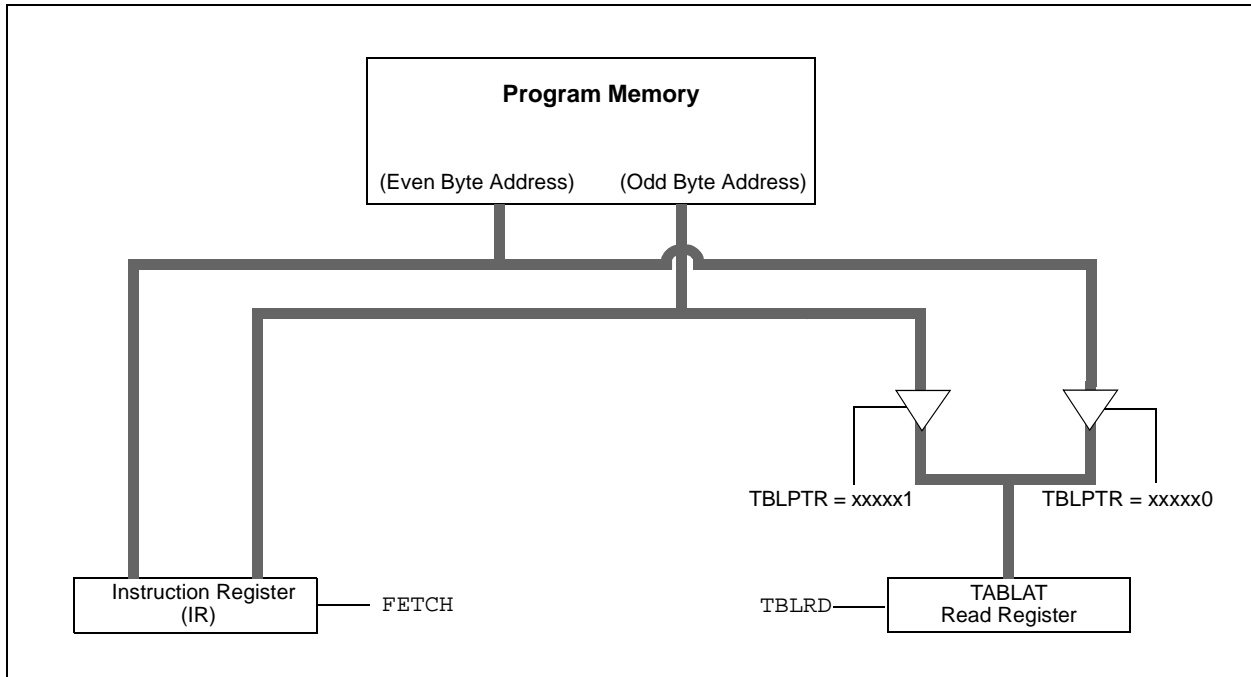
6.3 Reading the Flash Program Memory

The `TBLRD` instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing `TBLRD` places the byte pointed to into `TABLAT`. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 6-4 shows the interface between the internal program memory and the `TABLAT`.

FIGURE 6-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 6-1: READING A FLASH PROGRAM MEMORY WORD

```

MOV LW CODE_ADDR_UPPER      ; Load TBLPTR with the base
MOV WF TBLPTRU              ; address of the word
MOV LW CODE_ADDR_HIGH
MOV WF TBLPTRH
MOV LW CODE_ADDR_LOW
MOV WF TBLPTRL

READ_WORD
TBLRD*+                    ; read into TABLAT and increment
MOV F TABLAT, W            ; get data
MOV WF WORD_EVEN
TBLRD*+                    ; read into TABLAT and increment
MOV F TABLAT, W            ; get data
MOV F WORD_ODD

```

6.5 Writing to Flash Program Memory

The minimum programming block is 32 words or 64 bytes. Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

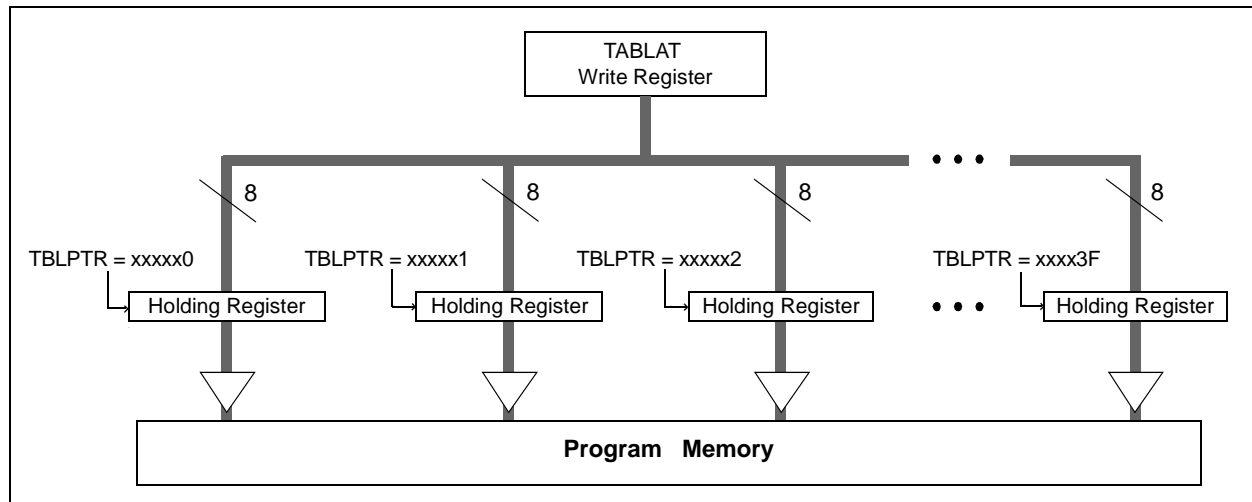
Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note: The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all 64 holding registers before executing a write operation.

FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY



6.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the row erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the EECON1 register for the write operation:
 - set EEPGD bit to point to program memory;
 - clear the CFGS bit to access program memory;
 - set WREN to enable byte writes.

