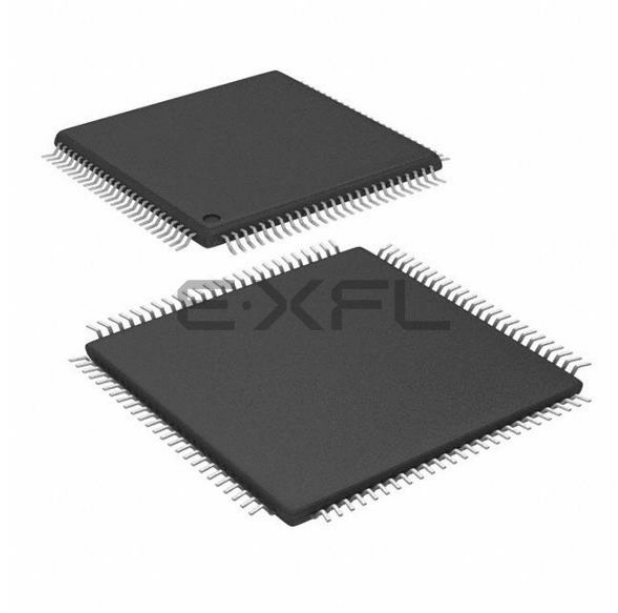


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	86
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f95j94-i-pt

3.3.2 OSCCON3 – CLOCK DIVIDER REGISTER (IRCF<2:0> BITS)

The IRCFx bits (OSCCON3<2:0>) select the postscaler option for the FRC Oscillator output, allowing users to choose a lower clock frequency than the nominal 8 MHz.

This option is described in more detail in **Section 3.10.2 "FRC Postscaler Mode (FRCDIV)"** and **Section 3.10.3 "FRC Oscillator with PLL Mode (FRCPLL)"**.

REGISTER 3-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-1
—	—	—	—	—	IRCF2(1)	IRCF1(1)	IRCF0(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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **IRCF<2:0>:** Reference Clock Divider bits(1)

000 = FRC divide-by-1
001 = FRC divide-by-2 **(default)**
010 = FRC divide-by-4
011 = FRC divide-by-8
100 = FRC divide-by-16
101 = FRC divide-by-32
110 = FRC divide-by-64
111 = FRC divide-by-256

Note 1: The default FRC divide-by setting on an 8-bit device corresponds to 1 MIPS operation.

REGISTER 3-4: OSCCON4: OSCILLATOR CONTROL REGISTER 4

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
CPDIV1	CPDIV0	PLLEN	—	—	—	—	—
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 7-6 **CPDIV<1:0>:** USB System Clock Select bits (postscaler select from 64 MHz clock branch)

00 = Input clock/1
01 = Input clock/2
10 = Input clock/4
11 = Input clock/8

bit 5 **PLLEN:** PLL Enable bit

1 = PLL is enabled even though it is not requested by the CPU; provides ability to "warm-up" the PLL and keep it running to avoid the PLL start-up time. This setting will force the PLL and associated clock source to stay active in Sleep.
0 = PLL is disabled; PLL will be automatically turned on when SRC1 is selected, or when REFO1 or REFO2 is enabled and using the PLL clock as its source. In either case, the PLL will require a start-up time.

bit 4-0 **Unimplemented:** Read as '0'

3.5.1 SELECTING A PRIMARY OSCILLATOR MODE

The main difference between the MS and HS modes is the gain of the internal inverter of the oscillator circuit, which allows the different frequency ranges. The MS mode is a medium power, medium frequency mode. HS mode provides the highest oscillator frequencies with a crystal. OSC2 provides crystal feedback in both HS and MS Oscillator modes.

The EC and HS modes that use the PLL circuit provide the highest device operating frequencies. The oscillator circuit will consume the most current in these modes because the PLL is enabled to multiply the frequency of the oscillator.

In general, users should select the oscillator option with the lowest possible gain that still meets their specifications. This will result in lower dynamic currents (I_{DD}). The frequency range of each oscillator mode is the recommended frequency cutoff, but the selection of a different gain mode is acceptable as long as a thorough validation is performed (voltage, temperature and component variations, such as resistor, capacitor and internal oscillator circuitry).

The oscillator feedback circuit is disabled in all EC modes. The OSC1 pin is a high-impedance input and can be driven by a CMOS driver.

If the Primary Oscillator is configured for an external clock input, the OSC2 pin is not required to support the oscillator function. For these modes, the OSC2 pin can be used as an additional device I/O pin or a clock output pin. When the OSC2 pin is used as a clock output pin, the output frequency is $F_{OSC}/4$.

3.6 Crystal Oscillators and Ceramic Resonators

In MS and HS modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 3-3). The PIC18F oscillator design requires the use of a parallel cut crystal. Using a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

