

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	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	96KB (48K 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 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f6627-i-pt

PIC18F8722 FAMILY

4.4 Brown-out Reset (BOR)

The PIC18F8722 family of devices implements a BOR circuit that provides the user with a number of configuration and power-saving options. The BOR is controlled by the BORV<1:0> and BOREN<1:0> Configuration bits. There are a total of four BOR configurations which are summarized in Table 4-1.

The BOR threshold is set by the BORV<1:0> bits. If BOR is enabled (any values of BOREN<1:0>, except '00'), any drop of VDD below VBOR (parameter D005, **Section 28.1 "DC Characteristics"**) for greater than TBOR (parameter 35, Table 28-12) will reset the device. A Reset may or may not occur if VDD falls below VBOR for less than TBOR. The chip will remain in Brown-out Reset until VDD rises above VBOR.

If the Power-up Timer is enabled, it will be invoked after VDD rises above VBOR; it then will keep the chip in Reset for an additional time delay, TPWRT (parameter 33, Table 28-12). If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above VBOR, the Power-up Timer will execute the additional time delay.

BOR and the Power-on Timer (PWRT) are independently configured. Enabling BOR Reset does not automatically enable the PWRT.

4.4.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = 01, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<6>). Setting SBOREN enables the BOR to function as previously described. Clearing SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise it is read as '0'.

Placing the BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change the BOR configuration. It also allows the user to tailor device power consumption in software by eliminating the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note: Even when BOR is under software control, the BOR Reset voltage level is still set by the BORV<1:0> Configuration bits. It cannot be changed in software.

4.4.2 DETECTING BOR

When BOR is enabled, the $\overline{\text{BOR}}$ bit always resets to '0' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of $\overline{\text{BOR}}$ alone. A more reliable method is to simultaneously check the state of both $\overline{\text{POR}}$ and $\overline{\text{BOR}}$. This assumes that the $\overline{\text{POR}}$ bit is reset to '1' in software immediately after any POR event. If $\overline{\text{BOR}}$ is '0' while $\overline{\text{POR}}$ is '1', it can be reliably assumed that a BOR event has occurred.

4.4.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, the BOR remains under hardware control and operates as previously described. Whenever the device enters Sleep mode, however, the BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

TABLE 4-1: BOR CONFIGURATIONS

BOR Configuration		Status of SBOREN (RCON<6>)	BOR Operation
BOREN1	BOREN0		
0	0	Unavailable	BOR disabled; must be enabled by reprogramming the Configuration bits.
0	1	Available	BOR enabled in software; operation controlled by SBOREN.
1	0	Unavailable	BOR enabled in hardware in Run and Idle modes, disabled during Sleep mode.
1	1	Unavailable	BOR enabled in hardware; must be disabled by reprogramming the Configuration bits.

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.

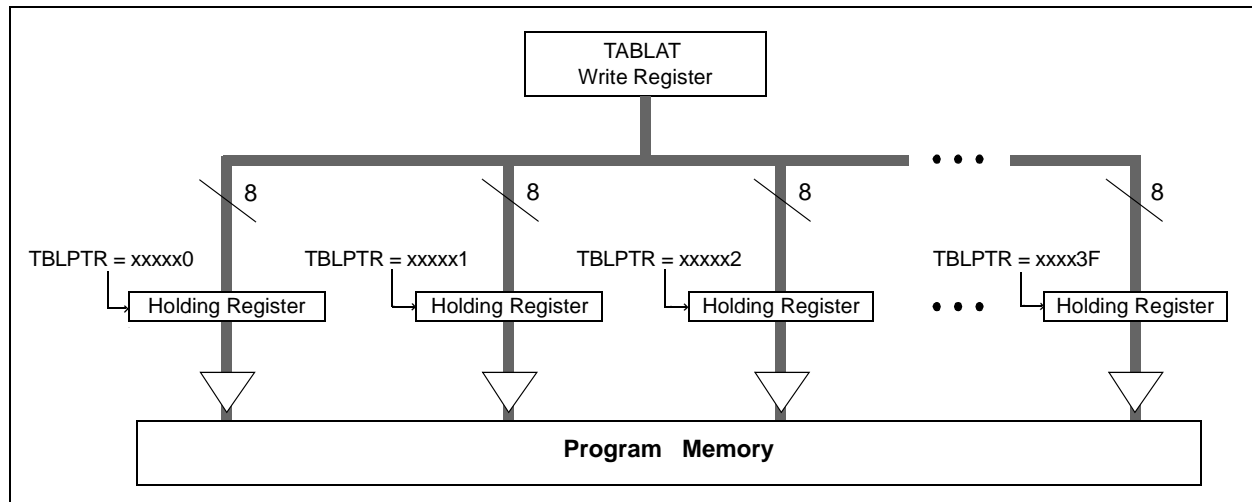
Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers,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

7.2 Address and Data Width

PIC18F8527/8622/8627/8722 devices can be independently configured for different address and data widths on the same memory bus. Both address and data width are set by Configuration bits in the CONFIG3L register. As Configuration bits, this means that these options can only be configured by programming the device and are not controllable in software.

The BW bit selects an 8-bit or 16-bit data bus width. Setting this bit (default) selects a data width of 16 bits.

The ADW<1:0> bits determine the address bus width. The available options are 20-bit (default), 16-bit, 12-bit and 8-bit. Selecting any of the options other than 20-bit width makes a corresponding number of high-order lines available for I/O functions; these pins are no longer affected by the setting of the EBDIS bit. For example, selecting a 16-bit Address mode (ADW<1:0> = 10) disables A<19:16> and allows PORTH<3:0> to function without interruptions from the bus. Using smaller address widths allows users to tailor the memory bus to the size of the external memory space for a particular design while freeing up pins for dedicated I/O operation.

