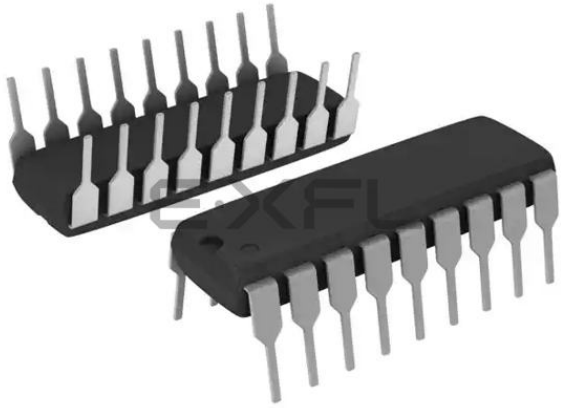
Microchip Technology - PIC18F1220-H/P Datasheet



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	25MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K 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 ~ 150°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/pic18f1220-h-p

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

2.7.1 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 2-2) controls several aspects of the system clock's operation, both in full-power operation and in power managed modes.

The System Clock Select bits, SCS1:SCS0, select the clock source that is used when the device is operating in power managed modes. The available clock sources are the primary clock (defined in Configuration Register 1H), the secondary clock (Timer1 oscillator) and the internal oscillator block. The clock selection has no effect until a `SLEEP` instruction is executed and the device enters a power managed mode of operation. The SCS bits are cleared on all forms of Reset.

The Internal Oscillator Select bits, IRCF2:IRCF0, select the frequency output of the internal oscillator block that is used to drive the system clock. The choices are the INTRC source, the INTOSC source (8 MHz), or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). If the internal oscillator block is supplying the system clock, changing the states of these bits will have an immediate change on the internal oscillator's output.

The OSTS, IOFS and T1RUN bits indicate which clock source is currently providing the system clock. The OSTS indicates that the Oscillator Start-up Timer has timed out and the primary clock is providing the system clock in Primary Clock modes. The IOFS bit indicates when the internal oscillator block has stabilized and is providing the system clock in RC Clock modes or during Two-Speed Start-ups. The T1RUN bit (T1CON<6>) indicates when the Timer1 oscillator is providing the system clock in Secondary Clock modes. In power managed modes, only one of these three bits will be set at any time. If none of these bits are set, the INTRC is providing the system clock, or the internal oscillator block has just started and is not yet stable.

The IDLEN bit controls the selective shutdown of the controller's CPU in power managed modes. The uses of these bits are discussed in more detail in **Section 3.0 "Power Managed Modes"**.

Note 1: The Timer1 oscillator must be enabled to select the secondary clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select a secondary clock source when executing a `SLEEP` instruction will be ignored.

2: It is recommended that the Timer1 oscillator be operating and stable before executing the `SLEEP` instruction or a very long delay may occur while the Timer1 oscillator starts.