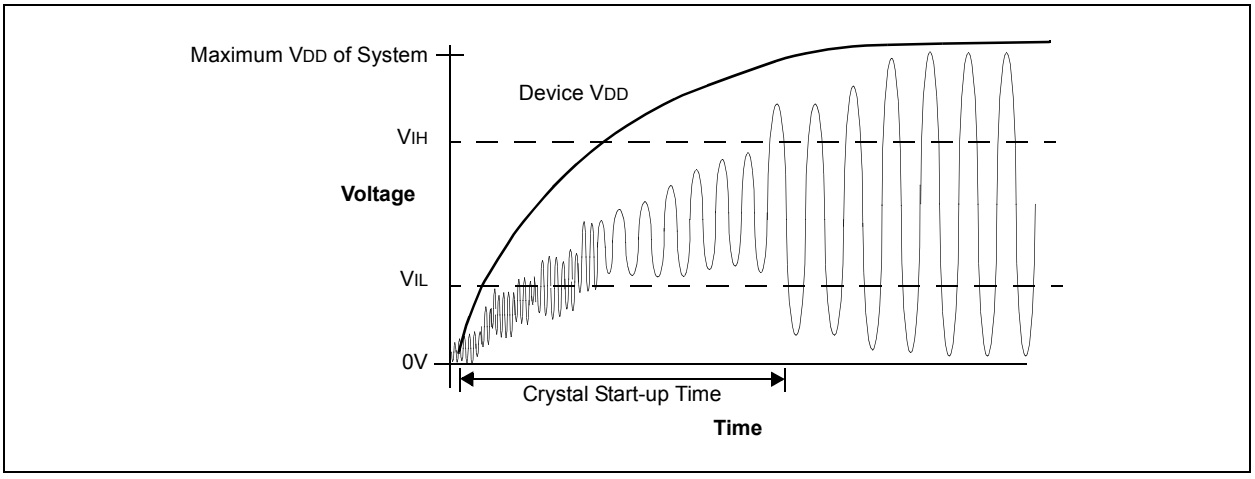
3.6.1 OSCILLATOR/RESONATOR START-UP

As the device voltage increases from V_{SS}, the oscillator will start its oscillations. The time required for the oscillator to start oscillating depends on many factors, including:

- Crystal/resonator frequency
- Capacitor values used
- Series resistor, if used, and its value and type
- Device V_{DD} rise time
- System temperature
- Oscillator mode selection of device (selects the gain of the internal oscillator inverter)
- Crystal quality
- Oscillator circuit layout
- System noise

The course of a typical crystal or resonator start-up is shown in Figure 3-4. Notice that the time to achieve stable oscillation is not instantaneous.

FIGURE 3-4: EXAMPLE OSCILLATOR/RESONATOR START-UP CHARACTERISTICS



6.0 MEMORY ORGANIZATION

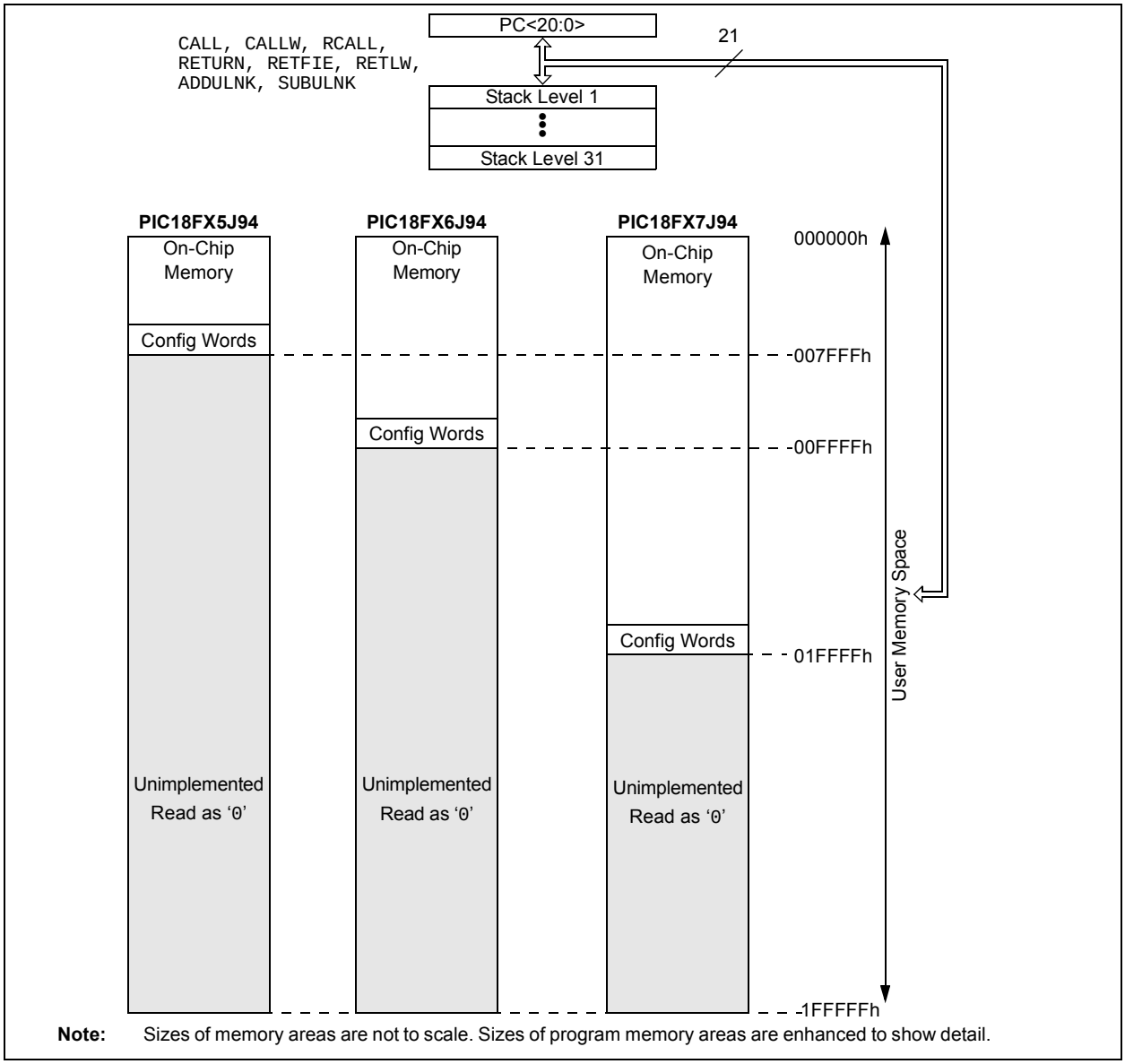
PIC18FXXJ94 devices have these types of memory:

- Program Memory
- Data RAM

As Harvard architecture devices, the data and program memories use separate buses. This enables concurrent access of the two memory spaces.

Additional detailed information on the operation of the Flash program memory is provided in **Section 7.0 "Flash Program Memory"**.

