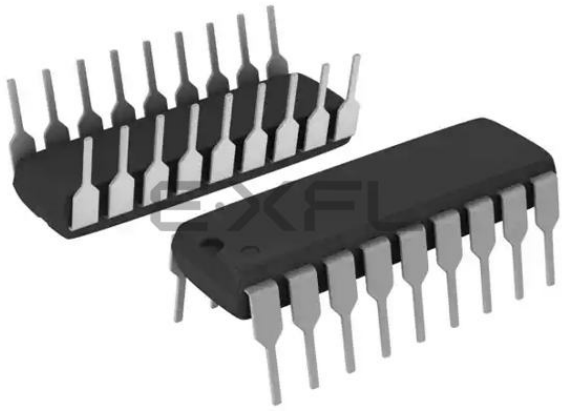


Welcome to E-XFL.COM

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	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1320-i-p

TABLE 1-2: PIC18F1220/1320 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP/ SOIC	SSOP	QFN			
MCLR/VPP/RA5	4	4	1			Master Clear (input) or programming voltage (input).
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P	—	Programming voltage input.
RA5				I	ST	Digital input.
OSC1/CLKI/RA7	16	18	21			Oscillator crystal or external clock input.
OSC1				I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise.
CLKI				I	CMOS	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7				I/O	ST	General purpose I/O pin.
OSC2/CLKO/RA6	15	17	20			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC, EC and INTRC modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes instruction cycle rate.
RA6				I/O	ST	General purpose I/O pin.
						PORTA is a bidirectional I/O port.
RA0/AN0	1	1	26			
RA0				I/O	ST	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1/LVDIN	2	2	27			
RA1				I/O	ST	Digital I/O.
AN1				I	Analog	Analog input 1.
LVDIN				I	Analog	Low-Voltage Detect input.
RA2/AN2/VREF-	6	7	7			
RA2				I/O	ST	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
RA3/AN3/VREF+	7	8	8			
RA3				I/O	ST	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI	3	3	28			
RA4				I/O	ST/OD	Digital I/O. Open-drain when configured as output.
T0CKI				I	ST	Timer0 external clock input.
RA5						See the MCLR/VPP/RA5 pin.
RA6						See the OSC2/CLKO/RA6 pin.
RA7						See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
O = Output
OD = Open-drain (no P diode to VDD)
CMOS = CMOS compatible input or output
I = Input
P = Power

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
BSR	1220	1320	---- 0000	---- 0000	---- uuuu
INDF2	1220	1320	N/A	N/A	N/A
POSTINC2	1220	1320	N/A	N/A	N/A
POSTDEC2	1220	1320	N/A	N/A	N/A
PREINC2	1220	1320	N/A	N/A	N/A
PLUSW2	1220	1320	N/A	N/A	N/A
FSR2H	1220	1320	---- 0000	---- 0000	---- uuuu
FSR2L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	1220	1320	---x xxxx	---u uuuu	---u uuuu
TMR0H	1220	1320	0000 0000	0000 0000	uuuu uuuu
TMR0L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	1220	1320	1111 1111	1111 1111	uuuu uuuu
OSCCON	1220	1320	0000 q000	0000 q000	uuuu qquu
LVDCON	1220	1320	--00 0101	--00 0101	--uu uuuu
WDTCON	1220	1320	---- ---0	---- ---0	---- ---u
RCON(4)	1220	1320	0--1 11q0	0--q qquu	u--u qquu
TMR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	1220	1320	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	1220	1320	0000 0000	0000 0000	uuuu uuuu
PR2	1220	1320	1111 1111	1111 1111	1111 1111
T2CON	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADRESH	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1220	1320	00-0 0000	00-0 0000	uu-u uuuu
ADCON1	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADCON2	1220	1320	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
PWM1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
ECCPAS	1220	1320	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 4-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the Oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 5 of PORTA is enabled if $\overline{\text{MCLR}}$ is disabled.

FIGURE 4-6: SLOW RISE TIME (MCLR TIED TO VDD, VDD RISE > TPWRT)