FIGURE 2-8: PIC18F1220/1320 CLOCK DIAGRAM

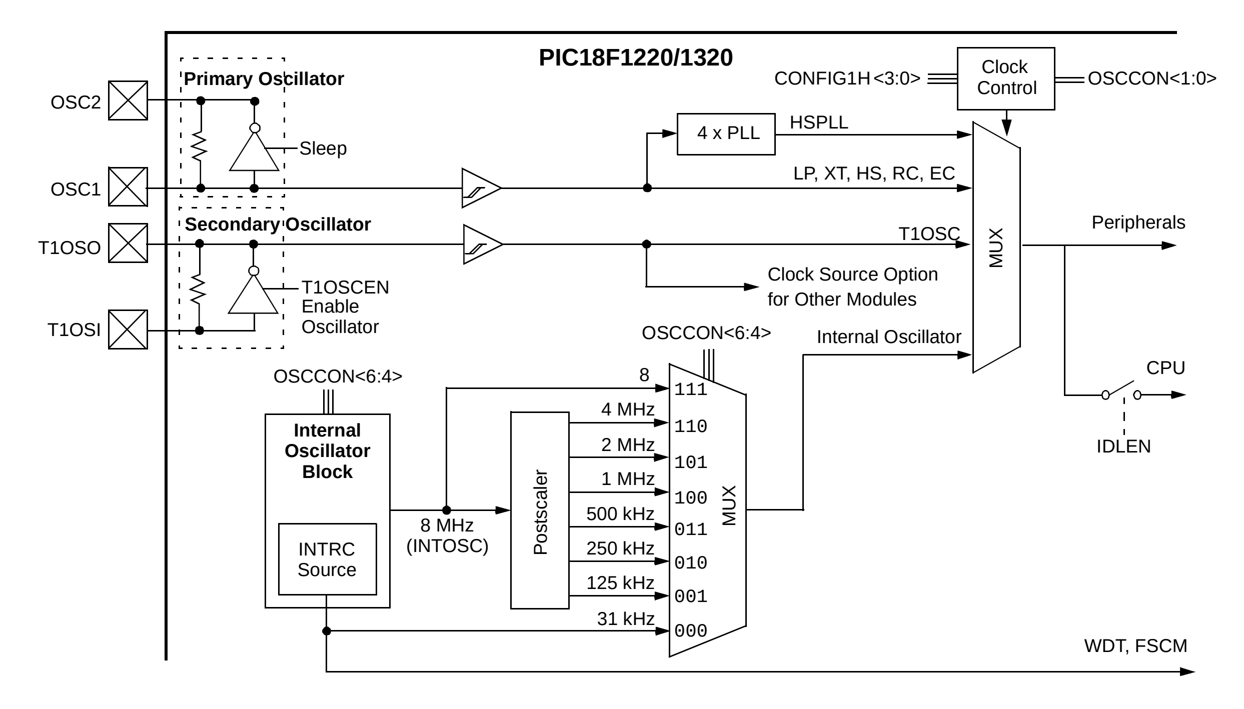


FIGURE 3-1: TIMING TRANSITION FOR ENTRY TO SLEEP MODE

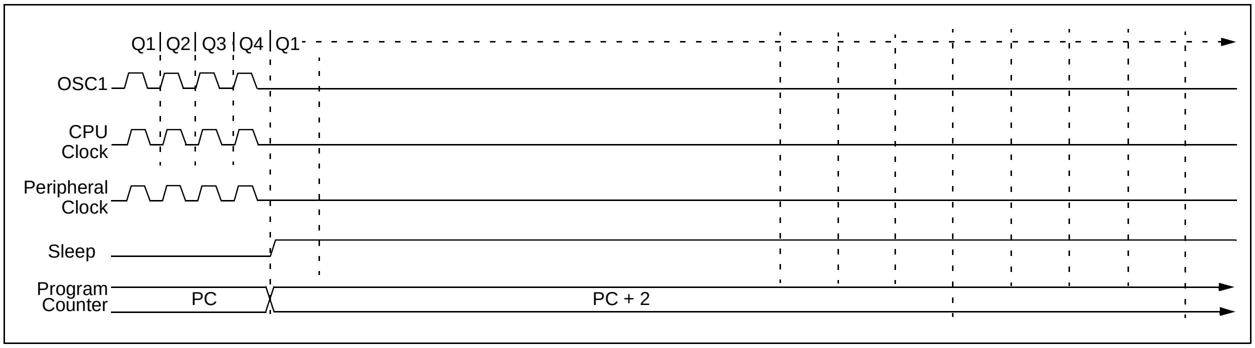
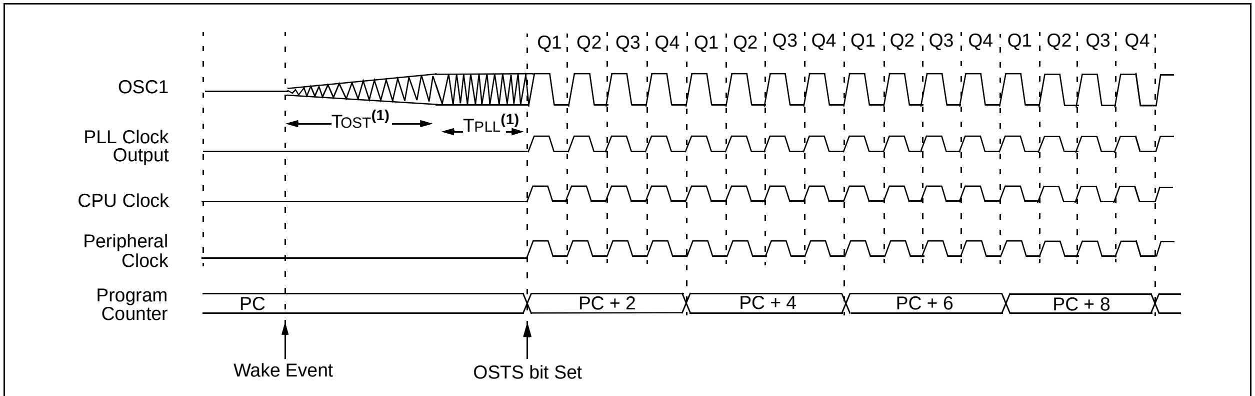


FIGURE 3-2: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

3.4.4 EXIT TO IDLE MODE

An exit from a power managed Run mode to its corresponding Idle mode is executed by setting the IDLEN bit and executing a SLEEP instruction. The CPU is halted at the beginning of the instruction following the SLEEP instruction. There are no changes to any of the clock source status bits (OSTS, IOFS or T1RUN). While the CPU is halted, the peripherals continue to be clocked from the previously selected clock source.