8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write for T_{W} (see parameter D133A).
13. Re-enable interrupts.
14. Verify the memory (table read).

An example of the required code is shown in Example 6-3 on the following page.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

PROGRAM_MEMORY		BSF	EECON1, EEPGD	; point to Flash program memory
		BCF	EECON1, CFGS	; access Flash program memory
		BSF	EECON1, WREN	; enable write to memory
		BCF	INTCON, GIE	; disable interrupts
Required Sequence		MOVLW	55h	
		MOVWF	EECON2	; write 55h
		MOVLW	0AAh	
		MOVWF	EECON2	; write 0AAh
		BSF	EECON1, WR	; start program (CPU stall)
		BSF	INTCON, GIE	; re-enable interrupts
		BCF	EECON1, WREN	; disable write to memory

6.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

6.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

6.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 25.0 “Special Features of the CPU”** for more detail.

6.6 Flash Program Operation During Code Protection

See **Section 25.5 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

TABLE 6-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TBLPTRU	—	—	bit 21 ⁽¹⁾	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					57
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								57
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								57
TABLAT	Program Memory Table Latch								57
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
EECON2	EEPROM Control Register 2 (not a physical register)								59
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	59
IPR2	OSCFIP	CMIP	—	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	60
PIR2	OSCFIF	CMIF	—	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	60
PIE2	OSCFIE	CMIE	—	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	60

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used during Flash/EEPROM access.

Note 1: Bit 21 of TBLPTRU allows access to the device Configuration bits.

PIC18F8722 FAMILY

7.6 8-Bit Data Width Modes

In 8-Bit Data Width mode, the External Memory Bus operates only in Multiplexed mode; that is, data shares the 8 least significant bits of the address bus.

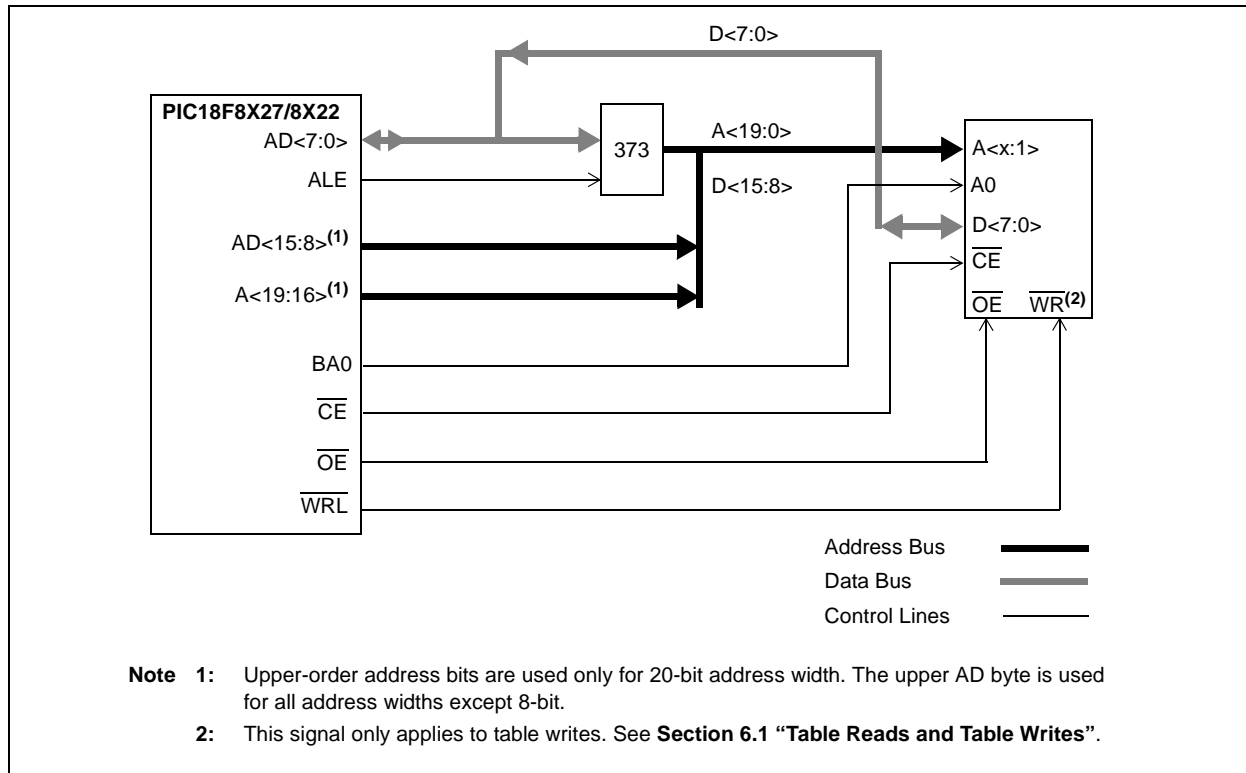
Figure 7-7 shows an example of 8-bit Multiplexed mode for PIC18F8527/8622/8627/8722 devices. This mode is used for a single 8-bit memory connected for 16-bit operation. The instructions will be fetched as two 8-bit bytes on a shared data/address bus. The two bytes are sequentially fetched within one instruction cycle (TCY). Therefore, the designer must choose external memory devices according to timing calculations based on 1/2 TCY (2 times the instruction rate). For proper memory speed selection, glue logic propagation delay times must be considered along with setup and hold times.

The Address Latch Enable (ALE) pin indicates that the address bits A<15:0> are available on the External Memory Interface bus. The Output Enable signal (\overline{OE}) will enable one byte of program memory for a portion of the instruction cycle, then BA0 will change and the second byte will be enabled to form the 16-bit instruction word. The least significant bit of the address, BA0, must be connected to the memory devices in this mode. The Chip Enable signal (\overline{CE}) is active at any time that the microcontroller accesses external memory, whether reading or writing; it is inactive (asserted high) whenever the device is in Sleep mode.

This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

The appropriate level of BA0 control line is strobed on the LSb of the TBLPTR.