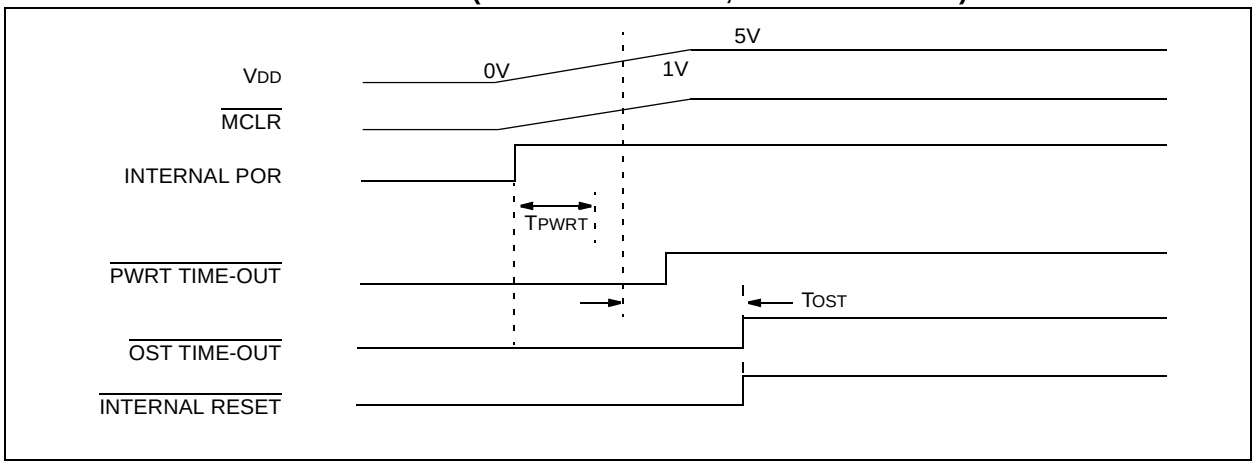
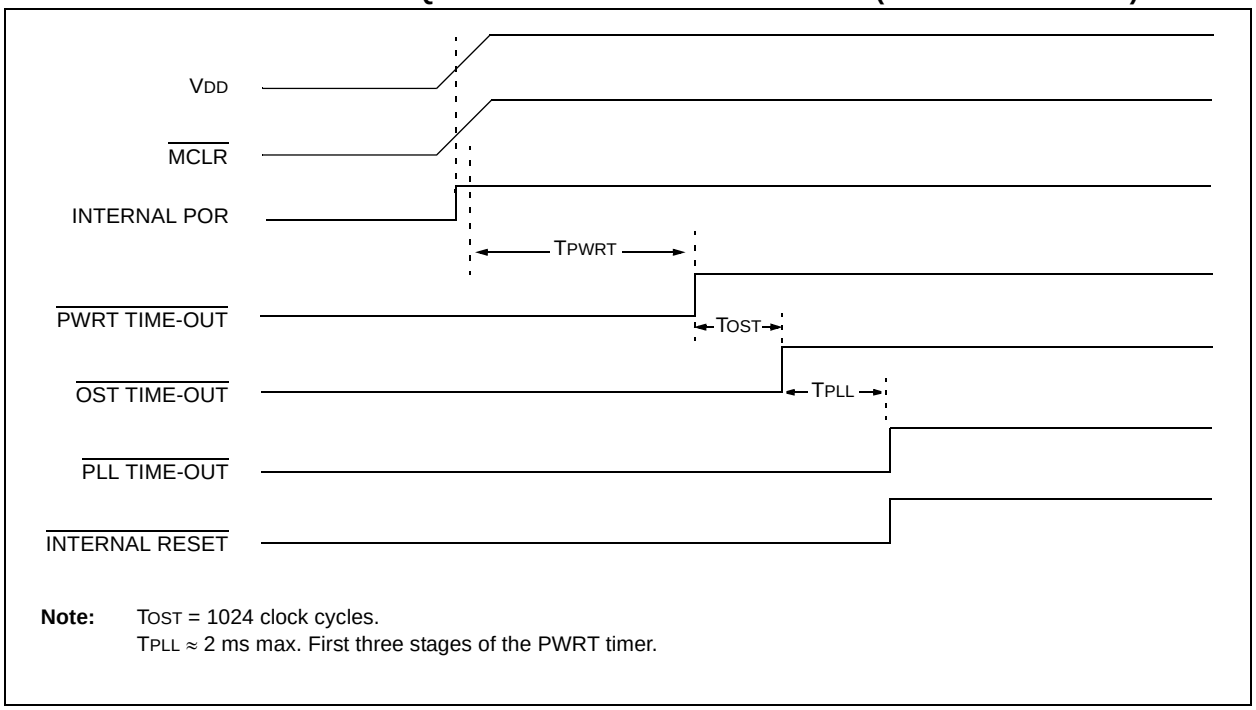


FIGURE 4-7: TIME-OUT SEQUENCE ON POR W/PLL ENABLED (MCLR TIED TO VDD)



Note: TOST = 1024 clock cycles.
TPLL ≈ 2 ms max. First three stages of the PWRT timer.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F1220/1320) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
TMR1H	Timer1 Register High Byte								xxxx xxxx	35, 103
TMR1L	Timer1 Register Low Byte								xxxx xxxx	35, 103
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	35, 98
TMR2	Timer2 Register								0000 0000	35, 104
PR2	Timer2 Period Register								1111 1111	35, 104
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	35, 104
ADRESH	A/D Result Register High Byte								xxxx xxxx	35, 159
ADRESL	A/D Result Register Low Byte								xxxx xxxx	35, 159
ADCON0	VCFG1	VCFG0	—	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	00-0 0000	35, 150
ADCON1	—	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	-000 0000	35, 151
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	35, 152
CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	35. 110
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	35, 110
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	35, 109
PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	35, 121
ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	35, 122
TMR3H	Timer3 Register High Byte								xxxx xxxx	36, 108
TMR3L	Timer3 Register Low Byte								xxxx xxxx	36, 108
T3CON	RD16	—	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0-00 0000	36, 106
SPBRGH	EUSART Baud Rate Generator High Byte								0000 0000	36
SPBRG	EUSART Baud Rate Generator Low Byte								0000 0000	36, 130
RCREG	EUSART Receive Register								0000 0000	36, 138, 137
TXREG	EUSART Transmit Register								0000 0000	36, 135, 137
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	36, 127
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	36, 128
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	36, 129
EEADR	EEPROM Address Register								0000 0000	36, 64
EEDATA	EEPROM Data Register								0000 0000	36, 67
EECON2	EEPROM Control Register 2 (not a physical register)								0000 0000	36, 56, 64
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	36, 57, 65
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	TMR3IP	—	1--1 -11-	36, 80
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	TMR3IF	—	0--0 -00-	36, 76
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	TMR3IE	—	0--0 -00-	36, 78
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	36, 79
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	36, 75
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	36, 77
OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	36, 14
TRISB	Data Direction Control Register for PORTB								1111 1111	36, 94
TRISA	TRISA7(2)	TRISA6(1)	—	Data Direction Control Register for PORTA					11-1 1111	36, 85
LATB	Read/Write PORTB Data Latch								xxxx xxxx	36, 94
LATA	LATA<7>(2)	LATA<6>(1)	—	Read/Write PORTA Data Latch					xx-x xxxx	36, 85
PORTB	Read PORTB pins, Write PORTB Data Latch								xxxx xxxx	36, 94
PORTA	RA7(2)	RA6(1)	RA5(4)	Read PORTA pins, Write PORTA Data Latch					xx0x 0000	36, 85

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

Note 1: RA6 and associated bits are configured as port pins in RCIO, ECIO and INTIO2 (with port function on RA6) Oscillator mode only and read '0' in all other oscillator modes.

2: RA7 and associated bits are configured as port pins in INTIO2 Oscillator mode only and read '0' in all other modes.

3: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

4: The RA5 port bit is only available when MCLRE fuse (CONFIG3H<7>) is programmed to '0'. Otherwise, RA5 reads '0'. This bit is read-only.