3.4.5 EXIT TO SLEEP MODE

An exit from a power managed Run mode to Sleep mode is executed by clearing the IDLEN and SCS1:SCS0 bits and executing a SLEEP instruction. The code is no different than the method used to invoke Sleep mode from the normal operating (full-power) mode.

The primary clock and internal oscillator block are disabled. The INTRC will continue to operate if the WDT is enabled. The Timer1 oscillator will continue to run, if enabled in the T1CON register (Register 12-1). All clock source Status bits are cleared (OSTS, IOFS and T1RUN).

3.5 Wake from Power Managed Modes

An exit from any of the power managed modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power managed modes. The clocking subsystem actions are discussed in each of the power managed modes (see Sections 3.2 through 3.4).

Note:	If application code is timing sensitive, it should wait for the OSTS bit to become set before continuing. Use the interval during the low-power exit sequence (before OSTS is set) to perform timing insensitive “housekeeping” tasks.

Device behavior during Low-Power mode exits is summarized in Table 3-3.

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit a power managed mode and resume full-power operation. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set. On all exits from Low-Power mode by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

REGISTER 5-1: STKPTR: STACK POINTER REGISTER

R/C-0/0	R/C-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
STKFUL[(1)]	STKUNF[(1)]	—	SP<4:0>				
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	C = Clearable only bit

bit 7 **STKFUL:** Stack Full Flag bit[(1)]
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit[(1)]
1 = Stack Underflow occurred
0 = Stack Underflow did not occur

bit 5 **Unimplemented:** Read as ‘0’

bit 4-0 **SP<4:0>:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

5.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable option. To push the current PC value onto the stack, a PUSH instruction can be executed. This will increment the Stack Pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place data or a return address on the stack.

The ability to pull the TOS value off of the stack and replace it with the value that was previously pushed onto the stack, without disturbing normal execution, is achieved by using the POP instruction. The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

5.2.4 STACK FULL/UNDERFLOW RESETS

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

6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A “Bulk Erase” operation may not be issued from user code.

While writing or erasing program memory, instruction fetches cease until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

The program memory space is 16 bits wide, while the data RAM space is eight bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

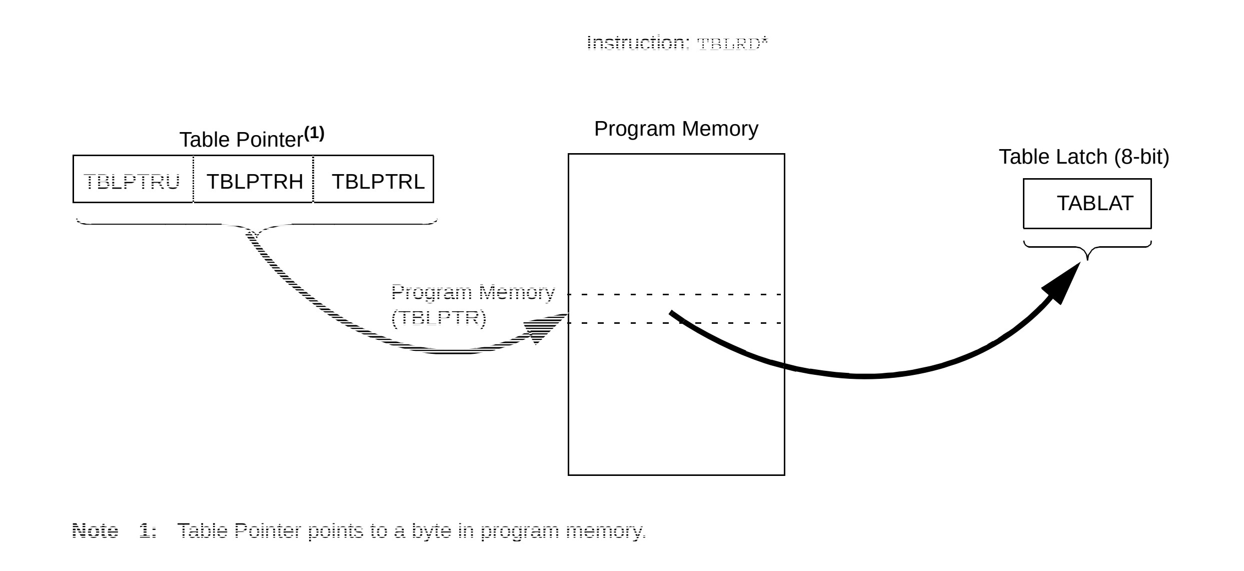
Table read operations retrieve data from program memory and place it into TABLAT in the data RAM space. Figure 6-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from TABLAT in the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.5 “Writing to Flash Program Memory”**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned (TBLPTRL<0> = 0).