FIGURE 6-1: MEMORY MAPS FOR PIC18F97J94 FAMILY DEVICES



Note: Sizes of memory areas are not to scale. Sizes of program memory areas are enhanced to show detail.

10.6 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are six Peripheral Interrupt Enable registers (PIE1 through PIE6). When IPEN (RCON<7>) = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 10-10: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE
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 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit 5 **RC1IE:** EUSART1 Receive Interrupt Enable bit
1 = Enables the EUSART1 receive interrupt
0 = Disables the EUSART1 receive interrupt

bit 4 **TX1IE:** EUSART1 Transmit Interrupt Enable bit
1 = Enables the EUSART1 transmit interrupt
0 = Disables the EUSART1 transmit interrupt

bit 3 **SSP1IE:** Master Synchronous Serial Port 1 Interrupt Enable bit
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt

bit 2 **TMR1GIE:** TMR1 Gate Interrupt Enable bit
1 = Enables the gate
0 = Disables the gate

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

TABLE 11-4: PORTD FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD3/PSP3/ RP23/SEG3/AD3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	PSP3	x	I/O	ST/DIG	Parallel Slave Port Data Bus Bit 3.
	RP23	x	x	DIG	Reconfigurable Pin 23 for PPS-Lite; TRIS must be set to match input/ output of module.
	SEG3	0	O	ANA	LCD Segment 3 output; disables all other pin functions.
	AD3	x	I/O	ST/DIG	External Memory Bus Address Line 3.
RD4/PSP4/ RP24/SEG4/AD4	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	PSP4	x	I/O	ST/DIG	Parallel Slave Port Data Bus Bit 4.
	RP24	x	x	DIG	Reconfigurable Pin 24 for PPS-Lite; TRIS must be set to match input/ output of module.
	SEG4	0	O	ANA	LCD Segment 4 output; disables all other pin functions.
	AD4	x	I/O	ST/DIG	External Memory Bus Address Line 4.
RD5/PSP5/ RP25/SDA2/ SEG5/AD5	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	PSP5	x	I/O	ST/DIG	Parallel Slave Port Data Bus Bit 5.
	RP25	x	x	DIG	Reconfigurable Pin 25 for PPS-Lite; TRIS must be set to match input/ output of module.
	SDA2	x	I/O	ST/DIG	I^2C mode data I/O.
	SEG5	0	O	ANA	LCD Segment 5 output; disables all other pin functions.
	AD5	x	I/O	ST/DIG	External Memory Bus Address Line 5.
RD6/PSP6/ RP26/SCL2/ SEG6/AD6	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	PSP6	x	I/O	ST/DIG	Parallel Slave Port Data Bus Bit 6.
	RP26	x	x	DIG	Reconfigurable Pin 26 for PPS-Lite; TRIS must be set to match input/ output of module.
	SCL2	x	I/O	I^2C	Synchronous serial clock input/output for I^2C mode.
	SEG6	0	O	ANA	LCD Segment 6 output; disables all other pin functions.
	AD6	x	I/O	ST/DIG	External Memory Bus Address Line 6.
RD7/PSP7/ RP27/REFO2/ SEG7/AD7	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	PSP7	x	I/O	ST/DIG	Parallel Slave Port Data Bus Bit 7.
	RP27	x	x	DIG	Reconfigurable Pin 27 for PPS-Lite; TRIS must be set to match input/ output of module.
	REFO2	0	O	DIG	Reference Clock 2 output.
	SEG7	0	O	ANA	LCD Segment 7 output; disables all other pin functions.
	AD7	x	I/O	ST/DIG	External Memory Bus Address Line 7.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, I^2C = I^2C/SMBus, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 11-6: PORTF FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RF6/RP40/C1INA/ AN11/SEG24	RF6	0	O	DIG	LATF<6> data output.
		1	I	ST	PORTF<6> data input.
	RP40	x	x	DIG	Reconfigurable Pin 40 for PPS-Lite; TRIS must be set to match input/ output of module.
	C1INA	1	I	ANA	Comparator 1 Input A.
	AN11	1	I	ANA	A/D Input Channel 11. Default input configuration on POR; does not affect digital output.
	SEG24	0	O	ANA	LCD Segment 24 output; disables all other pin functions.
RF7/RP38/AN5/ SEG25	RF7	0	O	DIG	LATF<7> data output.
		1	I	ST	PORTF<7> data input.
	RP38	x	x	DIG	Reconfigurable Pin 38 for PPS-Lite; TRIS must be set to match input/ output of module.
	AN5	1	I	ANA	A/D Input Channel 5. Default input configuration on POR; does not affect digital output.
	SEG25	0	O	ANA	LCD Segment 25 output; disables all other pin functions.

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

TABLE 11-13: RPIN REGISTERS AND AVAILABLE FUNCTIONS (CONTINUED)

PPS-Lite Input Peripheral Group 4n + 2	
(1) To Map this Signal	(4) to the Associated RPIN Register
$\overline{SS1}$	RPINR10_11<3:0>
INT2	RPINR26_27<7:4>
IOC2	RPINR20_21<3:0>
IOC6	RPINR24_25<3:0>
MDMIN	RPINR28_29<7:4>
U1TX	RPINR0_1<7:4>
U2RX	RPINR2_3<3:0>
SCK2	RPINR10_11<7:4>
ECCP3	RPINR16_17<7:4>
CCP6	RPINR34_35<3:0>
CCP10	RPINR38_39<3:0>
RVP2	RPINR48_49<3:0>
RVP6	RPINR52_53<3:0>
(2) with this RPn Pin	(3) Write this Corresponding Value
RP2	h'0
RP6	h'1
RP10	h'2
RP14	h'3
RP18	h'4
RP22	h'5
RP26	h'6
RP30	h'7
RP34	h'8
RP38	h'9
RP42	h'A
RP46	h'B
—	h'C
—	h'D
—	h'E
VSS	h'F