FIGURE 7-7: 8-BIT MULTIPLEXED MODE EXAMPLE



PIC18F8722 FAMILY

15.1 Timer3 Operation

Timer3 can operate in one of three modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle ($F_{osc}/4$). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 15-1: TIMER3 BLOCK DIAGRAM

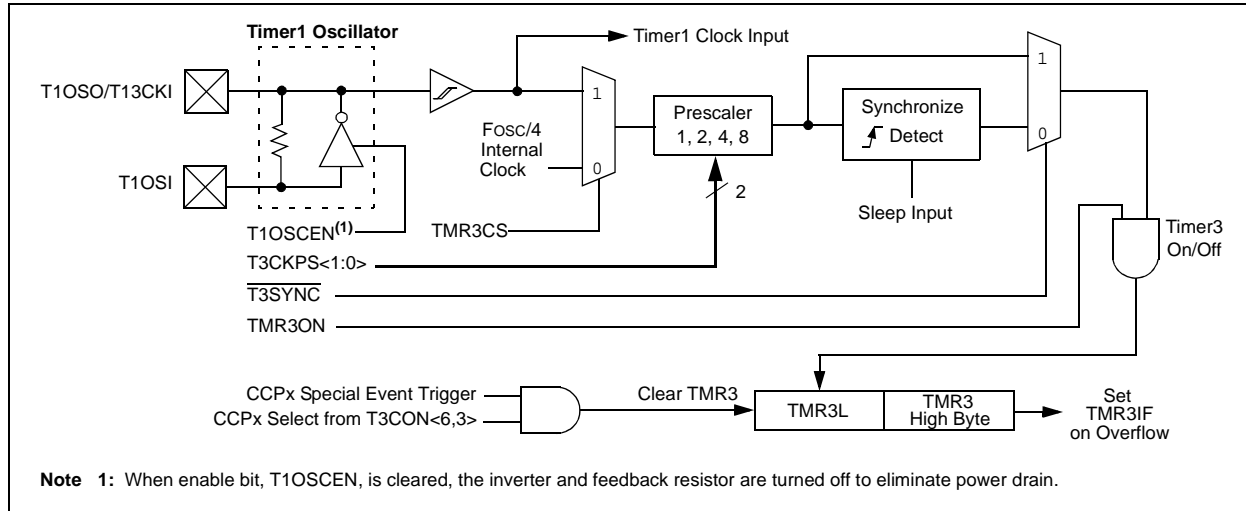
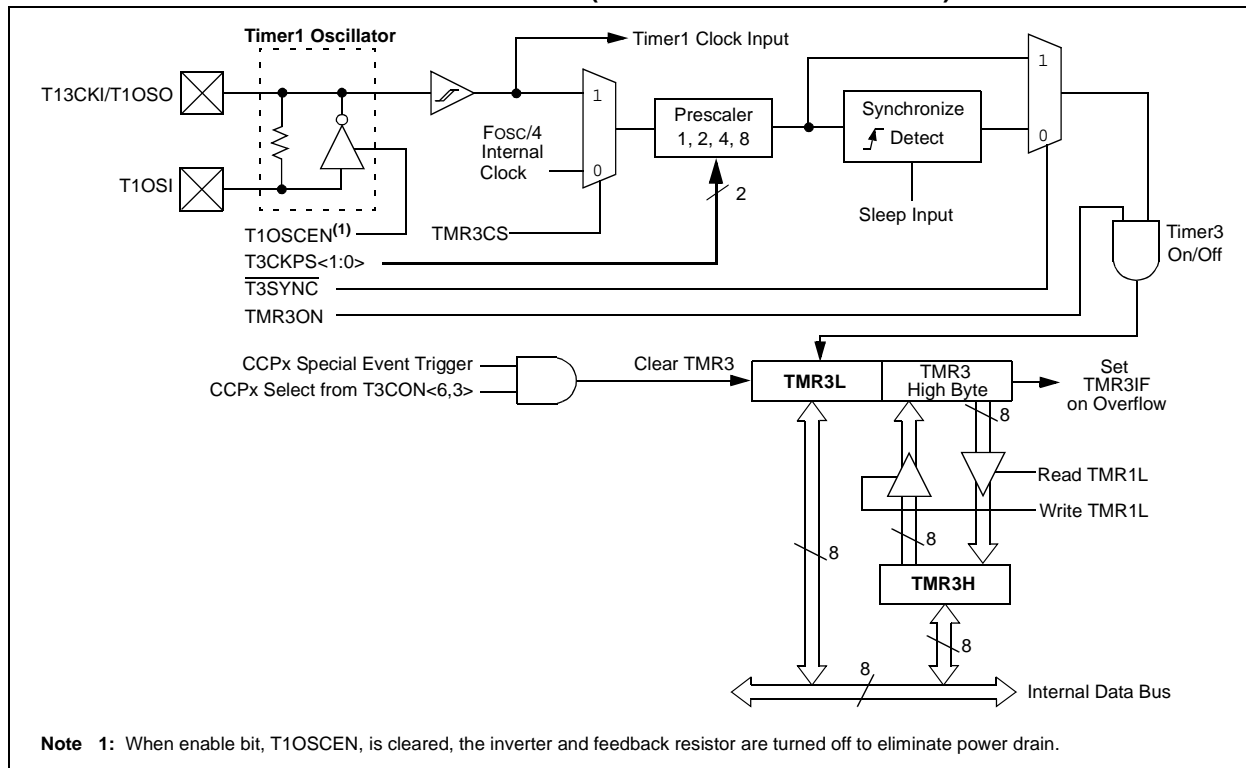


FIGURE 15-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



PIC18F8722 FAMILY

18.1 ECCP Outputs and Configuration

Each of the Enhanced CCP modules may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated PxA through PxD, are multiplexed with various I/O pins. Some ECCPx pin assignments are constant, while others change based on device configuration. For those pins that do change, the controlling bits are:

- CCP2MX Configuration bit (CONFIG3H<0>)
- ECCPMX Configuration bit (CONFIG3H<1>)
- Program Memory mode (set by Configuration bits, CONFIG3L<1:0>)

The pin assignments for the Enhanced CCP modules are summarized in Table 18-1, Table 18-2 and Table 18-3. To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the PxMx and CCPxMx bits (CCPxCON<7:6> and <3:0>, respectively). The appropriate TRIS direction bits for the corresponding port pins must also be set as outputs.