The EEPROM on-chip timer controls the write and erase times. The write and erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

FIGURE 6-1: TABLE READ OPERATION



Note 1: Table Pointer points to a byte in program memory.

12.7 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 12.2 “Timer1 Oscillator”**, above), gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, `RTCisr`, shown in Example 12-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine, which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflow.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take two seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1), as shown in the routine, `RTCinit`. The Timer1 oscillator must also be enabled and running at all times.

EXAMPLE 12-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```
RTCinit
           MOVLW   0x80                ; Preload TMR1 register pair
           MOVWF   TMR1H               ; for 1 second overflow
           CLRF    TMR1L
           MOVLW   b’00001111’         ; Configure for external clock,
           MOVWF   T1OSC               ; Asynchronous operation, external oscillator
           CLRF    secs                ; Initialize timekeeping registers
           CLRF    mins                ;
           MOVLW   .12
           MOVWF   hours
           BSF     PIE1, TMR1IE        ; Enable Timer1 interrupt
           RETURN
RTCisr
           BSF     TMR1H, 7            ; Preload for 1 sec overflow
           BCF     PIR1, TMR1IF        ; Clear interrupt flag
           INCF    secs, F             ; Increment seconds
           MOVLW   .59                 ; 60 seconds elapsed?
           CPFSGT  secs
           RETURN                      ; No, done
           CLRF    secs                ; Clear seconds
           INCF    mins, F             ; Increment minutes
           MOVLW   .59                 ; 60 minutes elapsed?
           CPFSGT  mins
           RETURN                      ; No, done
           CLRF    mins                ; clear minutes
           INCF    hours, F            ; Increment hours
           MOVLW   .23                 ; 24 hours elapsed?
           CPFSGT  hours
           RETURN                      ; No, done
           MOVLW   .01                 ; Reset hours to 1
           MOVWF   hours
           RETURN                      ; Done
```

13.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

Timer2 has a control register shown in Register 13-1. TMR2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption. Figure 13-1 is a simplified block diagram of the Timer2 module. Register 13-1 shows the Timer2 Control register. The prescaler and postscaler selection of Timer2 are controlled by this register.