PPS-Lite Input Peripheral Group 4n + 3	
(1) To Map this Signal	(4) to the Associated RPIN Register
$\overline{SS2}$	RPINR12_13<7:4>
INT3	RPINR28_29<3:0>
IOC3	RPINR20_21<7:4>
IOC7	RPINR24_25<7:4>
U1RX	RPINR0_1<3:0>
U2TX	RPINR2_3<7:4>
SCK1	RPINR8_9<3:0>
ECCP1	RPINR14_15<7:4>
ECCP2	RPINR16_17<3:0>
CCP4	RPINR32_33<3:0>
RVP3	RPINR48_49<7:4>
RVP7	RPINR52_53<7:4>
(2) with this RPn Pin	(3) Write this Corresponding Value
RP3	h'0
RP7	h'1
RP11	h'2
RP15	h'3
RP19	h'4
RP23	h'5
RP27	h'6
RP31	h'7
RP35	h'8
RP39	h'9
RP43	h'A
—	h'B
—	h'C
—	h'D
—	h'E
VSS	h'F

11.15.3.2 Output Mapping

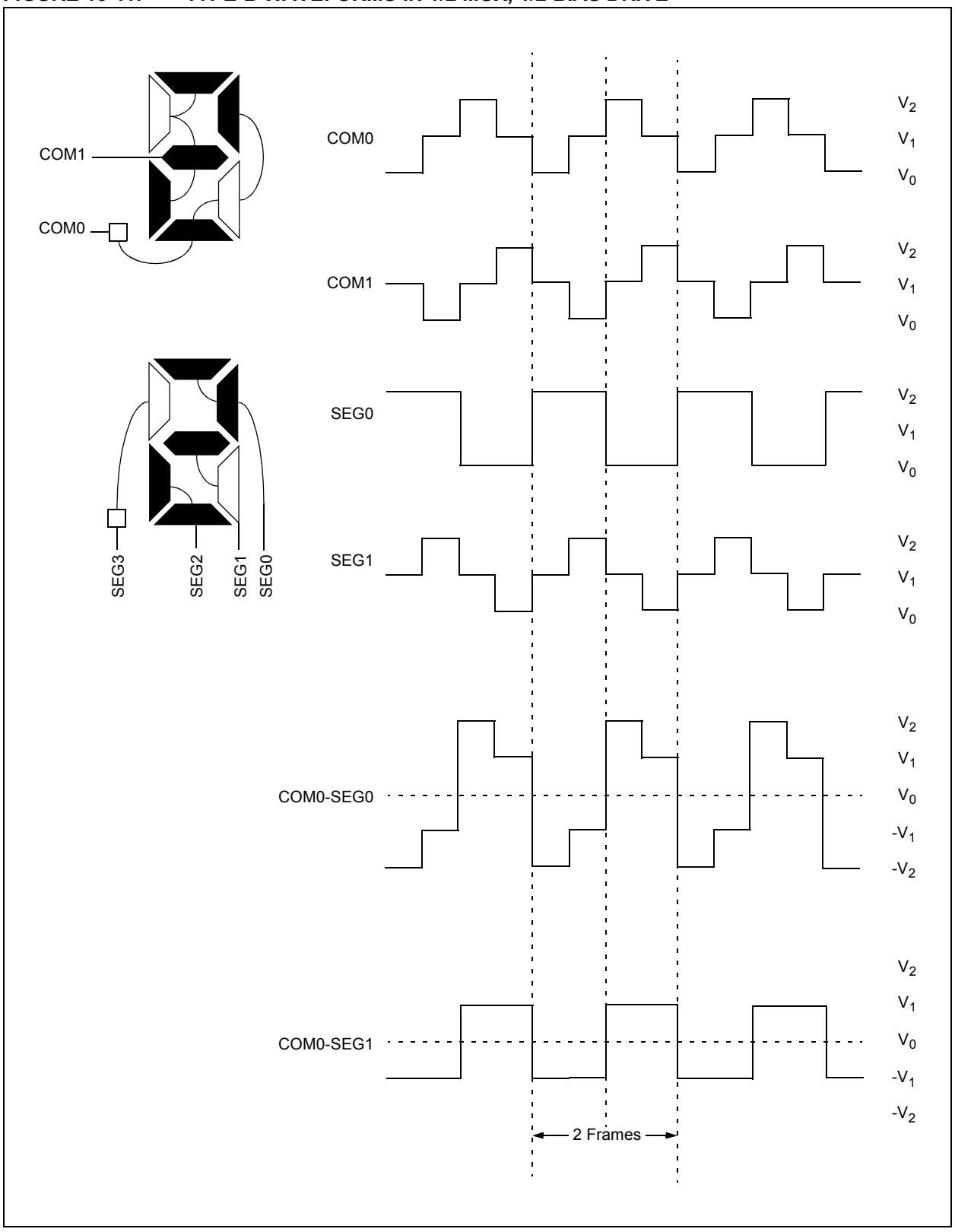
In contrast to the inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a bit field associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers contain sets of 4-bit fields, with each associated with one RPn pin (see Register 11-5). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin. Each pin has a limited set of peripherals to choose from.

The PPS-Lite peripheral outputs and associated RPn pins have been organized into four groups. It is not possible to map a peripheral to an RPn pin which is outside of its group. To map a peripheral output signal to an RPn pin, use the 4-step process, as indicated in Table 11-14. Choose the RPn pin and the signal; the column on the right shows which value to write to the associated RPORx register.

The peripheral outputs that support Peripheral Pin Selection have no default pins. Since the RPORx registers reset to all '0's, the outputs are all disconnected in the device's default (Reset) state.

The list of peripherals for output mapping also includes a null value of b'0000' because of the mapping technique. This allows unused peripherals to not be connected to a pin. Not all peripherals are available on all pins. For example, the "SDO2" signal is only available on RP0, RP4, RP8, etc. The "SDO2" signal is not available on RP1.