REGISTER 6-1: EECON1: EEPROM CONTROL 1 REGISTER

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
- 1 = Access program Flash memory
- 0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Accesses Configuration, User ID and Device ID Registers
- 0 = Accesses Flash Program or data EEPROM Memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit
- 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation – TBLPTR<5:0> are ignored)
- 0 = Perform write only

bit 3 **WRERR:** EEPROM Error Flag bit(1)
- 1 = A write operation was prematurely terminated (any Reset during self-timed programming)
- 0 = The write operation completed normally

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle completed

bit 0 **RD:** Read Control bit
- 1 = Initiates a memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
- 0 = Read completed

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

6.5 Writing to Flash Program Memory

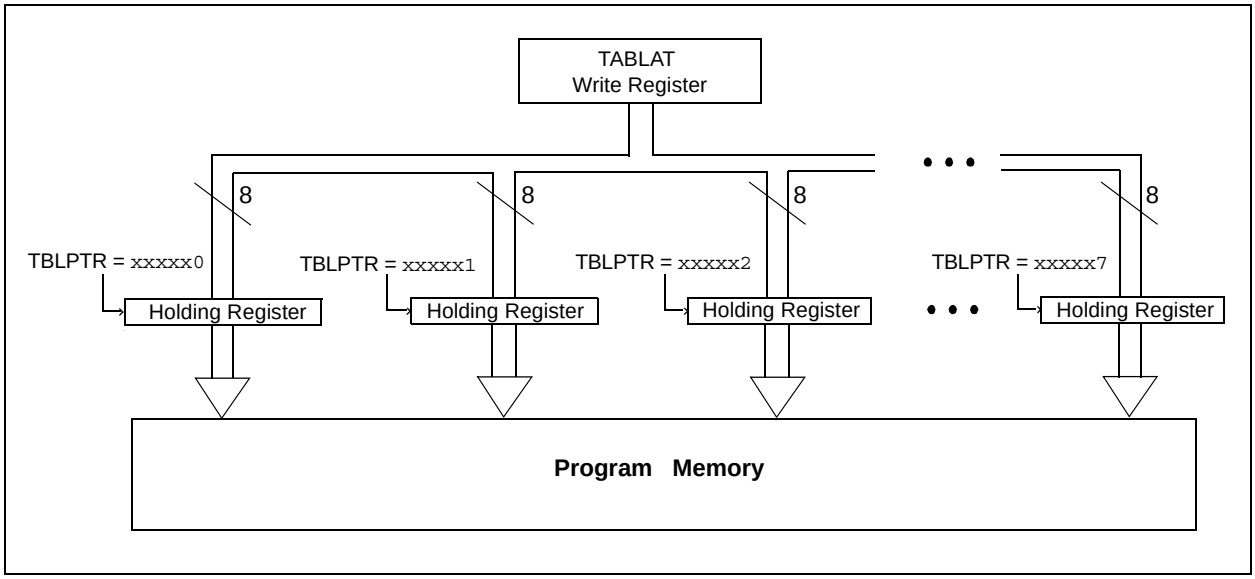
The programming block size is 4 words or 8 bytes. Word or byte programming is not supported.

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

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction must be executed eight 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 eight registers, the EECON1 register must be written to, 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.

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 with address being erased.
4. Do the row erase procedure (see **Section 6.4.1 "Flash Program Memory Erase Sequence"**).
5. Load Table Pointer with address of first byte being written.
6. Write the first 8 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 bit to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Execute a NOP.
14. Re-enable interrupts.
15. Repeat steps 6-14 seven times to write 64 bytes.
16. Verify the memory (table read).

This procedure will require about 18 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 6-3.

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

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

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

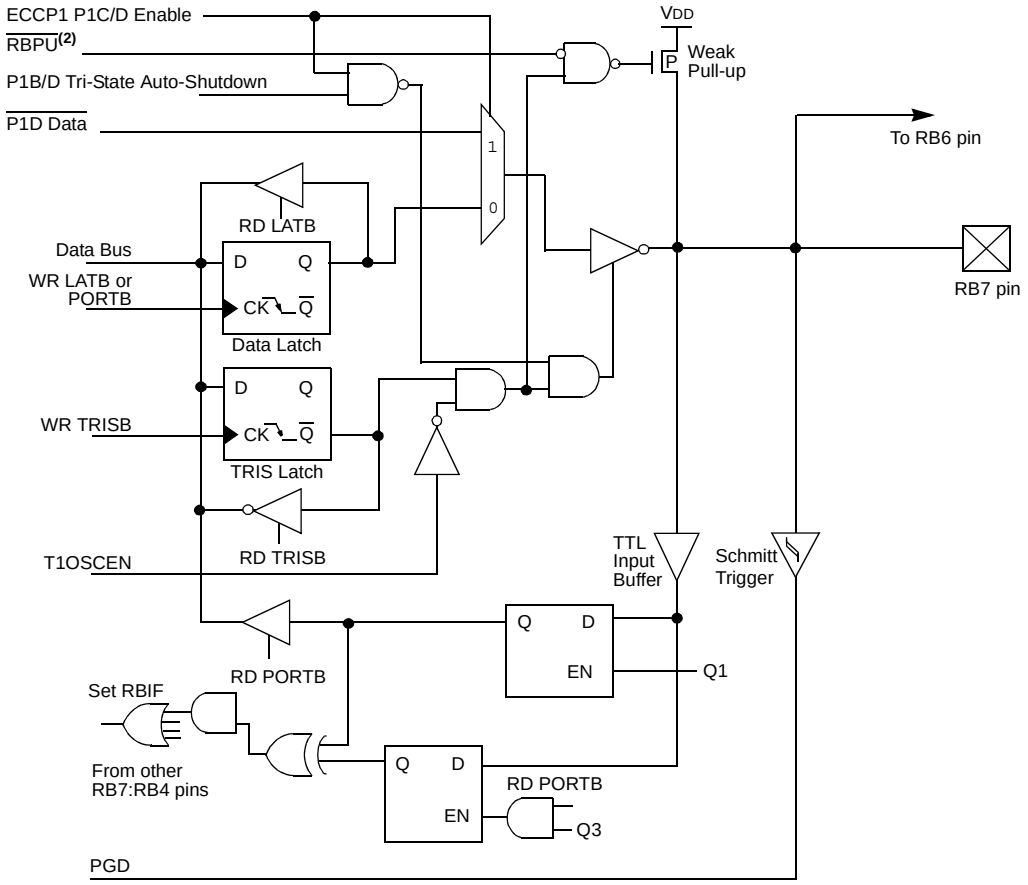
bit 2 **Unimplemented:** Read as ‘0’

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

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

FIGURE 10-14: BLOCK DIAGRAM OF RB7/PGD/T1OSI/P1D/KBI3 PIN



Note 1: I/O pins have diode protection to VDD and VSS.