13.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 register is readable and writable and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit, TMR2IF (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 **Unimplemented:** Read as ‘0’

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

14.1 Timer3 Operation

Timer3 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

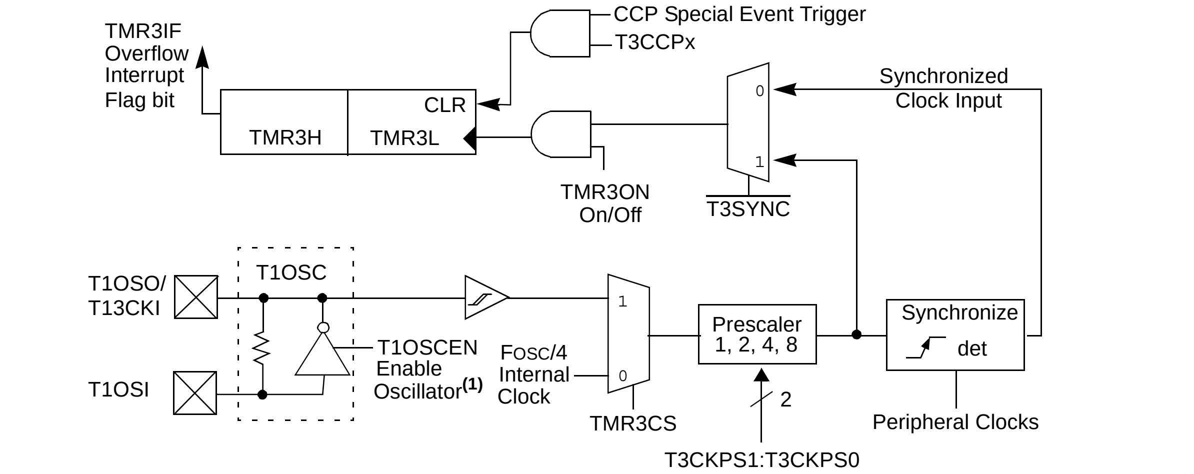
The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>).

When TMR3CS = 0, Timer3 increments every instruction cycle. When TMR3CS = 1, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RB7/PGD/T1OSI/P1D/KBI3 and RB6/PGC/T1OSO/T13CKI/P1C/KBI2 pins become inputs. That is, the TRISB7:TRISB6 value is ignored and the pins are read as ‘0’.

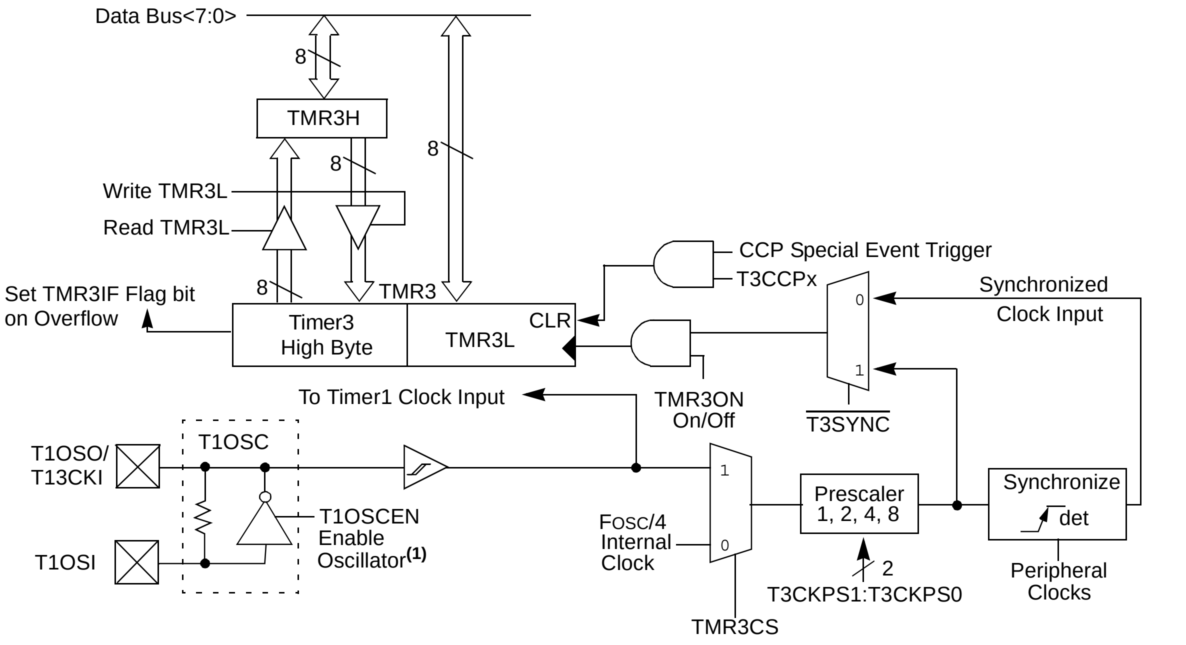
Timer3 also has an internal “Reset input”. This Reset can be generated by the CCP module (see **Section 15.4.4 “Special Event Trigger”**).

FIGURE 14-1: TIMER3 BLOCK DIAGRAM



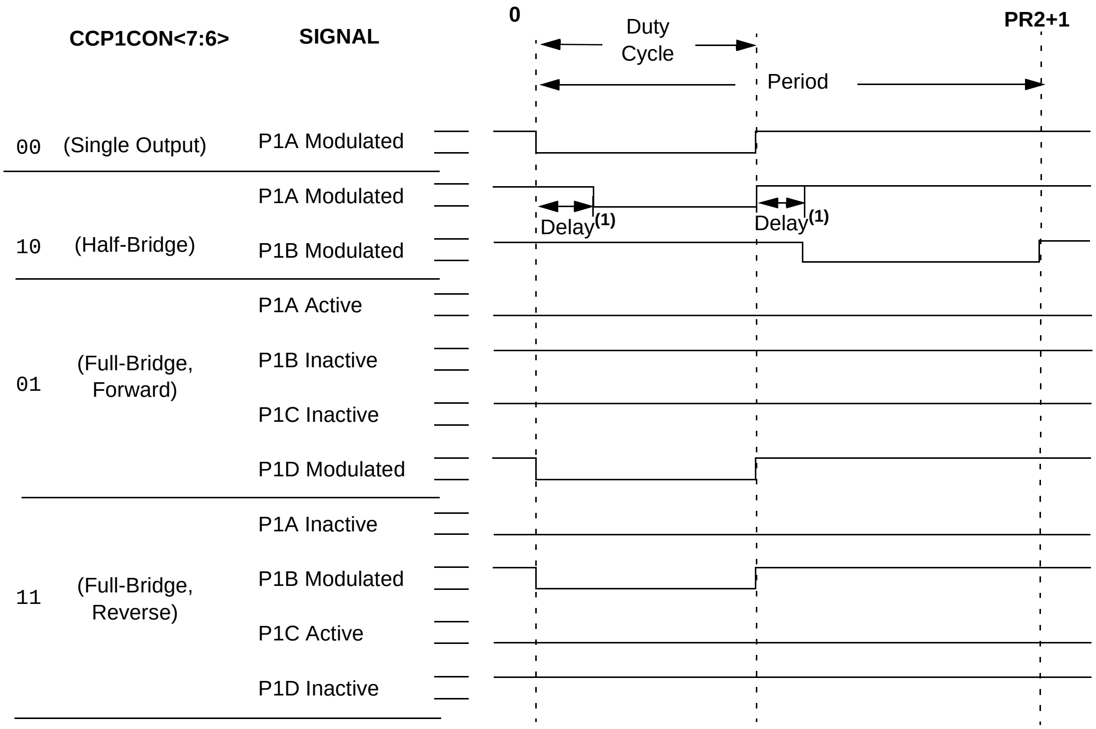
Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

FIGURE 14-2: TIMER3 BLOCK DIAGRAM CONFIGURED IN 16-BIT READ/WRITE MODE



Note 1: When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

FIGURE 15-5: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)

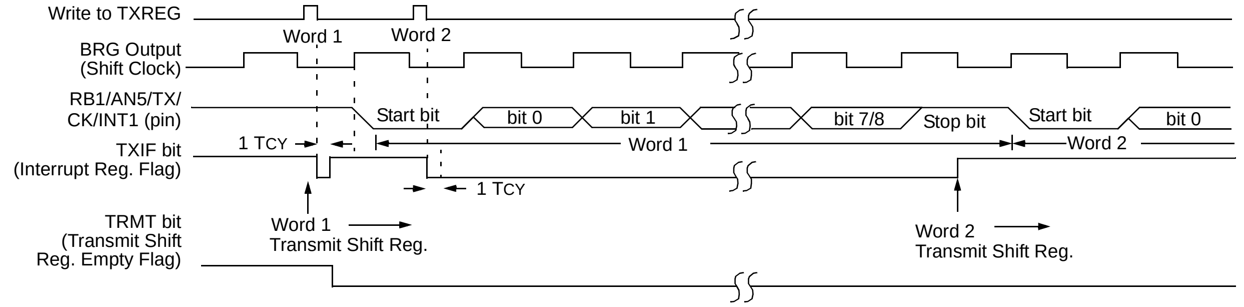


Relationships:

- Period = 4 * TOSC * (PR2 + 1) * (TMR2 Prescale Value)
- Duty Cycle = TOSC * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * TOSC * (PWM1CON<6:0>)

Note 1: Dead-band delay is programmed using the PWM1CON register (**Section 15.5.6 “Programmable Dead-Band Delay”**).

FIGURE 16-4: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)


Write to TXREG
Word 1
Word 2
BRG Output (Shift Clock)
RB1/AN5/TX/ CK/INT1 (pin)
Start bit
bit 0
bit 1
bit 7/8
Stop bit
Start bit
bit 0
TXIF bit (Interrupt Reg. Flag)
1 TCY
Word 1
Word 2
1 TCY
TRMT bit (Transmit Shift Reg. Empty Flag)
Word 1 Transmit Shift Reg.
Word 2 Transmit Shift Reg.

Note: This timing diagram shows two consecutive transmissions.

TABLE 16-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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 000x	0000 000u
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
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	EUSART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	Baud Rate Generator Register High Byte								0000 0000	0000 0000
SPBRG	Baud Rate Generator Register Low Byte								0000 0000	0000 0000

Legend: x = unknown, – = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

16.4.2 EUSART SYNCHRONOUS MASTER RECEPTION

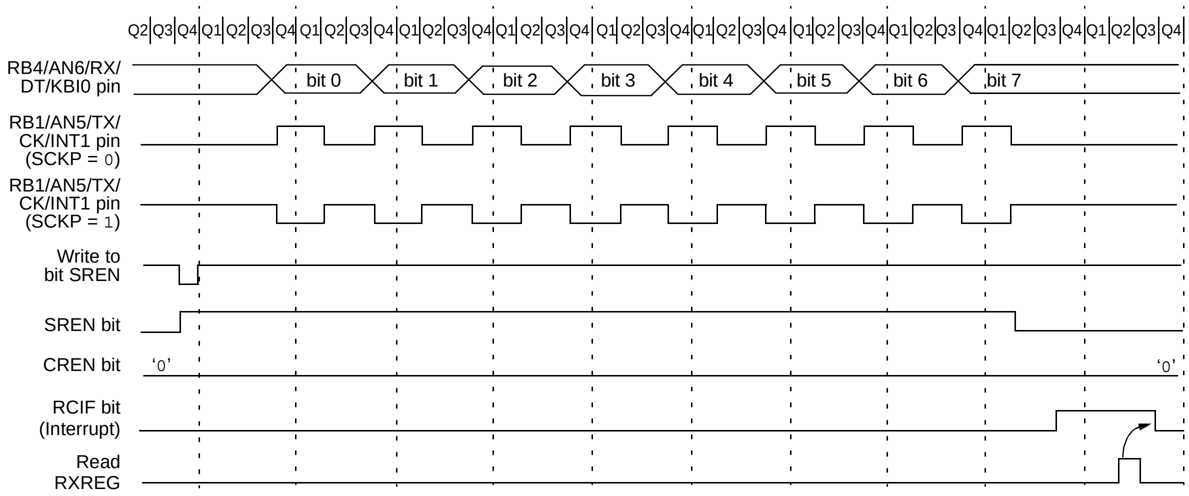
Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RB4/AN6/RX/DT/KBI0 pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH:SPBRG 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 synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set enable bit RCIE.