FIGURE 13-11: TYPE-B WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE



15.3 Timer1/3/5 16-Bit Read/Write Mode

Timer1/3/5 can be configured for 16-bit reads and writes (see Figure 15-3). When the RD16 control bit (TxCON<1>) is set, the address for TMRxH is mapped to a buffer register for the high byte of Timer1/3/5. A read from TMRxL will load the contents of the high byte of Timer1/3/5 into the Timerx High Byte Buffer register. This provides users with the ability to accurately read all 16 bits of Timer1/3/5 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer1/3/5 must also take place through the TMRxH Buffer register. The Timer1/3/5 high byte is updated with the contents of TMRxH when a write occurs to TMRxL. This allows users to write all 16 bits to both the high and low bytes of Timer1/3/5 at once.

The high byte of Timer1/3/5 is not directly readable or writable in this mode. All reads and writes must take place through the Timerx High Byte Buffer register.

Writes to TMRxH do not clear the Timer1/3/5 prescaler. The prescaler is only cleared on writes to TMRxL.

15.4 Using the SOSC Oscillator as the Timer1/3/5 Clock Source

The SOSC Internal Oscillator may be used as the clock source for Timer1/3/5. It can be enabled in one of these ways:

- Setting the SOSCEN bit in either of the TxCON registers (TxCON<3>)
- Setting the SOSCGO bit in the OSCCON2 register (OSCCON2<1>)
- Setting the NOSC bits to secondary clock source in the OSCCON register (OSCCON<2:0> = 100)

The SOSCGO bit is used to warm up the SOSC so that it is ready before any peripheral requests it.

To use it as the Timer3 clock source, the TMR3CSx bits must also be set. As previously noted, this also configures Timer3 to increment on every rising edge of the oscillator source.

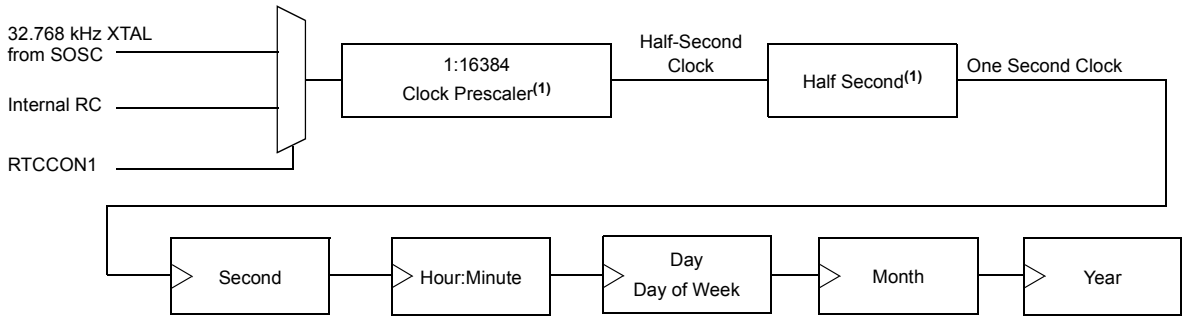
The SOSC Oscillator is described in **Section 15.4 “Using the SOSC Oscillator as the Timer1/3/5 Clock Source”**.

17.2.2 CLOCK SOURCE

As mentioned earlier, the RTCC module is intended to be clocked by an external Real-Time Clock (RTC) crystal, oscillating at 32.768 kHz, but an internal oscillator can be used. The RTCC clock selection is decided by the RTCOSC bit (CONFIG3L<0>).

Calibration of the crystal can be done through this module to yield an error of 3 seconds or less per month. (For further details, see **Section 17.2.9 "Calibration"**.)

FIGURE 17-4: CLOCK SOURCE MULTIPLEXING



Note 1: Writing to the lower half of the MINSEC register resets all counters, allowing fraction of a second synchronization; clock prescaler is held in Reset when RTCEN = 0.

17.2.2.1 Real-Time Clock Enable

The RTCC module can be clocked by an external, 32.768 kHz crystal (SOSC Oscillator) or the LF-INTOSC Oscillator, which can be selected in CONFIG3L<0>.

If the external clock is used, the SOSC Oscillator should be enabled. If LF-INTOSC is providing the clock, the INTOSC clock can be brought out to the RTCC pin by the RTSECSEL<1:0> bits (RTCCON2<1:0>).

17.2.3 DIGIT CARRY RULES

This section explains which timer values are affected when there is a rollover:

- Time of Day: From 23:59:59 to 00:00:00 with a carry to the Day field
- Month: From 12/31 to 01/01 with a carry to the Year field
- Day of Week: From 6 to 0 with no carry (see Table 17-1)
- Year Carry: From 99 to 00; this also surpasses the use of the RTCC

For the day-to-month rollover schedule, see Table 17-2.

Because the following values are in BCD format, the carry to the upper BCD digit occurs at the count of 10, not 16 (SECONDS, MINUTES, HOURS, WEEKDAY, DAYS and MONTHS).

TABLE 17-1: DAY OF WEEK SCHEDULE

Day of Week	
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

TABLE 17-2: DAY TO MONTH ROLLOVER SCHEDULE

Month	Maximum Day Field
01 (January)	31
02 (February)	28 or 29(1)
03 (March)	31
04 (April)	30
05 (May)	31
06 (June)	30
07 (July)	31
08 (August)	31
09 (September)	30
10 (October)	31
11 (November)	30
12 (December)	31

Note 1: See **Section 17.2.4 "Leap Year"**.