Because the ADW bits have the effect of disabling pins for memory bus operations, it is important to always select an address width at least equal to the data width. If 8-bit or 12-bit address widths are used with a 16-bit data width, the upper bits of data will not be available on the bus.

All combinations of address and data widths require multiplexing of address and data information on the same lines. The address and data multiplexing, as well as I/O ports made available by the use of smaller address widths, are summarized in Table 7-2.

7.2.1 21-BIT ADDRESSING

As an extension of 20-bit address width operation, the External Memory Bus can also fully address a 2 Mbyte memory space. This is done by using the Bus Address bit 0 (BA0) control line as the Least Significant bit of the address. The \overline{UB} and \overline{LB} control signals may also be used with certain memory devices to select the upper and lower bytes within a 16-bit wide data word.

This addressing mode is available in both 8-bit and certain 16-bit Data Width modes. Additional details are provided in **Section 7.5.3 “16-bit Byte Select Mode”** and **Section 7.6 “8-Bit Data Width Modes”**.

7.3 Wait States

While it may be assumed that external memory devices will operate at the microcontroller clock rate, this is often not the case. In fact, many devices require longer times to write or retrieve data than the time allowed by the execution of table read or table write operations.

To compensate for this, the External Memory Bus can be configured to add a fixed delay to each table operation using the bus. Wait states are enabled by setting the WAITx bit. When enabled, the amount of delay is set by the WAIT<1:0> bits (MEMCON<5:4>). The delay is based on multiples of microcontroller instruction cycle time and are added following the instruction cycle when the table operation is executed. The range is from no delay to 3 Tcy (default value).

TABLE 7-2: ADDRESS AND DATA LINES FOR DIFFERENT ADDRESS AND DATA WIDTHS

Data Width	Address Width	Multiplexed Data and Address Lines (and Corresponding Ports)	Address-Only Lines (and Corresponding Ports)	Ports Available for I/O
8-bit	8-bit	AD<7:0> (PORTD<7:0>)	—	All of PORTE and PORTH
	12-bit		AD<11:8> (PORTE<3:0>)	PORTE<7:4>, All of PORTH
	16-bit		AD<15:8> (PORTE<7:0>)	All of PORTH
	20-bit		A<19:16>, AD<15:8> (PORTH<3:0>, PORTE<7:0>)	—
16-bit	16-bit	AD<15:0> (PORTD<7:0>, PORTE<7:0>)	—	All of PORTH
	20-bit		A<19:16> (PORTH<3:0>)	—

PIC18F8722 FAMILY

7.5.2 16-BIT WORD WRITE MODE

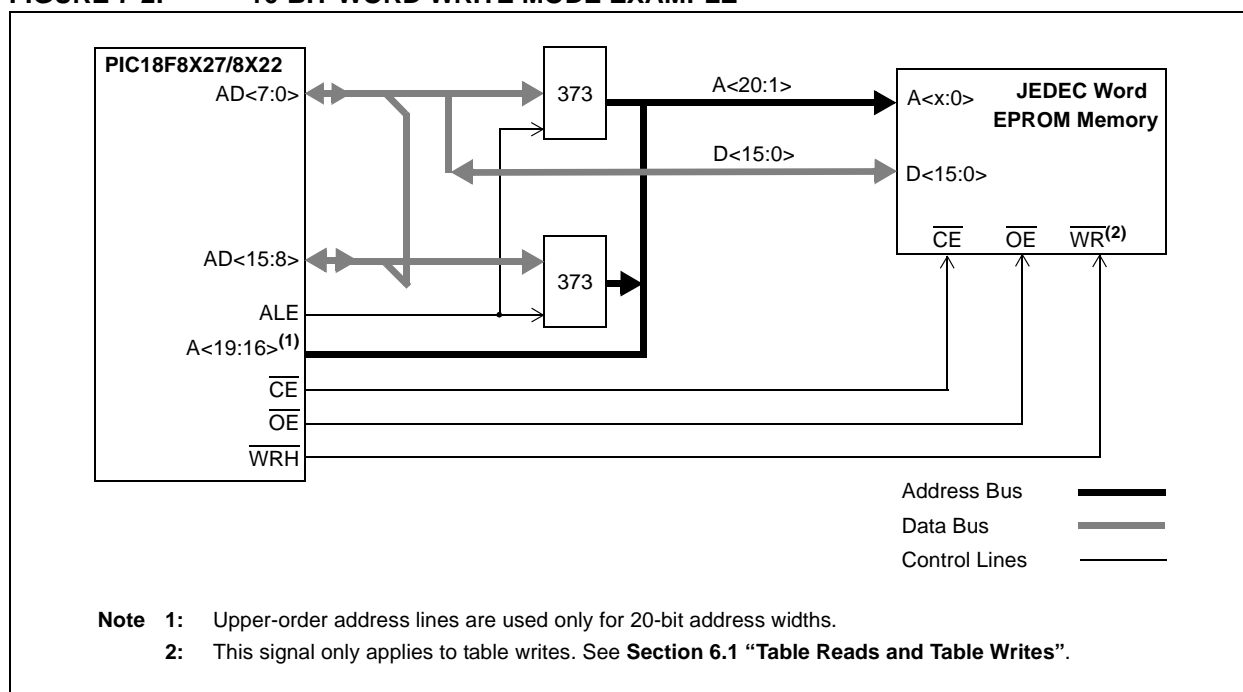
Figure 7-2 shows an example of 16-bit Word Write mode for PIC18F8527/8622/8627/8722 devices. This mode is used for word-wide memories which includes some of the EPROM and Flash-type memories. This mode allows opcode fetches and table reads from all forms of 16-bit memory and table writes to any type of word-wide external memories. This method makes a distinction between TBLWT cycles to even or odd addresses.

During a TBLWT cycle to an even address (TBLPTR<0> = 0), the TABLAT data is transferred to a holding latch and the external address data bus is tri-stated for the data portion of the bus cycle. No write signals are activated.

During a TBLWT cycle to an odd address (TBLPTR<0> = 1), the TABLAT data is presented on the upper byte of the AD15:AD0 bus. The contents of the holding latch are presented on the lower byte of the AD<15:0> bus.

The $\overline{\text{WRH}}$ signal is strobed for each write cycle; the $\overline{\text{WRL}}$ pin is unused. The signal on the BA0 pin indicates the Least Significant bit of TBLPTR but it is left unconnected. Instead, the $\overline{\text{UB}}$ and $\overline{\text{LB}}$ signals are active to select both bytes. The obvious limitation to this method is that the table write must be done in pairs on a specific word boundary to correctly write a word location.

FIGURE 7-2: 16-BIT WORD WRITE MODE EXAMPLE



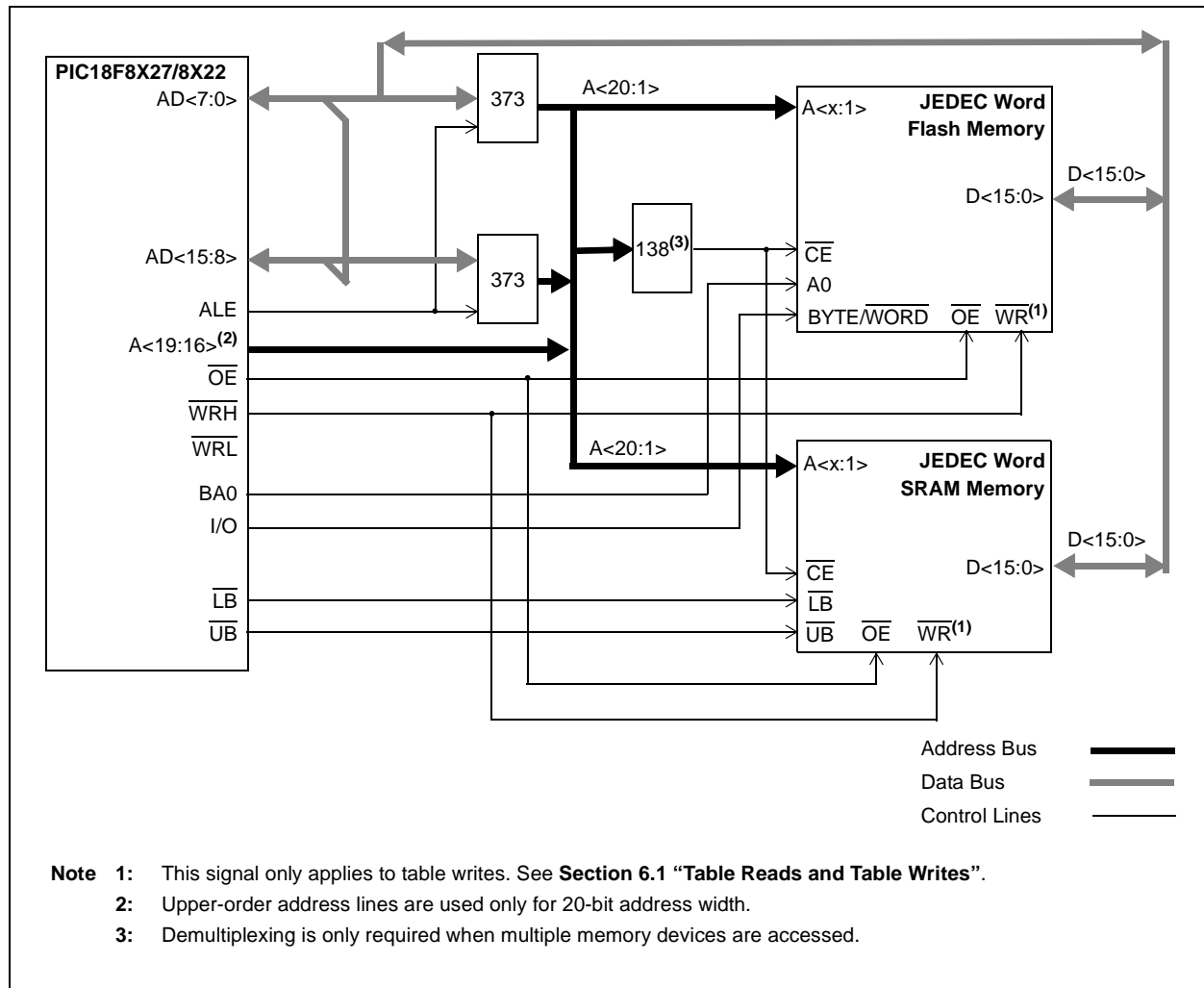
7.5.3 16-BIT BYTE SELECT MODE

Figure 7-3 shows an example of 16-bit Byte Select mode. This mode allows table write operations to word-wide external memories with byte selection capability. This generally includes both word-wide Flash and SRAM devices.

During a TBLWT cycle, the TABLAT data is presented on the upper and lower byte of the AD<15:0> bus. The WRH signal is strobed for each write cycle; the WRL pin is not used. The BA0 or $\overline{UB}/\overline{LB}$ signals are used to select the byte to be written, based on the Least Significant bit of the TBLPTR register.

Flash and SRAM devices use different control signal combinations to implement Byte Select mode. JEDEC standard Flash memories require that a controller I/O port pin be connected to the memory's BYTE/WORD pin to provide the select signal. They also use the BA0 signal from the controller as a byte address. JEDEC standard static RAM memories, on the other hand, use the \overline{UB} or \overline{LB} signals to select the byte.

FIGURE 7-3: 16-BIT BYTE SELECT MODE EXAMPLE



10.0 INTERRUPTS

The PIC18F8722 family of devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit, which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit, which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the interrupt control registers while any interrupt is enabled. Doing so may cause erratic microcontroller behavior.
--

PIC18F8722 FAMILY

TABLE 11-13: PORTG FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RG0/ECCP3/P3A	RG0	0	O	DIG	LATG<0> data output.
		1	I	ST	PORTG<0> data input.
	ECCP3	0	O	DIG	ECCP3 compare and ECCP3 PWM output. Takes priority over port data.
		1	I	ST	ECCP3 capture input.
	P3A	0	O	DIG	ECCP3 Enhanced PWM output, channel B. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RG1/TX2/CK2	RG1	0	O	DIG	LATG<1> data output.
		1	I	ST	PORTG<1> data input.
	TX2	0	O	DIG	Asynchronous serial transmit data output (EUSART2 module). Takes priority over port data.
		0	O	DIG	Synchronous serial clock output (EUSART2 module). Takes priority over port data.
		1	I	ST	Synchronous serial clock input (EUSART2 module).
RG2/RX2/DT2	RG2	0	O	DIG	LATG<2> data output.
		1	I	ST	PORTG<2> data input.
	RX2	1	I	ST	Asynchronous serial receive data input (EUSART2 module).
		1	O	DIG	Synchronous serial data output (EUSART2 module). Takes priority over port data. User must configure as an input.
		1	I	ST	Synchronous serial data input (EUSART2 module). User must configure as an input.
RG3/CCP4/P3D	RG3	0	O	DIG	LATG<3> data output.
		1	I	ST	PORTG<3> data input.
	CCP4	0	O	DIG	CCP4 compare and PWM output; takes priority over port data and P3D function.
		1	I	ST	CCP4 capture input.
	P3D	0	O	DIG	ECCP3 Enhanced PWM output, channel D. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RG4/CCP5/P1D	RG4	0	O	DIG	LATG<4> data output.
		1	I	ST	PORTG<4> data input.
	CCP5	0	O	DIG	CCP5 compare and PWM output. Takes priority over port data and P1D function.
		1	I	ST	CCP5 capture input.
	P1D	0	O	DIG	ECCP1 Enhanced PWM output, channel B. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RG5/MCLR/VPP	RG5	— ⁽¹⁾	I	ST	PORTG<5> data input; enabled when MCLRE Configuration bit is clear.
	MCLR	—	I	ST	External Master Clear input; enabled when MCLRE Configuration bit is set.
	VPP	—	I	ANA	High-voltage detection; used for ICSP™ mode entry detection. Always available regardless of pin mode.

Legend: PWR = Power Supply, O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: RG5 does not have a corresponding TRISG bit.

PIC18F8722 FAMILY

11.10 Parallel Slave Port

PORTD can also function as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (PSPCON<4>) is set. It is asynchronously readable and writable by the external world through the RD and WR control input pins.

Note: For PIC18F8527/8622/8627/8722 devices, the Parallel Slave Port is available only in Microcontroller mode.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/RD to be the RD input, RE1/WR to be the WR input and RE2/CS to be the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set).

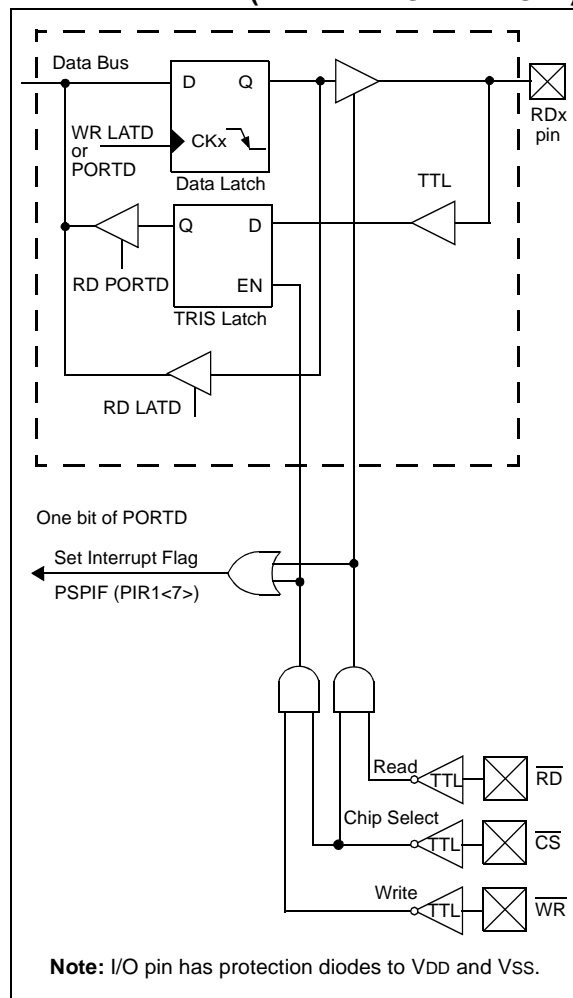
A write to the PSP occurs when both the CS and WR lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits are both set when the write ends.

A read from the PSP occurs when both the CS and RD lines are first detected low. The data in PORTD is read out and the OBF bit is set. If the user writes new data to PORTD to set OBF, the data is immediately read out; however, the OBF bit is not set.

When either the CS or RD lines are detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP; when this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 11-3 and Figure 11-4, respectively.

FIGURE 11-2: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



PIC18F8722 FAMILY

18.4.3 PWM OUTPUT CONFIGURATIONS

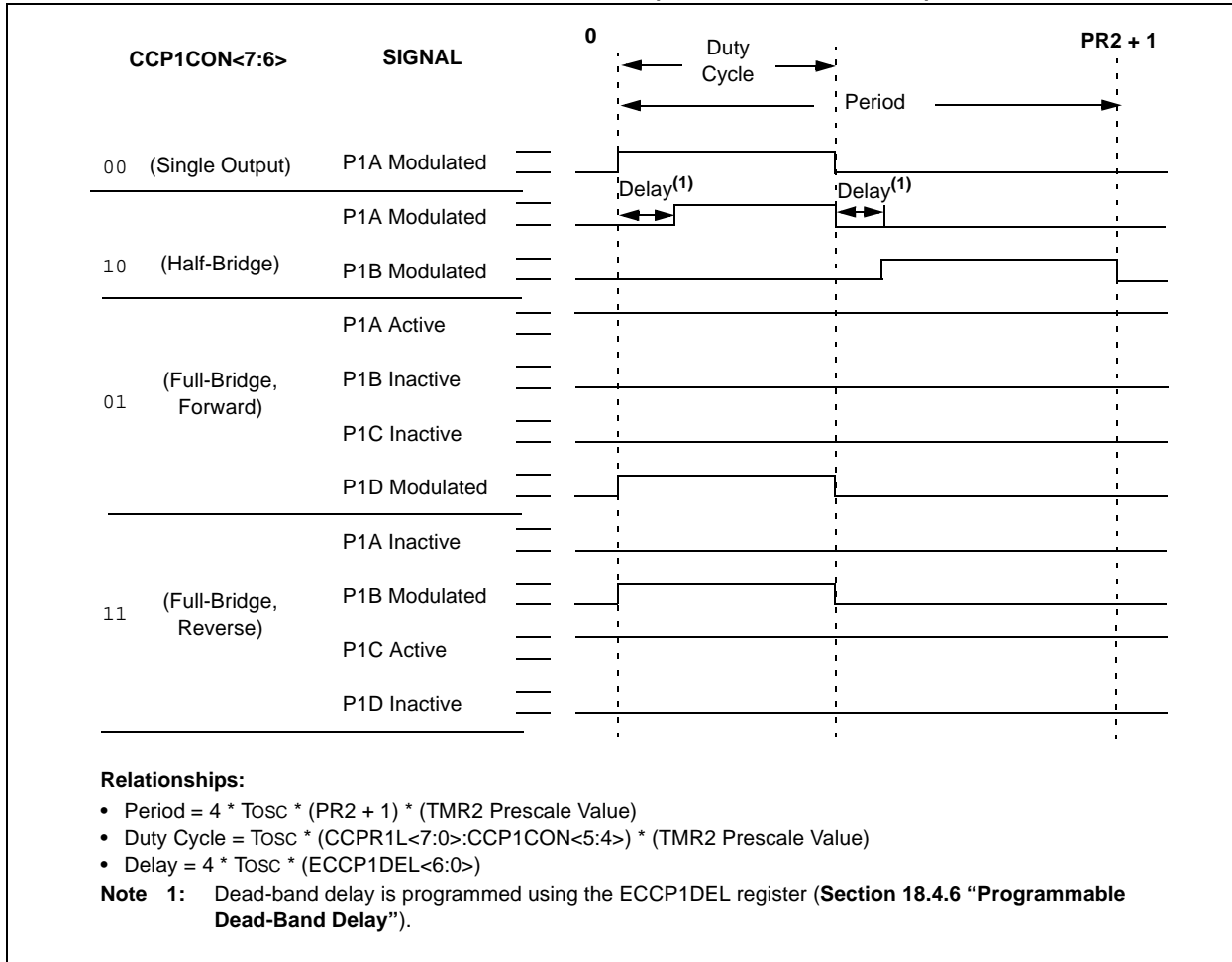
The P1M1:P1M0 bits in the CCP1CON register allow one of four configurations:

- Single Output
- Half-Bridge Output
- Full-Bridge Output, Forward mode
- Full-Bridge Output, Reverse mode

The Single Output mode is the standard PWM mode discussed in **Section 18.4 “Enhanced PWM Mode”**. The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 18-2.

FIGURE 18-2: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)



20.0 ENHANCED UNIVERSAL SYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break Character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-Wake-up on Character Reception
 - Auto-Baud Calibration
 - 12-bit Break Character Transmission
- Synchronous – Master (half duplex) with Selectable Clock Polarity
- Synchronous – Slave (half duplex) with Selectable Clock Polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2 and RG2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
 - bit SPEN (RCSTA1<7>) must be set (= 1)
 - bit TRISC<7> must be set (= 1)
 - bit TRISC<6> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode
- For EUSART2:
 - bit SPEN (RCSTA2<7>) must be set (= 1)
 - bit TRISG<2> must be set (= 1)
 - bit TRISG<1> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

Note: Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2

20.2 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTAx<4>). In this mode, the EUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit/16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate depending on the BRGH and BRG16 bits (TXSTAx<2> and BAUDCONx<3>). Parity is not supported by the hardware, but can be implemented in software and stored as the 9th data bit.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-bit Break Character Transmit
- Auto-Baud Rate Detection

20.2.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSRx). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx register is loaded with data in software. The TSRx register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSRx is loaded with new data from the TXREGx register (if available).

Once the TXREGx register transfers the data to the TSRx register (occurs in one Tcy), the TXREGx register is empty and the TXxIF flag bit (PIR1<4>) is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxIE (PIE1<4>). TXxIF will be set regardless of the state of TXxIE; it cannot be cleared in software. TXxIF is also not cleared immediately upon loading TXREGx, but becomes valid in the second instruction cycle following the load instruction. Polling TXxIF immediately following a load of TXREGx will return invalid results.

While TXxIF indicates the status of the TXREGx register, another bit, TRMT (TXSTAx<1>), shows the status of the TSRx register. TRMT is a read-only bit which is set when the TSRx register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSRx register is empty.

Note 1: The TSRx register is not mapped in data memory so it is not available to the user.

2: Flag bit, TXxIF, is set when enable bit TXEN is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, TXxIE.
4. If 9-bit transmission is desired, set transmit bit, TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TXxIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Load data to the TXREGx register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

20.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode and bit SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSRx register will transfer the data to the RCREGx register; if the RCxIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RCxIE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RCxIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCxIE, was set.
6. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREGx register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

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
RCREGx	EUSARTx Receive 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 reception.

PIC18F8722 FAMILY

21.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 21-3. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note: When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 21-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the A/D). The 1/2 LSB error is the maximum error allowed for the A/D to meet its specified resolution.

Example 21-3 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSB
VDD	=	5V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 21-1: ACQUISITION TIME

$$\begin{aligned} \text{TACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= \text{TAMP} + \text{TC} + \text{TCOFF} \end{aligned}$$

EQUATION 21-2: A/D MINIMUM CHARGING TIME

$$\begin{aligned} \text{VHOLD} &= (\text{VREF} - (\text{VREF}/2048)) \cdot (1 - e^{-(\text{TC}/\text{CHOLD}(\text{RIC} + \text{RSS} + \text{RS})))} \\ \text{or} \\ \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \end{aligned}$$

EQUATION 21-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$\text{TACQ} = \text{TAMP} + \text{TC} + \text{TCOFF}$$

$$\text{TAMP} = 0.2 \mu\text{s}$$

$$\begin{aligned} \text{TCOFF} &= (\text{Temp} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &= (85^\circ\text{C} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &= 1.2 \mu\text{s} \end{aligned}$$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, TCOFF = 0 ms.

$$\begin{aligned} \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2047) \mu\text{s} \\ &= -(25 \text{ pF})(1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \mu\text{s} \\ &= 1.05 \mu\text{s} \end{aligned}$$

$$\begin{aligned} \text{TACQ} &= 0.2 \mu\text{s} + 1 \mu\text{s} + 1.2 \mu\text{s} \\ &= 2.4 \mu\text{s} \end{aligned}$$

21.2 Selecting and Configuring Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set. It also gives users the option to use an automatically determined acquisition time.

Acquisition time may be set with the ACQT<2:0> bits (ADCON2<5:3>) which provides a range of 2 to 20 TAD. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

Manual acquisition is selected when ACQT<2:0> = 000. When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This option is also the default Reset state of the ACQT<2:0> bits and is compatible with devices that do not offer programmable acquisition times.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

21.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. There are seven possible options for TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible, but greater than the minimum TAD (see parameter 130, Table 28-27 for more information).

Table 21-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 21-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency	
Operation	ADCS<2:0>	PIC18FXXXX	PIC18LFXXXX ⁽⁴⁾
2 TOSC	000	2.86 MHz	1.43 kHz
4 TOSC	100	5.71 MHz	2.86 MHz
8 TOSC	001	11.43 MHz	5.72 MHz
16 TOSC	101	22.86 MHz	11.43 MHz
32 TOSC	010	40.0 MHz	22.86 MHz
64 TOSC	110	40.0 MHz	22.86 MHz
RC ⁽³⁾	x11	1.00 MHz ⁽¹⁾	1.00 MHz ⁽²⁾

Note 1: The RC source has a typical TAD time of 1.2 μ s.

2: The RC source has a typical TAD time of 2.5 μ s.

3: For device frequencies above 1 MHz, the device must be in Sleep for the entire conversion or the A/D accuracy may be out of specification.

4: Low-power (PIC18LFXXXX) devices only.

PIC18F8722 FAMILY

21.4 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 RC clock source should be selected.

Operation in the Sleep mode requires the A/D FRC clock to be selected. If bits ACQT<2:0> are set to '000' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode. The IDLEN bit (OSCCON<7>) must have already been cleared prior to starting the conversion.

21.5 Configuring Analog Port Pins

The ADCON1, TRISA, TRISF and TRISH registers all configure the A/D port pins. The port pins needed as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<3:0> bits and the TRIS bits.

Note 1: When reading the Port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert as analog inputs. Analog levels on a digitally configured input will be accurately converted.

2: Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

PIC18F8722 FAMILY

REGISTER 25-13: DEVID1: DEVICE ID REGISTER 1 FOR THE PIC18F8722 FAMILY

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
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

DEV<2:0>: Device ID bits

001 = PIC18F8722

111 = PIC18F8627

101 = PIC18F8622

011 = PIC18F8527

000 = PIC18F6722

110 = PIC18F6627

100 = PIC18F6622

010 = PIC18F6527

bit 4-0

REV<4:0>: Revision ID bits

These bits are used to indicate the device revision.

REGISTER 25-14: DEVID2: DEVICE ID REGISTER 2 FOR THE PIC18F8722 FAMILY

R	R	R	R	R	R	R	R
DEV10 ⁽¹⁾	DEV9 ⁽¹⁾	DEV8 ⁽¹⁾	DEV7 ⁽¹⁾	DEV6 ⁽¹⁾	DEV5 ⁽¹⁾	DEV4 ⁽¹⁾	DEV3 ⁽¹⁾
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 7-0

DEV<10:3>: Device ID bits⁽¹⁾

These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

0001 0100 = PIC18F6722/8722 devices

0001 0011 = PIC18F6527/6622/6627/8527/8622/8627 devices

Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

27.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

27.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

27.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

27.10 MPLAB PM3 Device Programmer

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

PIC18F8722 FAMILY

FIGURE 28-1: PIC18F8722 DEVICE FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

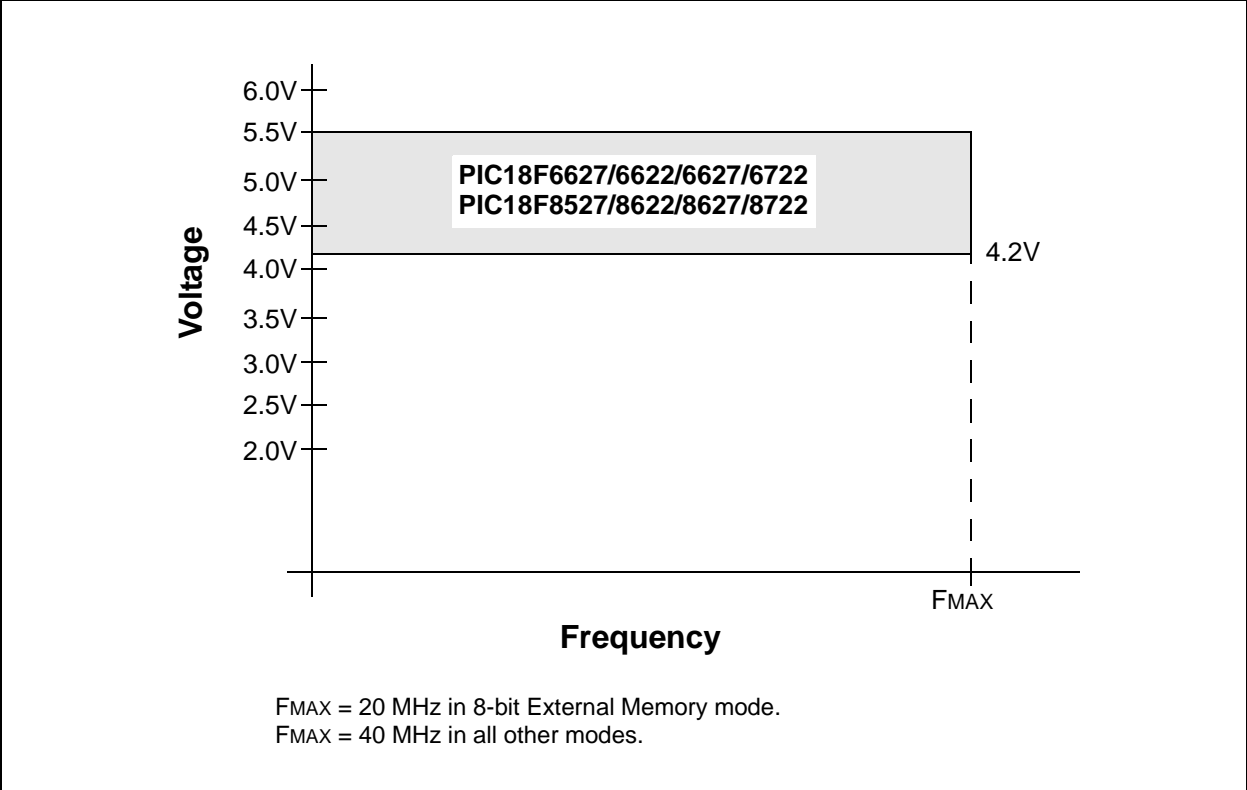
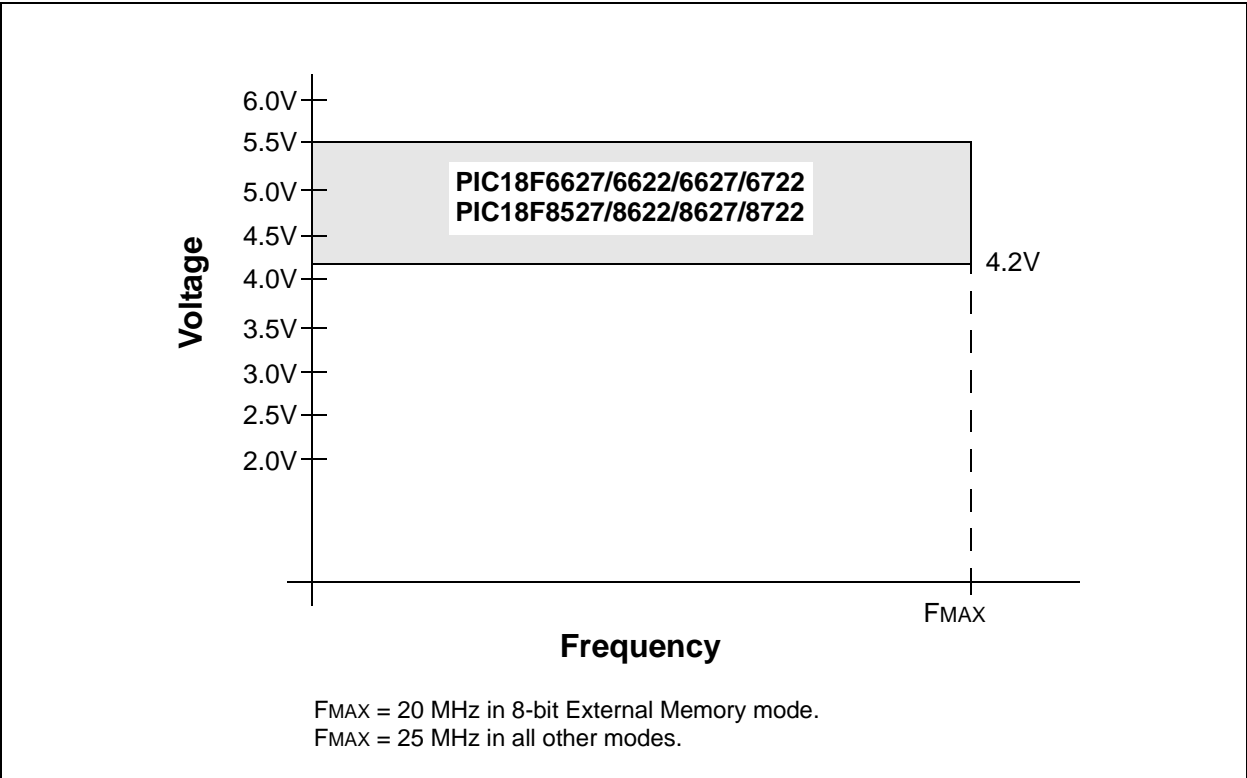