5. If 9-bit reception is desired, set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 16-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)


Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
RB4/AN6/RX/DT/KBI0 pin
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
RB1/AN5/TX/CK/INT1 pin (SCKP = 0)
RB1/AN5/TX/CK/INT1 pin (SCKP = 1)
Write to bit SREN
SREN bit
CREN bit
'0'
'0'
RCIF bit (Interrupt)
Read RXREG

Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

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/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$	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.

REGISTER 19-4: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

R/P-1	U-0	U-0	U-0	U-0	U-0	U-0	U-0
MCLRE	—	—	—	—	—	—	—
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 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit

1 = $\overline{\text{MCLR}}$ pin enabled, RA5 input pin disabled
0 = RA5 input pin enabled, $\overline{\text{MCLR}}$ disabled

bit 6-0 **Unimplemented:** Read as ‘0’

REGISTER 19-5: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVR
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 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit (see note)

1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6-3 **Unimplemented:** Read as ‘0’

bit 2 **LVP:** Low-Voltage ICSP Enable bit

1 = Low-Voltage ICSP enabled
0 = Low-Voltage ICSP disabled

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **STVR:** Stack Full/Underflow Reset Enable bit

1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

Note: The Timer1 oscillator shares the T1OSI and T1OSO pins with the PGD and PGC pins used for programming and debugging.

When using the Timer1 oscillator, In-Circuit Serial Programming (ICSP) may not function correctly (high voltage or low voltage), or the In-Circuit Debugger (ICD) may not communicate with the controller. As a result of using either ICSP or ICD, the Timer1 crystal may be damaged.

If ICSP or ICD operations are required, the crystal should be disconnected from the circuit (disconnect either lead) or installed after programming. The oscillator loading capacitors may remain in-circuit during ICSP or ICD operation.

REGISTER 19-12: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F1220/1320 DEVICES

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’
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 7-5 **DEV<2:0>:** Device ID bits

`111` = PIC18F1220
`110` = PIC18F1320

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

REGISTER 19-13: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F1220/1320 DEVICES

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