REGISTER 19-3: CCPTMRS2: CCP TIMER SELECT REGISTER 2

U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0
bit 7							bit 0

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

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

bit 4 **C10TSEL0:** CCP10 Timer Selection bit
0 = CCP10 is based off of TMR1/TMR2
1 = CCP10 is based off of TMR5/TMR2

bit 3 **Unimplemented:** Read as '0'

bit 2 **C9TSEL0:** CCP9 Timer Selection bit
0 = CCP9 is based off of TMR1/TMR2
1 = CCP9 is based off of TMR5/TMR4

bit 1-0 **C8TSEL<1:0>:** CCP8 Timer Selection bits
00 =CCP8 is based off of TMR1/TMR2
01 =CCP8 is based off of TMR3/TMR4
10 =CCP8 is based off of TMR3/TMR6
11 =Reserved; do not use

19.4 PWM Mode

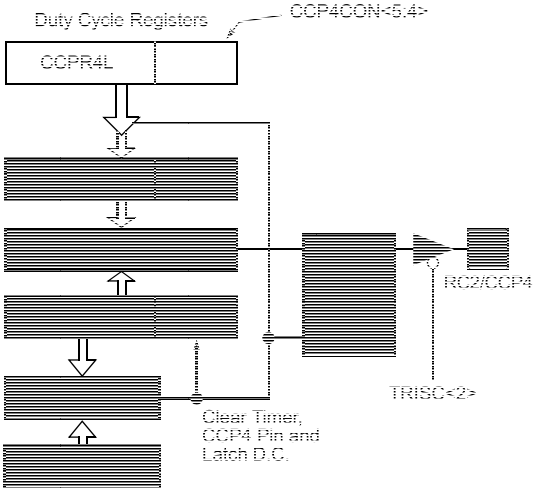
In Pulse-Width Modulation (PWM) mode, the CCP4 pin produces up to a 10-bit resolution PWM output. Since the CCP4 pin is multiplexed with a PORTC or PORTE data latch, the appropriate TRIS bit must be cleared to make the CCP4 pin an output.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch (depending on device configuration) to the default low level. This is not the PORTx I/O data latch.

Figure 19-3 shows a simplified block diagram of the CCP4 module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 19.4.3 "Setup for PWM Operation"**.

FIGURE 19-3: SIMPLIFIED PWM BLOCK DIAGRAM

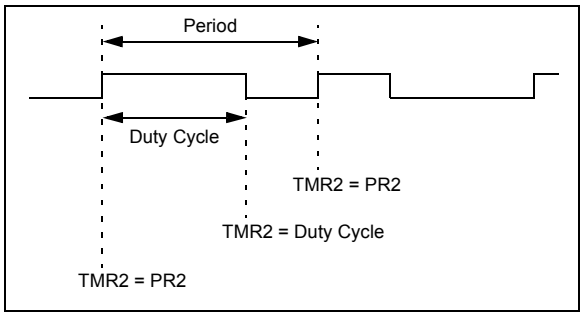


Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

2: CCP4 and its appropriate timers are used as an example. For details on all of the CCP modules and their timer assignments, see Table 19-2 and Table 19-3.

A PWM output (Figure 19-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 19-4: PWM OUTPUT



19.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 19-1: PWM PERIOD CALCULATION

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP4 pin is set

  (An exception: If PWM Duty Cycle = 0%, the CCP4 pin will not be set)
- The PWM duty cycle is latched from CCPR4L into CCPR4H

Note:	The Timer2 postscalers (see **Section 16.0 "Timer2/4/6/8 Modules"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

EXAMPLE 22-3: A/D INITIALIZATION CODE EXAMPLE

```
ADCON1H = 0x22;         // Configure sample clock source
ADCON1L = 0x00;         // and conversion trigger mode.
                        // Unsigned Fraction format (FORM<1:0>=10),
                        // Manual conversion trigger (SSRC<3:0>=0000),
                        // Manual start of sampling (ASAM=0),
                        // S/H in Sample (SAMP = 1)
ADCON2H = 0;            // Configure A/D voltage reference
ADCON2L = 0;            // and buffer fill modes.
                        // Vr+ and Vr- from AVdd and AVss(PVCFG<1:0>=00, NVCFG=0),
                        // Inputs are not scanned,
                        // Interrupt after every sample
ADCON3H = 0;            // Configure sample time = 1Tad,
ADCON3L = 0;            // A/D conversion clock as Tcy

ADCHS0H = 0;            // Configure input channels,
ADCHS0L = 0;            // S/H+ input is AN0,
                        // S/H- input is Vr- (AVss).
ADCSS0L = 0;            // No inputs are scanned.
ADCSS0H = 0;            // No inputs are scanned.
PIR1bits.ADIF = 0;      // Clear A/D conversion interrupt.

// Configure A/D interrupt priority bits (ADIP) here, if
// required. Default priority level is high.

PIE1bits.ADIE = 1;      // Enable A/D conversion interrupt
ADCON1Hbits.ADON = 1;   // Turn on A/D
ADCON1Lbits.SAMP = 1;   // Start sampling the input
Delay();                // Ensure the correct sampling time has elapsed
                        // before starting conversion.
ADCON1Lbits.SAMP = 0;   // End A/D sampling and start conversion

// Example code for A/D ISR:
#pragma interrupt _ADC1Interrupt
void _ADC1Interrupt(void)
{
    PIR1bits.ADIF = 0;
}
```

REGISTER 23-2: CMSTAT: COMPARATOR STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	R-x	R-x	R-x
—	—	—	—	—	C3OUT	C2OUT	C1OUT
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **C3OUT:C1OUT:** Comparator x Status bits

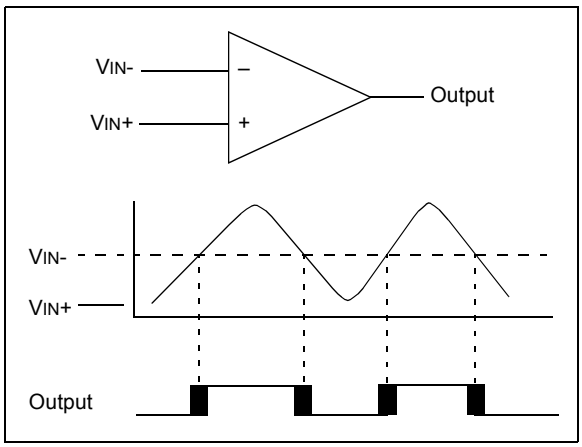
If CPOL (CMxCON<5>)= 0 (non-inverted polarity):
1 = Comparator x's VIN+ > VIN-
0 = Comparator x's VIN+ < VIN-

CPOL = 1 (inverted polarity):
1 = Comparator x's VIN+ < VIN-
0 = Comparator x's VIN+ > VIN-

23.2 Comparator Operation

A single comparator is shown in Figure 23-2, along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input, VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input, VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 23-2 represent the uncertainty due to input offsets and response time.

FIGURE 23-2: SINGLE COMPARATOR



23.3 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response to a comparator input change. Otherwise, the maximum delay of the comparators should be used (see **Section 30.0 "Electrical Specifications"**).

23.4 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 23-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

REGISTER 27-2: UCFG: USB CONFIGURATION REGISTER

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	UOEMON	—	UPUEN(1,2)	UTRDIS(1,3)	FSEN(1)	PPB1	PPB0
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 **UTEYE:** USB Eye Pattern Test Enable bit
1 = Eye pattern test is enabled
0 = Eye pattern test is disabled

bit 6 **UOEMON:** USB OE Monitor Enable bit
1 = $\overline{\text{UOE}}$ signal is active, indicating intervals during which the D+/D- lines are driving
0 = $\overline{\text{UOE}}$ signal is inactive

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

bit 4 **UPUEN:** USB On-Chip Pull-up Enable bit(1,2)
1 = On-chip pull-up is enabled (pull-up on D+ with FSEN = 1 or D- with FSEN = 0)
0 = On-chip pull-up is disabled

bit 3 **UTRDIS:** On-Chip Transceiver Disable bit(1,3)
1 = On-chip transceiver is disabled
0 = On-chip transceiver is active

bit 2 **FSEN:** Full-Speed Enable bit(1)
1 = Full-speed device: Controls transceiver edge rates; requires input clock at 48 MHz
0 = Low-speed device: Controls transceiver edge rates; requires input clock at 6 MHz

bit 1-0 **PPB<1:0>:** Ping-Pong Buffers Configuration bits
11 = Even/Odd ping-pong buffers are enabled for Endpoints 1 to 15
10 = Even/Odd ping-pong buffers are enabled for all endpoints
01 = Even/Odd ping-pong buffer are enabled for OUT Endpoint 0
00 = Even/Odd ping-pong buffers are disabled

Note 1: The UPUEN, UTRDIS and FSEN bits should never be changed while the USB module is enabled. These values must be preconfigured prior to enabling the module.
2: This bit is only valid when the on-chip transceiver is active (UTRDIS = 0); otherwise, it is ignored.
3: If UTRDIS is set, the $\overline{\text{UOE}}$ signal will be active, independent of the UOEMON bit setting.

27.6 USB Power Modes

Many USB applications will likely have several different sets of power requirements and configuration. The most common power modes encountered are Bus Power Only, Self-Power Only and Dual Power with Self-Power Dominance. The most common cases are presented here. Also provided is a means of estimating the current consumption of the USB transceiver.

27.6.1 BUS POWER ONLY

In Bus Power Only mode, all power for the application is drawn from the USB (Figure 27-9). This is effectively the simplest power method for the device.

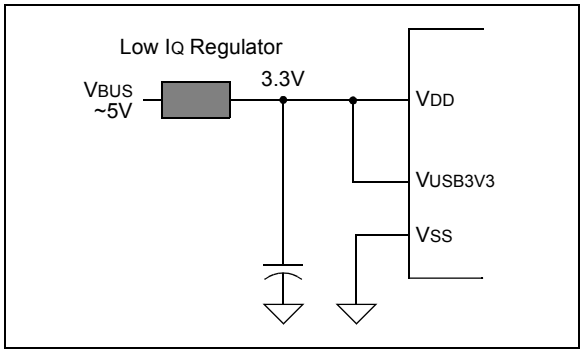
In order to meet the inrush current requirements of the "*USB 2.0 Specification*", the total effective capacitance appearing across VBUS and ground must be no more than 10 µF. If not, some kind of inrush timing is required. For more details, see **Section 7.2.4 of the "*USB 2.0 Specification*"**.

According to the "*USB 2.0 Specification*", all USB devices must also support a Low-Power Suspend mode. In the USB Suspend mode, devices must consume no more than 2.5 mA from the 5V VBUS line of the USB cable.

The host signals the USB device to enter the Suspend mode by stopping all USB traffic to that device for more than 3 ms. This condition will cause the IDLEIF bit in the UIR register to become set.

During the USB Suspend mode, the D+ or D- pull-up resistor must remain active, which will consume some of the allowed suspend current: 2.5 mA budget.

FIGURE 27-9: BUS POWER ONLY



27.6.2 SELF-POWER ONLY

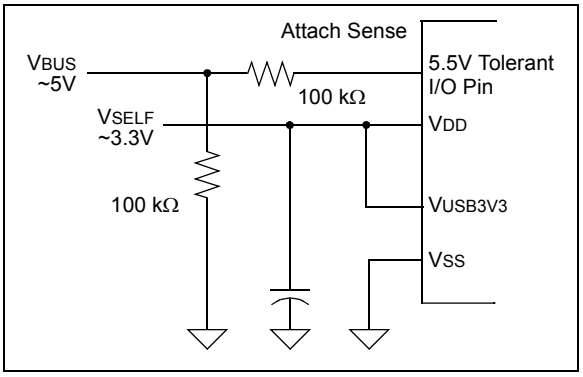
In Self-Power Only mode, the USB application provides its own power, with very little power being pulled from the USB. See Figure 27-10 for an example.

Note that an attach indication is added to indicate when the USB has been connected and the host is actively powering VBUS.

In order to meet compliance specifications, the USB module (and the D+ or D- pull-up resistor) should not be enabled until the host actively drives VBUS high. One of the 5.5V tolerant I/O pins may be used for this purpose.

The application should never source any current onto the 5V VBUS pin of the USB cable.

FIGURE 27-10: SELF-POWER ONLY

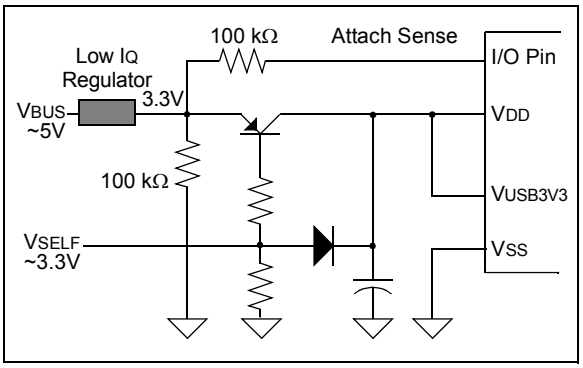


27.6.3 DUAL POWER WITH SELF-POWER DOMINANCE

Some applications may require a dual power option. This allows the application to use internal power primarily, but switch to power from the USB when no internal power is available. See Figure 27-11 for a simple Dual Power with Self-Power Dominance mode example, which automatically switches between Self-Power Only and USB Bus Power Only modes.

Dual power devices must also meet all of the special requirements for inrush current and Suspend mode current, and must not enable the USB module until VBUS is driven high. See **Section 27.6.1 "Bus Power Only"** and **Section 27.6.2 "Self-Power Only"** for descriptions of those requirements. Additionally, dual power devices must never source current onto the 5V VBUS pin of the USB cable.

FIGURE 27-11: DUAL POWER WITH SELF-POWER DOMINANCE



Note:	Users should keep in mind the limits for devices drawing power from the USB. According to USB Specification 2.0, this cannot exceed 100 mA per low-power device or 500 mA per high-power device.

ADDWFC **ADD W and Carry bit to f**

Syntax: ADDWFC f {,d {,a}}

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

Operation: (W) + (f) + (C) → dest

Status Affected: N,OV, C, DC, Z

Encoding:

0010	00da	ffff	ffff

Description: Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ADDWFC REG, 0, 1`

Before Instruction
Carry bit = 1
REG = 02h
W = 4Dh
After Instruction
Carry bit = 0
REG = 02h
W = 50h

ANDLW **AND Literal with W**

Syntax: ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. k → W

Status Affected: N, Z

Encoding:

0000	1011	kkkk	kkkk

Description: The contents of W are ANDed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: `ANDLW 05Fh`

Before Instruction
W = A3h
After Instruction
W = 03h

TBLRD **Table Read**

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

Operands: None

Operation: if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT

Status Affected: None

Encoding:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR<0> = 0: Least Significant Byte of Program Memory Word

TBLPTR<0> = 1: Most Significant Byte of Program Memory Word

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

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

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD **Table Read (Continued)**

Example 1:
```
TBLRD   *+ ;
```
Before Instruction
TABLAT = 55h
TBLPTR = 00A356h
MEMORY(00A356h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 00A357h

Example 2:
```
TBLRD   +* ;
```
Before Instruction
TABLAT = AAh
TBLPTR = 01A357h
MEMORY(01A357h) = 12h
MEMORY(01A358h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 01A358h

31.2 MPLAB XC Compilers

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

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

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

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

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

31.3 MPASM Assembler

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

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

The MPASM Assembler features include:

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

31.4 MPLINK Object Linker/ MPLIB Object Librarian

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

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

The object linker/library features include:

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

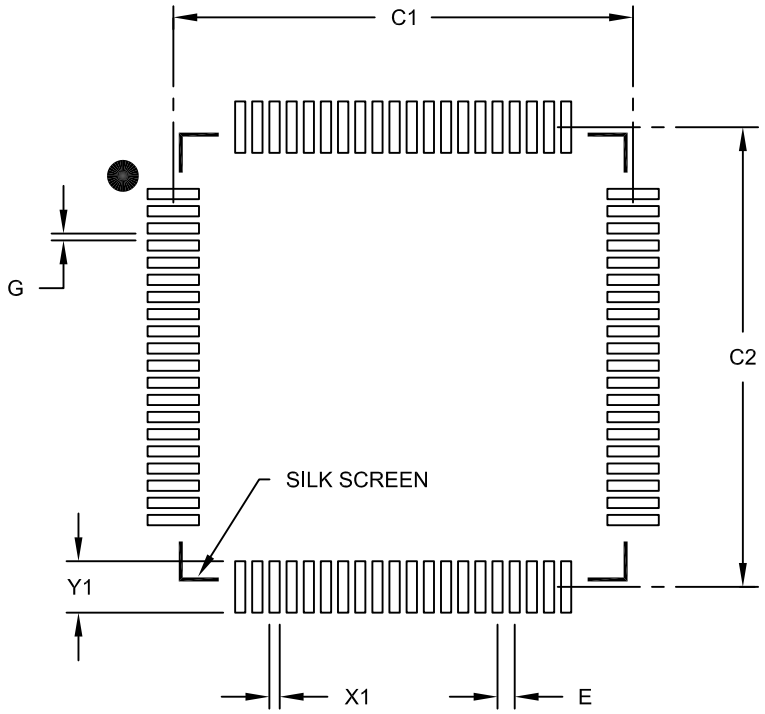
31.5 MPLAB Assembler, Linker and Librarian for Various Device Families

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

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

80-Lead Plastic Thin Quad Flatpack (PT) - 12x12x1mm Body, 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X80)	X1			0.30
Contact Pad Length (X80)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2092B