18.1.1 USE OF CCP4 AND CCP5 WITH ECCP1 AND ECCP3

Only the ECCP2 module has four dedicated output pins available for use. Assuming that the I/O ports or other multiplexed functions on those pins are not needed, they may be used whenever needed without interfering with any other CCP module.

ECCP1 and ECCP3, on the other hand, only have three dedicated output pins: ECCPx/P3A, PxB and PxC. Whenever these modules are configured for Quad PWM mode, the pin used for CCP4 or CCP5 takes priority over the D output pins for ECCP3 and ECCP1, respectively.

18.1.2 ECCP MODULE OUTPUTS, PROGRAM MEMORY MODES AND EMB ADDRESS BUS WIDTH

For PIC18F8527/8622/8627/8722 devices, the program memory mode of the device (**Section 7.2 “Address and Data Width”** and **Section 7.4 “Program Memory Modes and the External Memory Bus”**) impacts both pin multiplexing and the operation of the module.

The ECCP2 input/output (ECCP2/P2A) can be multiplexed to one of three pins. By default, this is RC1 for all devices; in this case, the default is in effect when CCP2MX is set and the device is operating in Microcontroller mode. With PIC18F8527/8622/8627/8722 devices, three other options exist. When CCP2MX is not set (= 0) and the device is in Microcontroller mode, ECCP2/P2A is multiplexed to RE7; in all other program memory modes, it is multiplexed to RB3.

Another option is for ECCPMX to be set while the device is operating in one of the three other program memory modes. In this case, ECCP1 and ECCP3 operate as compatible (i.e., single output) CCP modules. The pins used by their other outputs (PxB through PxD) are available for other multiplexed functions. ECCP2 continues to operate as an Enhanced CCP module regardless of the program memory mode.

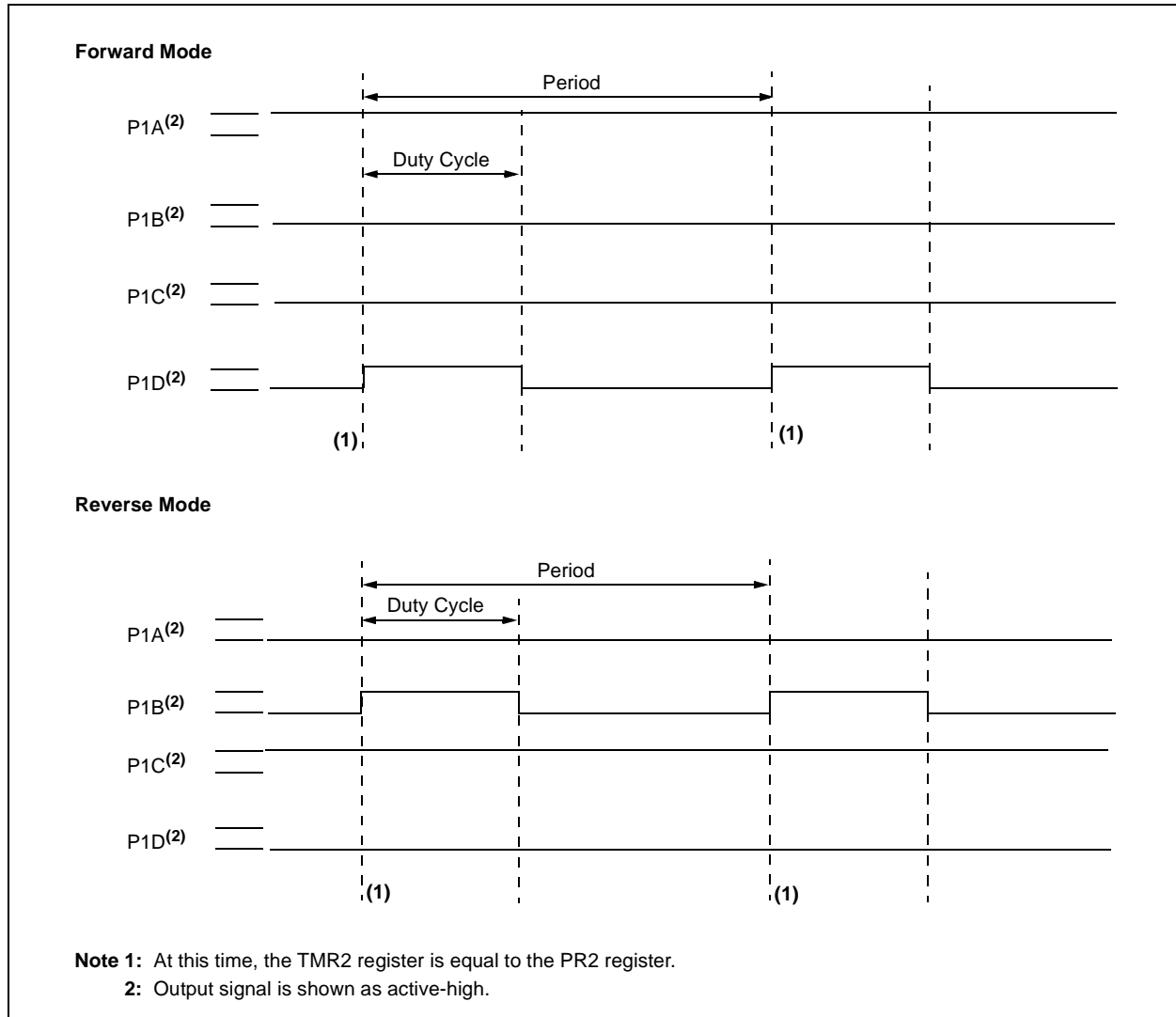
The final option is that the ABW<1:0> Configuration bits can be used to select 8, 12, 16 or 20-bit EMB addressing. Pins not assigned to EMB address pins are available for peripheral or port functions.

18.4.5 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin P1A is continuously active and pin P1D is modulated. In the Reverse mode, pin P1C is continuously active and pin P1B is modulated. These are illustrated in Figure 18-6.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORTC<2>, PORTE<6:5> and PORTG<4> data latches. Alternatively, P1B and P1C can be assigned to PORTH<7> and PORTH<6>, respectively, by programming the ECCPMX Configuration bit to '0'. See Table 18-1, Table 18-2 and Table 18-3 for more information. The associated bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 18-6: FULL-BRIDGE PWM OUTPUT



PIC18F8722 FAMILY

TABLE 18-5: REGISTERS ASSOCIATED WITH ECCP MODULES AND TIMER1 TO TIMER4

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
RCON	IPEN	SBOREN	—	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}	58
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	60
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	60
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	60
PIR2	OSCFIF	CMIF	—	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	60
PIE2	OSCFIE	CMIE	—	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	60
IPR2	OSCFIP	CMIP	—	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	60
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	60
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	60
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	60
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	60
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	60
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
TRISH ⁽²⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	60
TMR1L	Timer1 Register Low Byte								58
TMR1H	Timer1 Register High Byte								58
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	58
TMR2	Timer2 Register								58
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	58
PR2	Timer2 Period Register								58
TMR3L	Timer3 Register Low Byte								59
TMR3H	Timer3 Register High Byte								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	59
TMR4	Timer4 Register								61
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	61
PR4	Timer4 Period Register								61
CCPRxL ⁽¹⁾	Enhanced Capture/Compare/PWM Register x Low Byte								59, 61
CCPRxH ⁽¹⁾	Enhanced Capture/Compare/PWM Register x High Byte								59, 61
CCPxCON ⁽¹⁾	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	59
ECCPxAS ⁽¹⁾	ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0	59, 61
ECCPxDEL ⁽¹⁾	PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0	61

Legend: — = unimplemented, read as '0'. Shaded cells are not used during ECCP operation.

Note 1: Generic term for all of the identical registers of this name for all Enhanced CCP modules, where 'x' identifies the individual module (ECCP1, ECCP2 or ECCP3). Bit assignments and Reset values for all registers of the same generic name are identical.

2: This register is not implemented on PIC18F6527/6622/6627/6722 devices.

PIC18F8722 FAMILY

REGISTER 19-3: SSPxSTAT: MSSPx STATUS REGISTER (I²C™ MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ \overline{A}	P ⁽¹⁾	S ⁽¹⁾	R/ \overline{W} ^(2,3)	UA	BF
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 **SMP:** Slew Rate Control bit
In Master or Slave mode:
 1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
 0 = Slew rate control enabled for High-Speed mode (400 kHz)
- bit 6 **CKE:** SMBus Select bit
In Master or Slave mode:
 1 = Enable SMBus specific inputs
 0 = Disable SMBus specific inputs
- bit 5 **D/ \overline{A} :** Data/Address bit
In Master mode:
 Reserved.
In Slave mode:
 1 = Indicates that the last byte received or transmitted was data
 0 = Indicates that the last byte received or transmitted was address
- bit 4 **P:** Stop bit⁽¹⁾
 1 = Indicates that a Stop bit has been detected last
 0 = Stop bit was not detected last
- bit 3 **S:** Start bit⁽¹⁾
 1 = Indicates that a Start bit has been detected last
 0 = Start bit was not detected last
- bit 2 **R/ \overline{W} :** Read/Write Information bit^(2,3)
In Slave mode:
 1 = Read
 0 = Write
In Master mode:
 1 = Transmit is in progress
 0 = Transmit is not in progress
- bit 1 **UA:** Update Address bit (10-bit Slave mode only)
 1 = Indicates that the user needs to update the address in the SSPxADD register
 0 = Address does not need to be updated
- bit 0 **BF:** Buffer Full Status bit
In Transmit mode:
 1 = SSPxBUF is full
 0 = SSPxBUF is empty
In Receive mode:
 1 = SSPxBUF is full (does not include the \overline{ACK} and Stop bits)
 0 = SSPxBUF is empty (does not include the \overline{ACK} and Stop bits)

- Note 1:** This bit is cleared on Reset and when SSPEN is cleared.
- 2:** This bit holds the R/ \overline{W} bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not \overline{ACK} bit.
- 3:** ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.

PIC18F8722 FAMILY

TABLE 20-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	60
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	60
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	60
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
TXREGx	EUSARTx Transmit Register								59
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	59
BAUDCONx	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	61
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								61
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

PIC18F8722 FAMILY

20.4 EUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTAx<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CKx pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any low-power mode.

20.4.1 EUSART SYNCHRONOUS SLAVE TRANSMISSION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep mode.

If two words are written to the TXREGx and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSRx register and transmit.
- The second word will remain in the TXREGx register.
- Flag bit, TXxIF, will not be set.
- When the first word has been shifted out of TSRx, the TXREGx register will transfer the second word to the TSRx and flag bit, TXxIF, will now be set.
- If enable bit, TXxIE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
- Clear bits, CREN and SREN.
- If interrupts are desired, set enable bit, TXxIE.
- If 9-bit transmission is desired, set bit, TX9.
- Enable the transmission by setting enable bit, TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
- Start transmission by loading data to the TXREGx register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

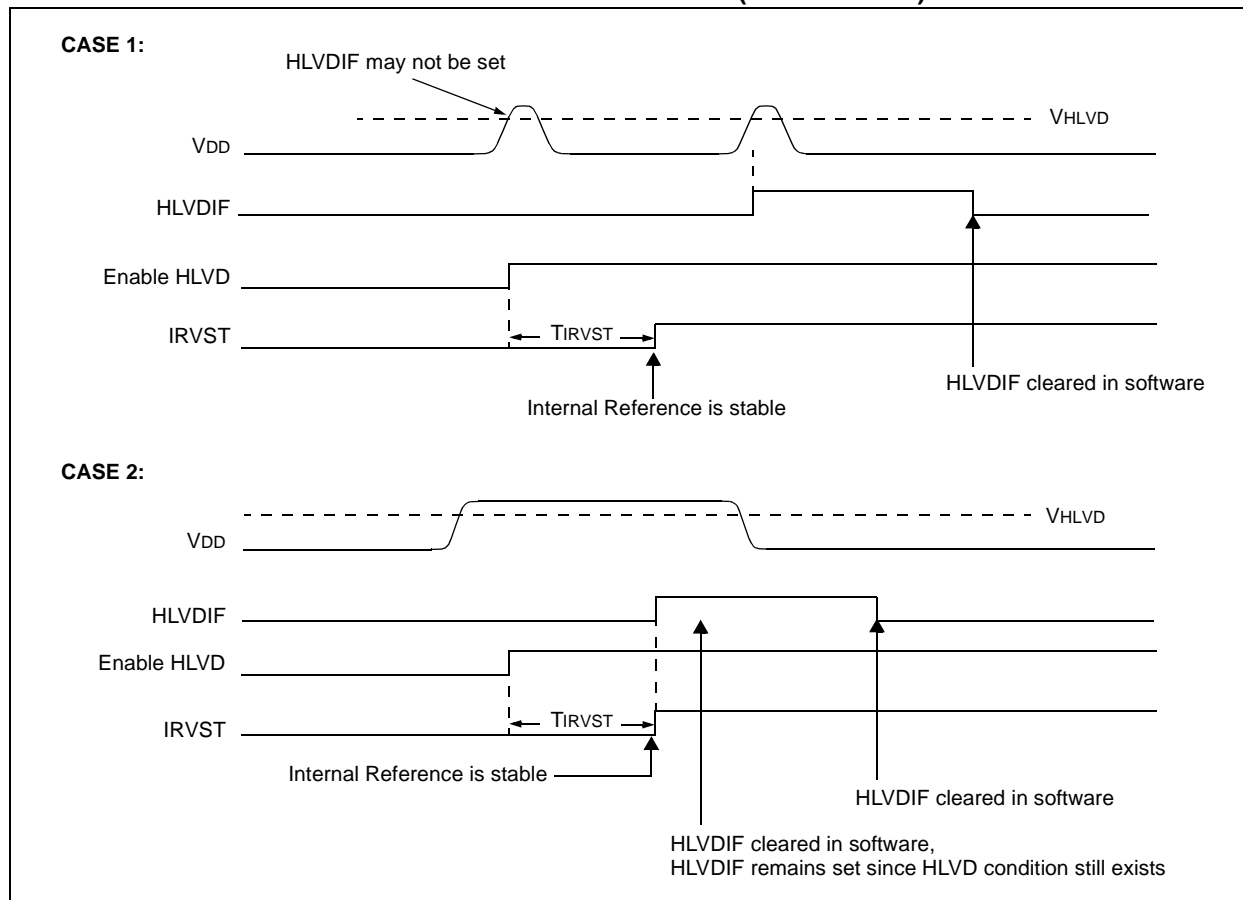
TABLE 20-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	60
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	60
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	60
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
TXREGx	EUSARTx Transmit Register								59
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	59
BAUDCONx	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	61
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								61
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

PIC18F8722 FAMILY

FIGURE 24-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)

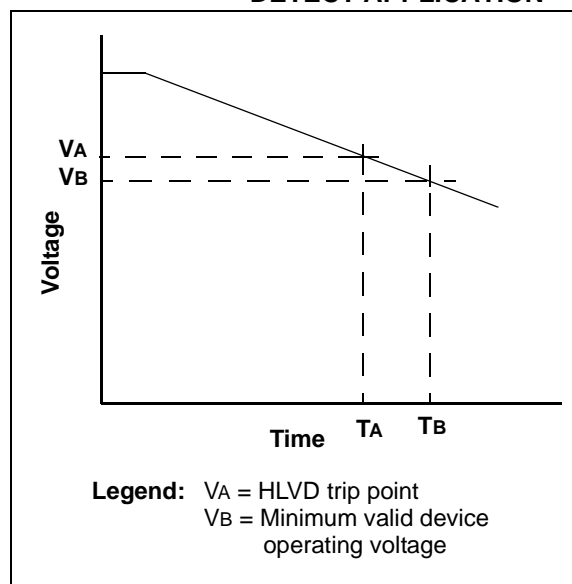


24.5 Applications

In many applications, the ability to detect a drop below or rise above a particular threshold is desirable. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a high-voltage detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 24-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage V_A, the HLVD logic generates an interrupt at time T_A. The interrupt could cause the execution of an ISR, which would allow the application to perform “house-keeping tasks” and perform a controlled shutdown before the device voltage exits the valid operating range at T_B. The HLVD, thus, would give the application a time window, represented by the difference between T_A and T_B, to safely exit.

FIGURE 24-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION



PIC18F8722 FAMILY

REGISTER 25-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	R/P-0	R/P-0	R/P-0	U-0	R/P-1	U-0	R/P-1
DEBUG	XINST	BBSIZ1	BBSIZ0	—	LVP	—	STVREN
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

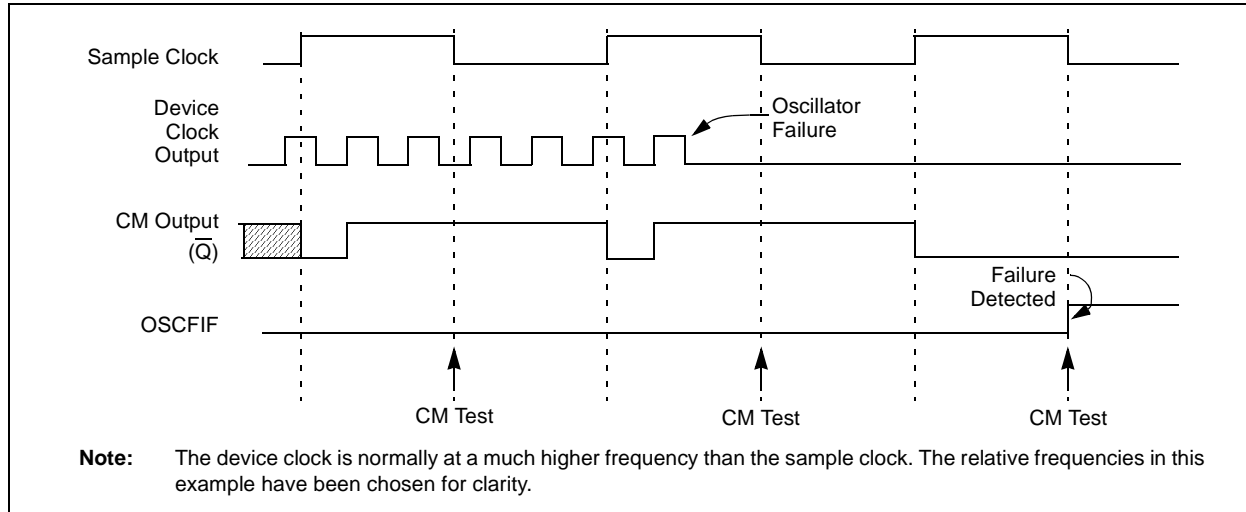
'0' = Bit is cleared

x = Bit is unknown

- bit 7 **DEBUG:** Background Debugger Enable bit
 1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
 0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug
- bit 6 **XINST:** Extended Instruction Set Enable bit
 1 = Instruction set extension and Indexed Addressing mode enabled
 0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)
- bit 5-4 **BBSIZ<1:0>:** Boot Block Size Select bits
 11 = 4K words (8 Kbytes) boot block size
 10 = 4K words (8 Kbytes) boot block size
 01 = 2K words (4 Kbytes) boot block size
 00 = 1K word (2 Kbytes) boot block size
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **LVP:** Single-Supply ICSP™ Enable bit
 1 = Single-Supply ICSP enabled
 0 = Single-Supply ICSP disabled
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **STVREN:** Stack Full/Underflow Reset Enable bit
 1 = Stack full/underflow will cause Reset
 0 = Stack full/underflow will not cause Reset

PIC18F8722 FAMILY

FIGURE 25-4: FSCM TIMING DIAGRAM



25.4.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexor selects the clock source selected by the OSCCON register. Fail-Safe Monitoring of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, subsequent interrupts while in Idle mode will cause the CPU to begin executing instructions while being clocked by the INTOSC source.

25.4.4 POR OR WAKE FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is EC, RC or INTRC modes, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note: The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 25.3.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

PIC18F8722 FAMILY

TABLE 28-23: MASTER SSP I²C™ 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	SDAx and SCLx 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	SDAx and SCLx 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	
			1 MHz mode ⁽¹⁾	TBD	—	ns	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
			1 MHz mode ⁽¹⁾	TBD	—	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	
			1 MHz mode ⁽¹⁾	TBD	—	ms	
D102	CB	Bus Capacitive Loading		—	400	pF	

Legend: TBD = To Be Determined

Note 1: Maximum pin capacitance = 10 pF for all I²C™ pins.

- 2:** A Fast mode I²C bus device can be used in a Standard mode I²C 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 SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode,) before the SCLx line is released.

PIC18F8722 FAMILY

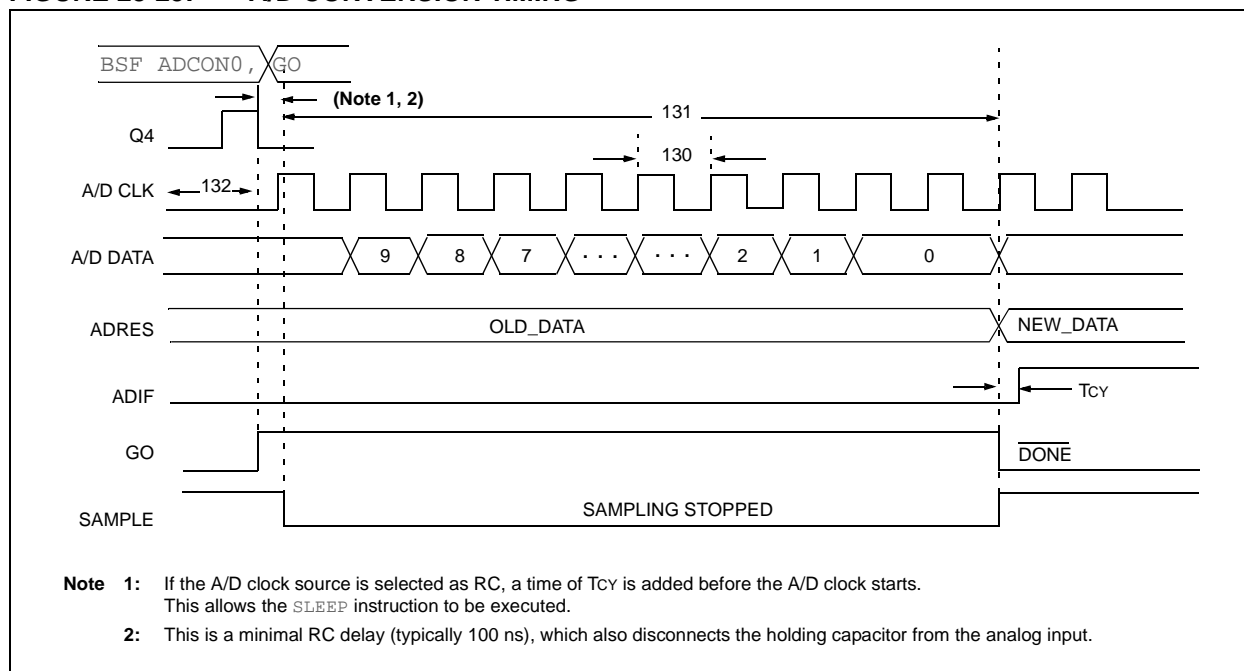
**TABLE 28-26: A/D CONVERTER CHARACTERISTICS: PIC18F6X27/6X22/8X27/8X22 (INDUSTRIAL)
PIC18LF6X27/6X22/8X27/8X22 (INDUSTRIAL)**

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
A01	NR	Resolution	—	—	10	bit	$\Delta V_{REF} \geq 3.0V$
A03	EIL	Integral Linearity Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	Differential Linearity Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A06	EOFF	Offset Error	—	—	$<\pm 2$	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	Gain Error	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A10	—	Monotonicity	Guaranteed ⁽¹⁾			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	ΔV_{REF}	Reference Voltage Range ($V_{REFH} - V_{REFL}$)	1.8	—	—	V	$V_{DD} < 3.0V$
			3	—	—	V	$V_{DD} \geq 3.0V$
A21	V_{REFH}	Reference Voltage High	V_{SS}	—	V_{REFH}	V	
A22	V_{REFL}	Reference Voltage Low	$V_{SS} - 0.3V$	—	$V_{DD} - 3.0V$	V	
A25	V_{AIN}	Analog Input Voltage	V_{REFL}	—	V_{REFH}	V	
A30	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	2.5	k Ω	
A40	I _{AD}	A/D Current from V_{DD}	PIC18FXXXX	—	180	—	μA
			PIC18LFXXXX	—	90	—	μA
A50	I _{REF}	V_{REF} Input Current ⁽²⁾	—	—	5	μA	Average current during conversion
			—	—	150	μA	

Note 1: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

Note 2: V_{REFH} current is from RA3/AN3/ V_{REF+} pin or V_{DD} , whichever is selected as the V_{REFH} source.
 V_{REFL} current is from RA2/AN2/ V_{REF-} pin or V_{SS} , whichever is selected as the V_{REFL} source.

FIGURE 28-25: A/D CONVERSION TIMING



PIC18F8722 FAMILY

Sleep		
OSC1 and OSC2 Pin States	40	
Sleep Mode	45	
Software Simulator (MPLAB SIM)	372	
Special Event Trigger. <i>See</i> Compare (CCP Mode).		
Special Event Trigger. <i>See</i> Compare (ECCP Module).		
Special Features of the CPU	297	
Special Function Registers	75	
Map	75	
SPI Mode (MSSP)	205	
Associated Registers	214	
Bus Mode Compatibility	213	
Clock Speed, Interactions	213	
Effects of a Reset	213	
Enabling SPI I/O	209	
Master Mode	210	
Master/Slave Connection	209	
Operation	208	
Operation in Power-Managed Modes	213	
Serial Clock	205	
Serial Data In	205	
Serial Data Out	205	
Slave Mode	211	
Slave Select	205	
Slave Select Synchronization	211	
SPI Clock	210	
SSPxBUF Register	210	
SSPxSR Register	210	
Typical Connection	209	
SSPOV	236	
SSPOV Status Flag	236	
SSPxSTAT Register		
R/W Bit	219, 220	
SSx	205	
Stack Full/Underflow Resets	68	
SUBFSR	367	
SUBFWB	356	
SUBLW	357	
SUBULNK	367	
SUBWF	357	
SUBWFB	358	
SWAPF	358	
T		
Table Pointer Operations (table)	90	
Table Reads/Table Writes	69	
TBLRD	359	
TBLWT	360	
Time-out in Various Situations (table)	53	
Timer0	161	
Associated Registers	163	
Operation	162	
Overflow Interrupt	163	
Prescaler	163	
Prescaler Assignment (PSA Bit)	163	
Prescaler Select (T0PS2:T0PS0 Bits)	163	
Prescaler. <i>See</i> Prescaler, Timer0.		
Reads and Writes in 16-Bit Mode	162	
Source Edge Select (T0SE Bit)	162	
Source Select (T0CS Bit)	162	
Switching Prescaler Assignment	163	
Timer1	165	
16-Bit Read/Write Mode	167	
Associated Registers	169	
Interrupt	168	
Operation	166	
Oscillator	165, 167	
Layout Considerations	168	
Overflow Interrupt	165	
Resetting, Using the CCP		
Special Event Trigger	168	
Special Event Trigger (ECCP)	192	
TMR1H Register	165	
TMR1L Register	165	
Use as a Real-Time Clock	168	
Timer2	171	
Associated Registers	172	
Interrupt	172	
Operation	171	
Output	172	
PR2 Register	184, 192	
TMR2 to PR2 Match Interrupt	184, 192	
Timer3	173	
16-Bit Read/Write Mode	175	
Associated Registers	175	
Operation	174	
Oscillator	173, 175	
Overflow Interrupt	173, 175	
Special Event Trigger (CCP)	175	
TMR3H Register	173	
TMR3L Register	173	
Timer4	177	
Associated Registers	178	
MSSP Clock Shift	178	
Operation	177	
Postscaler. <i>See</i> Postscaler, Timer4.		
PR4 Register	177	
Prescaler. <i>See</i> Prescaler, Timer4.		
TMR4 Register	177	
TMR4 to PR4 Match Interrupt	177, 178	
Timing Diagrams		
A/D Conversion	416	
Asynchronous Reception	261	
Asynchronous Transmission	258	
Asynchronous Transmission (Back to Back)	258	
Automatic Baud Rate Calculation	256	
Auto-Wake-up Bit (WUE) During		
Normal Operation	262	
Auto-Wake-up Bit (WUE) During Sleep	262	
Baud Rate Generator with Clock Arbitration	233	
BRG Overflow Sequence	256	
BRG Reset Due to SDAX Arbitration		
During Start Condition	242	
Brown-out Reset (BOR)	403	
Bus Collision During a Repeated Start		
Condition (Case 1)	243	
Bus Collision During a Repeated Start		
Condition (Case 2)	243	
Bus Collision During a Start		
Condition (SCLx = 0)	242	
Bus Collision During a Stop		
Condition (Case 1)	244	
Bus Collision During a Stop		
Condition (Case 2)	244	