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 7-0 **DEV<10:3>:** Device ID bits

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

`0000 0111` = PIC18F1220/1320 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.

CLRF — Clear f

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

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

Operation: 000h → f
1 → Z

Status Affected: Z

Encoding:

`0110`	`101a`	`ffff`	`ffff`

Description: Clears the contents of the specified register. 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

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: `CLRF FLAG_REG`

Before Instruction

FLAG_REG	=	0x5A

After Instruction

FLAG_REG	=	0x00

CLRWDT — Clear Watchdog Timer

Syntax: [*label*] CLRWDT

Operands: None

Operation: 000h → WDT,
000h → WDT postscaler,
1 → $\overline{\text{TO}}$,
1 → $\overline{\text{PD}}$

Status Affected: $\overline{\text{TO}}$, $\overline{\text{PD}}$

Encoding:

`0000`	`0000`	`0000`	`0100`

Description: `CLRWDT` instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, $\overline{\text{TO}}$ and $\overline{\text{PD}}$, are set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: `CLRWDT`

Before Instruction

WDT Counter	=	?

After Instruction

WDT Counter	=	0x00
WDT Postscaler	=	0
$\overline{\text{TO}}$	=	1
$\overline{\text{PD}}$	=	1

MOVFF — Move f to f

Syntax: [*label*] MOVFF f_s,f_d

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)	1100	ffff	ffff	$\mathtt{ffff}_s$
2nd word (destin.)	1111	ffff	ffff	$\mathtt{ffff}_d$

Description: The contents of source register ‘f_s’ are moved to destination register ‘f_d’. Location of source ‘f_s’ can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination ‘f_d’ can also be anywhere from 000h to FFFh. Either source or destination can be W (a useful special situation). `MOVFF` is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).

The `MOVFF` instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

The `MOVFF` instruction should not be used to modify interrupt settings while any interrupt is enabled (see page 70).

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’ (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register ‘f’ (dest)

Example: `MOVFF   REG1, REG2`

Before Instruction

REG1	=	0x33
REG2	=	0x11

After Instruction

REG1	=	0x33,
REG2	=	0x33

MOVLB — Move literal to low nibble in BSR

Syntax: [*label*] MOVLB k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow$ BSR

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk

Description: The 8-bit literal ‘k’ is loaded into the Bank Select Register (BSR).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write literal ‘k’ to BSR

Example: `MOVLB   5`

Before Instruction

BSR register	=	0x02

After Instruction

BSR register	=	0x05

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial) (Continued)