FIGURE 28-2: PIC18F8722 DEVICE FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED)



INDEX

A

A/D	271
A/D Converter Interrupt, Configuring	275
Acquisition Requirements	276
ADCON0 Register	271
ADCON1 Register	271
ADCON2 Register	271
ADRESH Register	271, 274
ADRESL Register	271
Analog Port Pins	158
Analog Port Pins, Configuring	278
Associated Registers	280
Configuring the Module	275
Conversion Clock (TAD)	277
Conversion Status (GO/DONE Bit)	274
Conversions	279
Converter Characteristics	416
Discharge	279
Operation in Power-Managed Modes	278
Selecting and Configuring Acquisition Time	277
Special Event Trigger (ECCP)	192
Special Event Trigger (ECCP2)	280
Use of the ECCP2 Trigger	280
Absolute Maximum Ratings	375
AC (Timing) Characteristics	396
Load Conditions for Device	
Timing Specifications	397
Parameter Symbolology	396
Temperature and Voltage Specifications	397
Timing Conditions	397
Access Bank	
Mapping in Indexed Literal Offset Mode	85
ACKSTAT	236
ACKSTAT Status Flag	236
ADCON0 Register	271
GO/DONE Bit	274
ADCON1 Register	271
ADCON2 Register	271
ADDFSR	364
ADDLW	327
ADDULNK	364
ADDWF	327
ADDWFC	328
ADRESH Register	271
ADRESL Register	271, 274
Analog-to-Digital Converter. <i>See</i> A/D.	
ANDLW	328
ANDWF	329
Assembler	
MPASM Assembler	372
Auto-Wake-up on Sync Break Character	262

B

Bank Select Register (BSR)	72
Baud Rate Generator	232
BC	329
BCF	330
BF	236
BF Status Flag	236
Block Diagrams	
16-Bit Byte Select Mode	103
16-Bit Byte Write Mode	101
16-Bit Word Write Mode	102
A/D	274

Analog Input Model	275
Baud Rate Generator	232
Capture Mode Operation	181
Comparator Analog Input Model	285
Comparator I/O Operating Modes	282
Comparator Output	284
Comparator Voltage Reference	288
Comparator Voltage Reference Output	
Buffer Example	289
Compare Mode Operation	182
Device Clock	37
Enhanced PWM	193
EUSART Receive	260
EUSART Transmit	258
External Power-on Reset Circuit	
(Slow VDD Power-up)	51
Fail-Safe Clock Monitor (FSCM)	315
Generic I/O Port Operation	135
High/Low-Voltage Detect with External Input	292
HSPLL	33
Interrupt Logic	120
INTOSC and PLL	34
MSSP (I ² C Master Mode)	230
MSSP (I ² C Mode)	215
MSSP (SPI Mode)	205
On-Chip Reset Circuit	49
PIC18F6527/6622/6627/6722	11
PIC18F8527/8622/8627/8722	12
PORTD and PORTE (Parallel Slave Port)	158
PWM Operation (Simplified)	184
Reads from Flash Program Memory	91
Single Comparator	283
Table Read Operation	87
Table Write Operation	88
Table Writes to Flash Program Memory	93
Timer0 in 16-Bit Mode	162
Timer0 in 8-Bit Mode	162
Timer1	166
Timer1 (16-Bit Read/Write Mode)	166
Timer2	172
Timer3	174
Timer3 (16-Bit Read/Write Mode)	174
Timer4	178
Watchdog Timer	312
BN	330
BNC	331
BNN	331
BN OV	332
BNZ	332
BOR. <i>See</i> Brown-out Reset.	
BOV	335
BRA	333
Break Character (12-Bit) Transmit and Receive	263
BRG. <i>See</i> Baud Rate Generator.	
Brown-out Reset (BOR)	52
Detecting	52
Disabling in Sleep Mode	52
Software Enabled	52
BSF	333
BTFSC	334
BTFSS	334
BTG	335
BZ	336

PIC18F8722 FAMILY

Sleep	
OSC1 and OSC2 Pin States	40
Sleep Mode	45
Software Simulator (MPLAB SIM)	372
Special Event Trigger. See Compare (CCP Mode).	
Special Event Trigger. See 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. See 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. See Postscaler, Timer4.	
PR4 Register	177
Prescaler. See 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