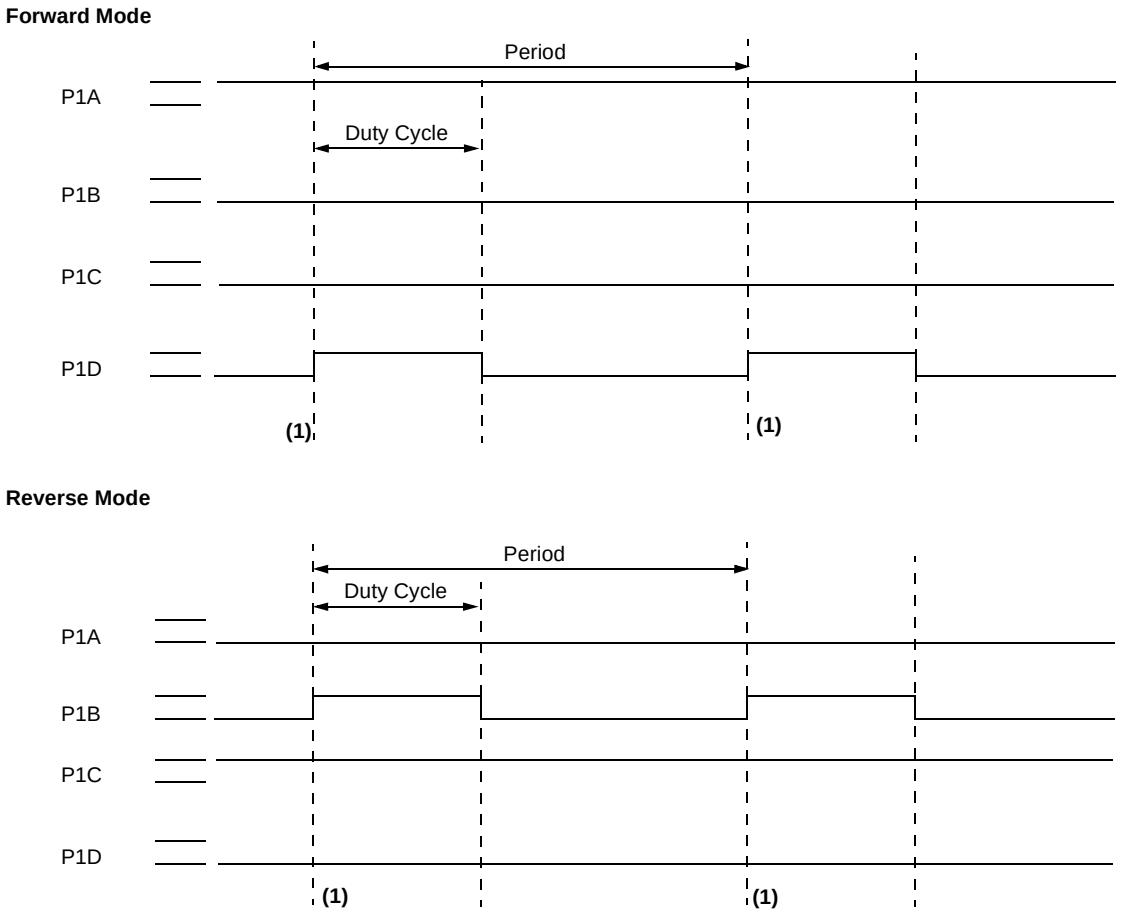
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (INTCON2<7>).

15.5.5 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 RB3/CCP1/P1A is continuously active and pin RB7/PGD/T1OSI/P1D/KBI3 is modulated. In the Reverse mode, pin RB6/PGC/T1OSO/T13CKI/P1C/KBI2 is continuously active and pin RB2/P1B/INT2 is modulated. These are illustrated in Figure 15-8.

The TRISB<3:2> and TRISB<7:6> bits must be cleared to make the P1A, P1B, P1C and P1D pins output.

FIGURE 15-8: FULL-BRIDGE PWM OUTPUT (ACTIVE-HIGH)



Note 1: At this time, the TMR2 register is equal to the PR2 register.

15.5.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the ECCP1 module for PWM operation:

1. Configure the PWM pins P1A and P1B (and P1C and P1D, if used) as inputs by setting the corresponding TRISB bits.
2. Set the PWM period by loading the PR2 register.
3. Configure the ECCP module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M1:P1M0 bits.
   - Select the polarities of the PWM output signals with the CCP1M3:CCP1M0 bits.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
5. For Half-Bridge Output mode, set the dead-band delay by loading PWM1CON<6:0> with the appropriate value.
6. If auto-shutdown operation is required, load the ECCPAS register:
   - Select the auto-shutdown sources using the ECCPAS<2:0> bits.
   - Select the shutdown states of the PWM output pins using PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits.
   - Set the ECCPASE bit (ECCPAS<7>).
7. If auto-restart operation is required, set the PRSEN bit (PWM1CON<7>).
8. Configure and start TMR2:
   - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
   - Set the TMR2 prescale value by loading the T2CKPS bits (T2CON<1:0>).
   - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
9. Enable PWM outputs after a new PWM cycle has started:
   - Wait until TMR2 overflows (TMR2IF bit is set).
   - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRISB bits.
   - Clear the ECCPASE bit (ECCPAS<7>).

15.5.10 OPERATION IN LOW-POWER MODES

In the Low-Power Sleep mode, all clock sources are disabled. Timer2 will not increment and the state of the module will not change. If the ECCP pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency may not be stable if the INTOSC is being used.

In PRI_IDLE mode, the primary clock will continue to clock the ECCP module without change.

In all other low-power modes, the selected low-power mode clock will clock Timer2. Other low-power mode clocks will most likely be different than the primary clock frequency.

15.5.10.1 Operation with Fail-Safe Clock Monitor

If the Fail-Safe Clock Monitor is enabled (CONFIG1H<6> is programmed), a clock failure will force the device into the Low-Power RC_RUN mode and the OSCFIF bit (PIR2<7>) will be set. The ECCP will then be clocked from the INTRC clock source, which may have a different clock frequency than the primary clock. By loading the IRCF2:IRCF0 bits on Resets, the user can enable the INTOSC at a high clock speed in the event of a clock failure.

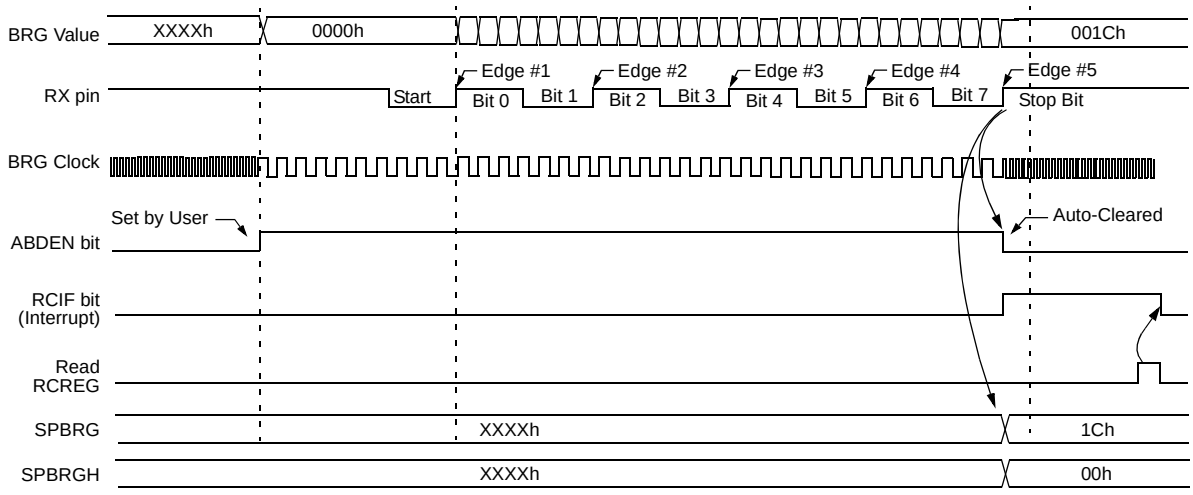
See the previous section for additional details.

15.5.11 EFFECTS OF A RESET

Both power-on and subsequent Resets will force all ports to input mode and the CCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.

FIGURE 16-1: AUTOMATIC BAUD RATE CALCULATION



Note 1: The ABD sequence requires the EUSART module to be configured in Asynchronous mode and WUE = 0.

16.3 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA<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 eight 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 (TXSTA<2> and BAUDCTL<3>). Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

Asynchronous mode is available in all low-power modes; it is available in Sleep mode only when auto-wake-up on Sync Break is enabled. When in PRI_IDLE mode, no changes to the Baud Rate Generator values are required; however, other low-power mode clocks may operate at another frequency than the primary clock. Therefore, the Baud Rate Generator values may need to be adjusted.

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

16.3.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 16-2. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one T_{CY}), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit, TXIF, will be set, regardless of the state of enable bit, TXIE, and cannot be cleared in software. Flag bit, TXIF, is not cleared immediately upon loading the Transmit Buffer register, TXREG. TXIF becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results.

While flag bit, TXIF, indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit, TRMT, is a read-only bit, which is set when the TSR 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 TSR register is empty.

Note 1:	The TSR register is not mapped in data memory, so it is not available to the user.
2:	Flag bit, TXIF, is set when enable bit, TXEN, is set.

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AV_{DD} and AV_{SS}), or the voltage level on the RA3/AN3/V_{REF+} and RA2/AN2/V_{REF-} pins.

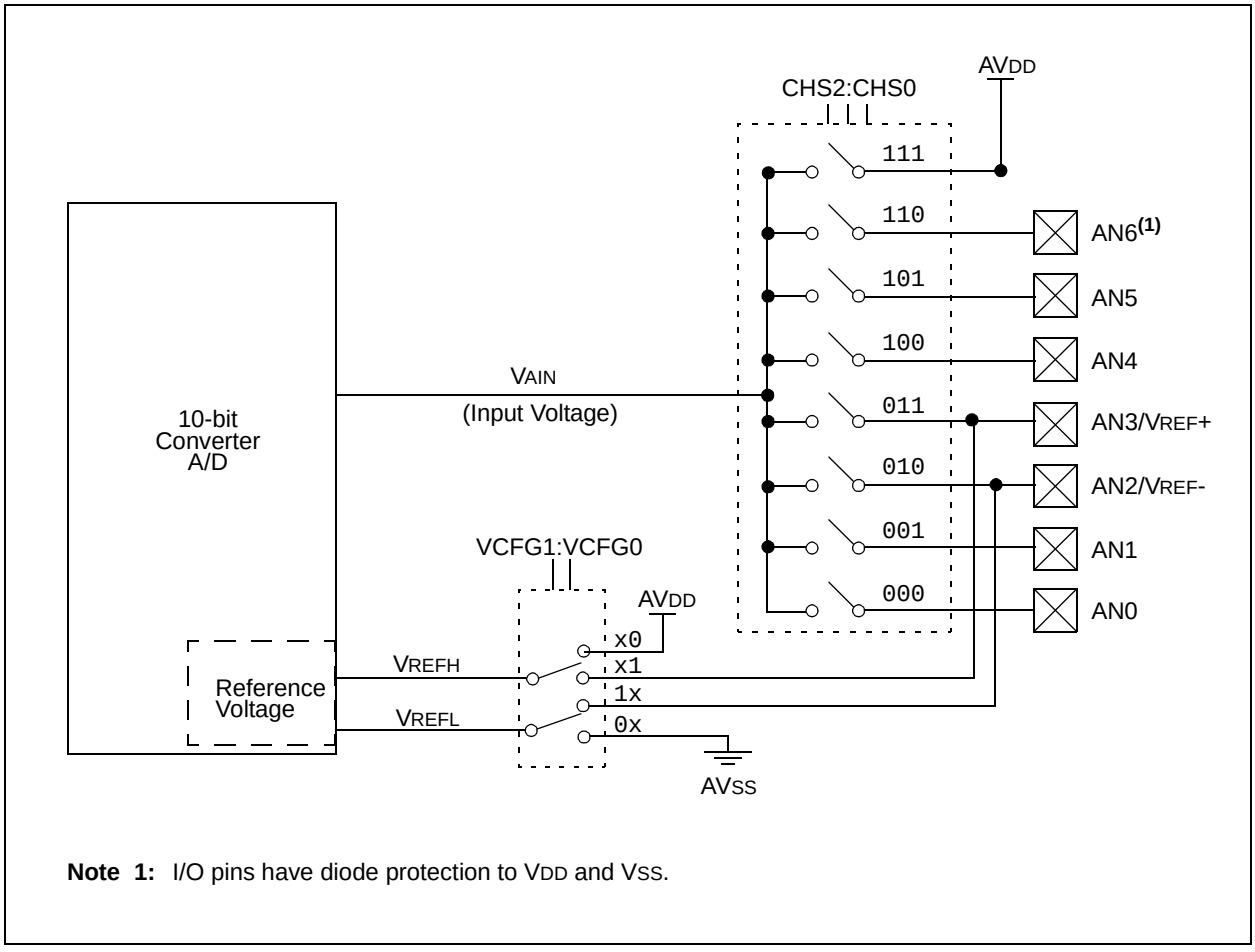
The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted.

Each port pin associated with the A/D converter can be configured as an analog input, or as a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ADRESL registers, the GO/$\overline{DONE}$ bit (ADCON0 register) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 17-1.

FIGURE 17-1: A/D BLOCK DIAGRAM



Note 1: I/O pins have diode protection to V_{DD} and V_{SS}.

17.8 Use of the CCP1 Trigger

An A/D conversion can be started by the "special event trigger" of the CCP1 module. This requires that the CCP1M3:CCP1M0 bits (CCP1CON<3:0>) be programmed as '1011' and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D acquisition and conversion and the Timer1 (or Timer3) counter will be reset to zero. Timer1 (or Timer3) is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving ADRESH/ADRESL to the desired location). The appropriate analog input channel must be selected and the minimum acquisition period is either timed by the user, or an appropriate TACQ time selected before the "special event trigger" sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), the "special event trigger" will be ignored by the A/D module, but will still reset the Timer1 (or Timer3) counter.

TABLE 17-2: SUMMARY OF A/D REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	TMR3IF	—	0--0 -00-	0--0 -00-
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	TMR3IE	—	0--0 -00-	0--0 -00-
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	TMR3IP	—	1--1 -11-	1--1 -11-
ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
ADCON0	VCFG1	VCFG0	—	CHS2	CHS1	CHS0	GO/DONE	ADON	00-0 0000	00-0 0000
ADCON1	—	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	-000 0000	-000 0000
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	0-00 0000
PORTA	RA7(3)	RA6(2)	RA5(1)	RA4	RA3	RA2	RA1	RA0	qq0x 0000	uu0u 0000
TRISA	TRISA7(3)	TRISA6(2)	—	PORTA Data Direction Register					qq-1 1111	11-1 1111
PORTB	Read PORTB pins, Write LATB Latch								xxxx xxxx	uuuu uuuu
TRISB	PORTB Data Direction Register								1111 1111	1111 1111
LATB	PORTB Output Data Latch								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, q = depends on CONFIG1H<3:0>, – = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: RA5 port bit is available only as an input pin when the MCLRE bit in the Configuration register is '0'.

2: RA6 and TRISA6 are available only when the primary oscillator mode selection offers RA6 as a port pin; otherwise, RA6 always reads '0', TRISA6 always reads '1' and writes to both are ignored (see CONFIG1H<3:0>).

3: RA7 and TRISA7 are available only when the internal RC oscillator is configured as the primary oscillator in CONFIG1H<3:0>; otherwise, RA7 always reads '0', TRISA7 always reads '1' and writes to both are ignored.

18.0 LOW-VOLTAGE DETECT

In many applications, the ability to determine if the device voltage (V_{DD}) is below a specified voltage level is a desirable feature. A window of operation for the application can be created, where the application software can do “housekeeping tasks”, before the device voltage exits the valid operating range. This can be done using the Low-Voltage Detect module.

This module is a software programmable circuitry, where a device voltage trip point can be specified. When the voltage of the device becomes lower then the specified point, an interrupt flag is set. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to that interrupt source.

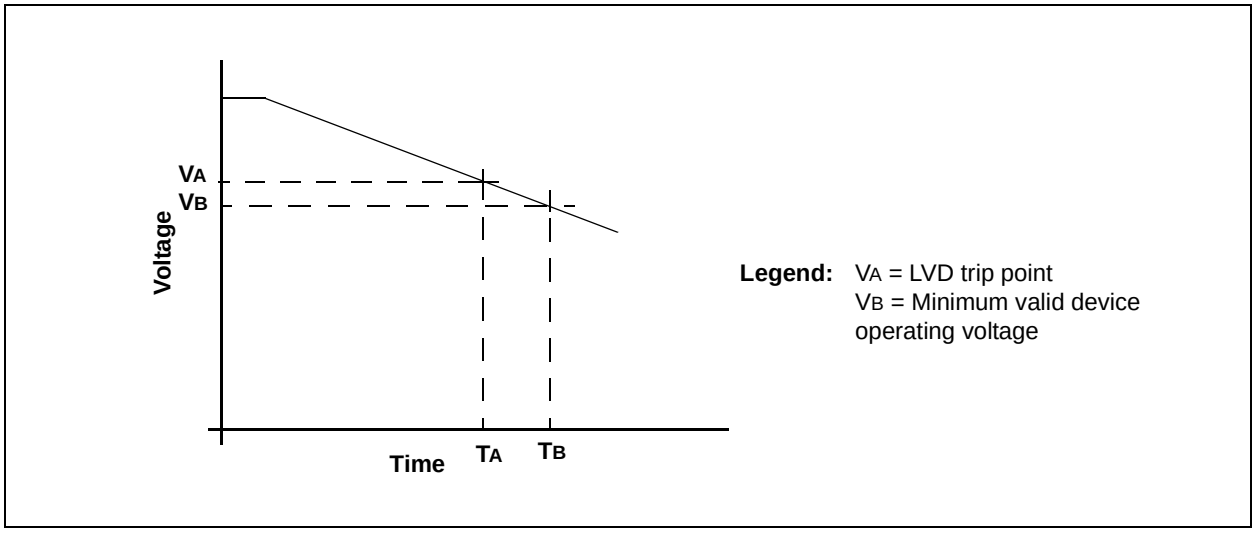
The Low-Voltage Detect circuitry is completely under software control. This allows the circuitry to be turned off by the software, which minimizes the current consumption for the device.

Figure 18-1 shows a possible application voltage curve (typically for batteries). Over time, the device voltage decreases. When the device voltage equals voltage V_A, the LVD logic generates an interrupt. This occurs at time T_A. The application software then has the time, until the device voltage is no longer in valid operating range, to shut down the system. Voltage point V_B is the minimum valid operating voltage specification. This occurs at time T_B. The difference, $T_B - T_A$, is the total time for shutdown.

The block diagram for the LVD module is shown in Figure 18-2 (following page). A comparator uses an internally generated reference voltage as the set point. When the selected tap output of the device voltage crosses the set point (is lower than), the LVDIF bit is set.

Each node in the resistor divider represents a “trip point” voltage. The “trip point” voltage is the minimum supply voltage level at which the device can operate before the LVD module asserts an interrupt. When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the 1.2V internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal setting the LVDIF bit. This voltage is software programmable to any one of 16 values (see Figure 18-2). The trip point is selected by programming the LVDL3:LVDL0 bits (LVDCON<3:0>).

FIGURE 18-1: TYPICAL LOW-VOLTAGE DETECT APPLICATION



REGISTER 19-1: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
IESO	FSCM	—	—	FOSC3	FOSC2	FOSC1	FOSC0
bit 7							bit 0

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

bit 7 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode enabled
0 = Internal External Switchover mode disabled

bit 6 **FSCM:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3-0 **FOSC<3:0>:** Oscillator Selection bits
11xx = External RC oscillator, CLKO function on RA6
1001 = Internal RC oscillator, CLKO function on RA6 and port function on RA7
1000 = Internal RC oscillator, port function on RA6 and port function on RA7
0111 = External RC oscillator, port function on RA6
0110 = HS oscillator, PLL enabled (clock frequency = 4 x FOSC1)
0101 = EC oscillator, port function on RA6
0100 = EC oscillator, CLKO function on RA6
0010 = HS oscillator
0001 = XT oscillator
0000 = LP oscillator

BRA **Unconditional Branch**

Syntax: [*label*] BRA n

Operands: -1024 ≤ n ≤ 1023

Operation: (PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1101	0nnn	nnnn	nnnn

Description: Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a 2-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: `HERE BRA Jump`

Before Instruction
PC = address (HERE)
After Instruction
PC = address (Jump)

BSF **Bit Set f**

Syntax: [*label*] BSF f,b[,a]

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

Operation: 1 → f<b>

Status Affected: None

Encoding:

1000	bbba	ffff	ffff

Description: Bit 'b' in register 'f' is set. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: `BSF FLAG_REG, 7`

Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC **Bit Test File, Skip if Clear**

Syntax: [*label*] BTFSC f,b[,a]

Operands: $0 \le f \le 255$
$0 \le b \le 7$
$a \in [0,1]$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1011	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped.
If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    BTFSC   FLAG, 1
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (TRUE)
If FLAG<1> = 1;
PC = address (FALSE)

BTFSS **Bit Test File, Skip if Set**

Syntax: [*label*] BTFSS f,b[,a]

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

1010	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    BTFSS   FLAG, 1
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (FALSE)
If FLAG<1> = 1;
PC = address (TRUE)

CPFSGT **Compare f with W, skip if f > W**

Syntax: [*label*] CPFSGT f [,a]

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

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	010a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE      CPFSGT REG
NGREATER  :
GREATER   :
```

Before Instruction
```
PC  =  Address (HERE)
W   =  ?
```
After Instruction
```
If REG  >  W;
    PC  =  Address (GREATER)
If REG  ≤  W;
    PC  =  Address (NGREATER)
```

CPFSLT **Compare f with W, skip if f < W**

Syntax: [*label*] CPFSLT f [,a]

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

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. If 'a' is '0', the Access Bank will be selected. If 'a' is '1', the BSR will not be overridden (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

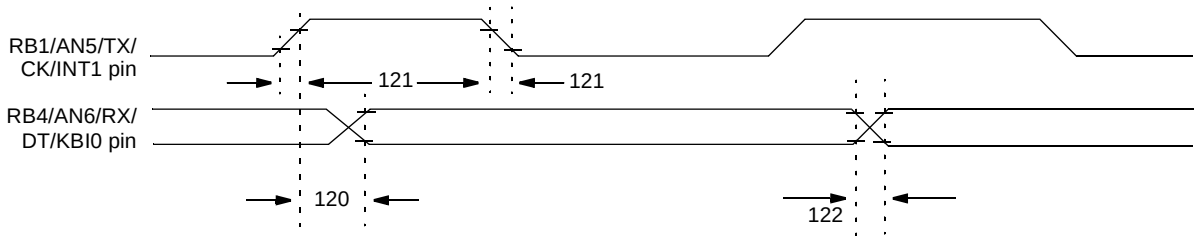
Example:
```
HERE    CPFSLT REG
NLESS   :
LESS    :
```

Before Instruction
```
PC  =  Address (HERE)
W   =  ?
```
After Instruction
```
If REG  <  W;
    PC  =  Address (LESS)
If REG  ≥  W;
    PC  =  Address (NLESS)
```

TABLE 22-10: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)

Param. No.	Symbol	Characteristic			Min.	Max.	Units	Conditions
50	TccL	CCPx Input Low Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18F1X20	10	—	ns	
				PIC18LF1X20	20	—	ns	
51	TccH	CCPx Input High Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18F1X20	10	—	ns	
				PIC18LF1X20	20	—	ns	
52	TccP	CCPx Input Period			3 TCY + 40 / N	—	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fall Time		PIC18F1X20	—	25	ns	
				PIC18LF1X20	—	45	ns	
54	TccF	CCPx Output Fall Time		PIC18F1X20	—	25	ns	
				PIC18LF1X20	—	45	ns	

FIGURE 22-12: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

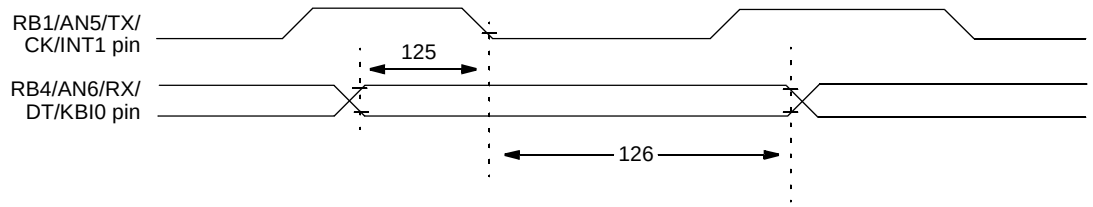


Note: Refer to Figure 22-5 for load conditions.

TABLE 22-11: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC18F1X20	—	40	ns	
			PIC18LF1X20	—	100	ns	
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	PIC18F1X20	—	20	ns	
			PIC18LF1X20	—	50	ns	
122	Tdtrf	Data Out Rise Time and Fall Time	PIC18F1X20	—	20	ns	
			PIC18LF1X20	—	50	ns	

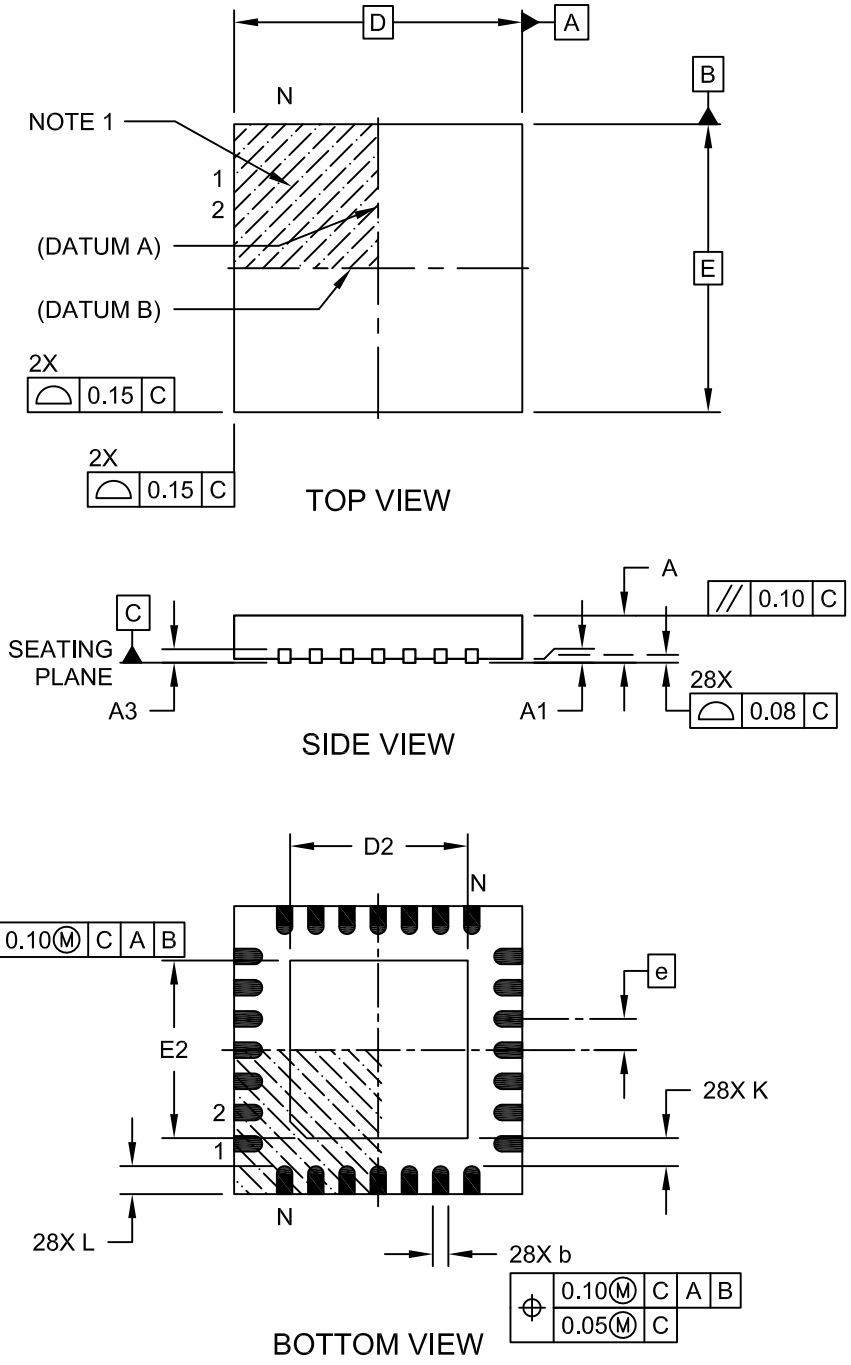
FIGURE 22-13: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 22-5 for load conditions.

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN]
With 0.55 mm Terminal Length

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-105C Sheet 1 of 2