PIC18LF1220/1320 (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F1220/1320 (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (IDD)[2,3]						
	PIC18LF1220/1320	4.7	8	µA	-40°C	VDD = 2.0V	FOSC = 31 kHz (**RC_IDLE** mode, Internal oscillator source)
		5.0	8	µA	+25°C		
		5.8	11	µA	+85°C		
	PIC18LF1220/1320	7.0	11	µA	-40°C	VDD = 3.0V	
		7.8	11	µA	+25°C		
		8.7	15	µA	+85°C		
	All devices	12	16	µA	-40°C	VDD = 5.0V	
		14	16	µA	+25°C		
		14	22	µA	+85°C		
	Extended devices	25	75	µA	+125°C		
	PIC18LF1220/1320	75	150	µA	-40°C	VDD = 2.0V	FOSC = 1 MHz (**RC_IDLE** mode, Internal oscillator source)
		85	150	µA	+25°C		
		95	150	µA	+85°C		
	PIC18LF1220/1320	110	180	µA	-40°C	VDD = 3.0V	
		125	180	µA	+25°C		
		135	180	µA	+85°C		
	All devices	180	380	µA	-40°C	VDD = 5.0V	
		195	380	µA	+25°C		
		200	380	µA	+85°C		
	Extended devices	350	435	µA	+125°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

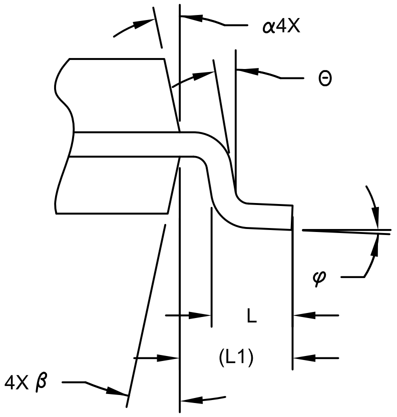
2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
$\overline{\text{OSC1}}$ = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

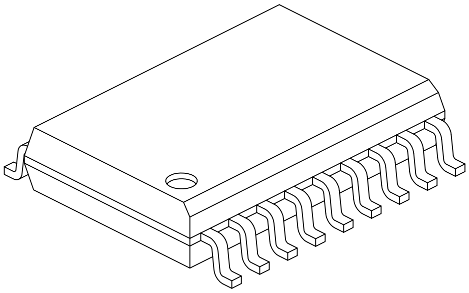
4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

18-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	18		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	11.55 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-051C Sheet 2 of 2

APPENDIX A: REVISION HISTORY

Revision A (August 2002)

Original data sheet for PIC18F1220/1320 devices.

Revision B (November 2002)

This revision includes significant changes to **Section 2.0**, **Section 3.0** and **Section 19.0**, as well as updates to the Electrical Specifications in **Section 22.0** and includes minor corrections to the data sheet text.

Revision C (May 2004)

This revision includes updates to the Electrical Specifications in **Section 22.0**, the DC and AC Characteristics Graphs and Tables in **Section 23.0** and includes minor corrections to the data sheet text.

Revision D (October 2006)

This revision includes updates to the packaging diagrams.

Revision E (January 2007)

This revision includes updates to the packaging diagrams.

Revision F (February 2007)

This revision includes updates to the packaging diagrams.

Revision G (April 2015)

Added Section 22.5: High Temperature Operation in the Electrical Specifications section.

APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1.

TABLE B-1: DEVICE DIFFERENCES

Features	PIC18F1220	PIC18F1320
Program Memory (Bytes)	4096	8192
Program Memory (Instructions)	2048	4096
Interrupt Sources	15	15
I/O Ports	Ports A, B	Ports A, B
Enhanced Capture/Compare/PWM Modules	1	1
10-bit Analog-to-Digital Module	7 input channels	7 input channels
Packages	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN