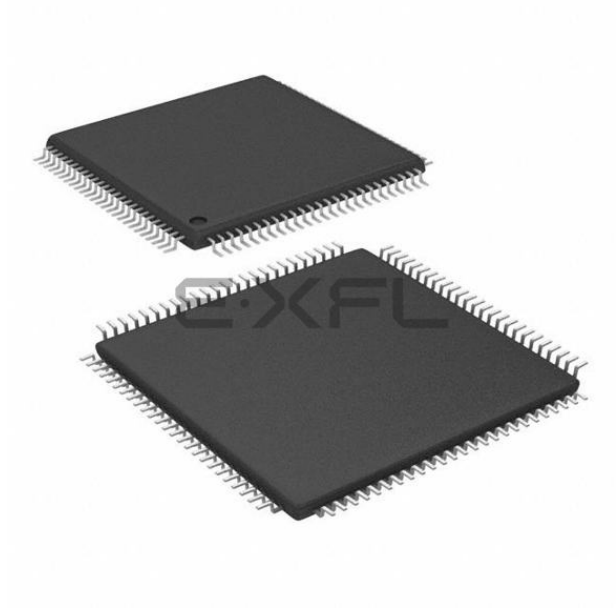


Welcome to E-XFL.COM

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	128KB (64K 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 (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f97j94-i-pf

3.15.3 CLOCK SYNCHRONIZATION

The Reference Clock Output is enabled only once (ON = 1). Note that the source of the clock and the divider values should be chosen prior to the bit being set to avoid glitches on the REFO output.

Once the ON bit is set, its value is synchronized to the reference clock domain to enable the output. This ensures that no glitches will be seen on the output. Similarly, when the ON bit is cleared, the output and the associated output enable signals will be synchronized, and disabled on the falling edge of the reference clock. Note that with large divider values, this will cause the REFO to be enabled for some period after ON is cleared.

3.15.4 OPERATION IN SLEEP MODE

If any clock source, other than the peripheral clock, is used as a base reference (i.e., ROSEL<3:0> ≠ 0001), the user has the option to configure the behavior of the oscillator in Sleep mode. The RSLP Configuration bit determines if the oscillator will continue to run in Sleep. If RSLP = 0, the oscillator will be shut down in Sleep (assuming no other consumers are requesting it). If RSLP = 1, the oscillator will continue to run in Sleep.

The Reference Clock Output is synchronized with the Sleep signal to avoid any glitches on its output.

3.15.5 MODULE ENABLE SIGNAL

The REFOx module may be enabled or disabled using the REFOxMD register bit (PMD3, bit 1 or 0). The module also needs to be turned on using the ON bit (REFO1CON<7>).

3.15.5.1 Registers and Bits

This module provides the following device registers and/or bits:

- REFOxCON – Reference Clock Output Control Register
- REFOxCON1 – Reference Clock Output Control 1 Register
- REFOxCON2 – Reference Clock Output Control 2 Register
- REFOxCON3 – Reference Clock Output Control 3 Register

5.2 Power-on Reset (POR)

The PIC18F97J94 family has two types of Power-on Resets:

- POR
- VBAT POR

POR is the legacy PIC18J series Power-on Reset which monitors core power supply. The second, VBAT POR, monitors voltage on the VBAT pin. These POR circuits use the same technique to enable and monitor their respective power source for adequate voltage levels to ensure proper chip operation. There are two threshold voltages associated with them. The first voltage is the device threshold voltage, VPOR. The device threshold voltage is the voltage at which the POR module becomes operable. The second voltage associated with a POR event is the POR circuit threshold voltage. Once the correct threshold voltage is detected, a power-on event occurs and the POR module hibernates to minimize current consumption.

A power-on event generates an internal POR pulse when a VDD rise is detected. The device supply voltage characteristics must meet the specified starting voltage, VPOR, and rise rate requirements, SVDD, to generate the POR pulse. In particular, VDD must fall below VPOR before a new POR is initiated. For more information on the VPOR and VDD rise rate specifications, refer to **Section 30.0 "Electrical Specifications"**.

5.2.1 POR CIRCUIT

The POR circuit behaves differently than VBAT POR once the POR state becomes active. The internal POR pulse resets the POR timer and places the device in the Reset state. The POR also selects the device clock source identified by the Oscillator Configuration bits. After the POR pulse is generated, the POR circuit inserts a small delay, TCSD, to ensure that internal device bias circuits are stable.

After the expiration of TCSD, a delay, TPWRT, is always inserted every time the device resumes operation after any power-down. During this time, code execution is disabled. The PWRT is used to extend the duration of a power-up sequence to permit the on-chip band gap and regulator to stabilize and to load the Configuration Word settings. The on-chip regulator is always enabled and its stabilization time is shorter than other concurrently running delays, and does not extend start-up time.

The power-on event clears the $\overline{\text{BOR}}$ and $\overline{\text{POR}}$ Status bits (RCON<1:0>); it does not change for any other Reset event. $\overline{\text{POR}}$ is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any Power-on Reset. Alternatively, the VDDPOR (RCON3<2>) bit can be used; it is set on a VDD POR event. It must be cleared after any Power-on Reset to detect subsequent VDD POR events.

After TPWRT expires, an additional start-up time for the system clock (either TOST, TIOBST and TRC, depending on the source) occurs while the clock source becomes stable. Internal Reset is then released and the device is no longer held in Reset (Table 5-2). Once all of the delays have expired, the system clock is released and code execution can begin. Refer to **Section 30.0 "Electrical Specifications"** for more information on the values of the delay parameters.

Note:	When the device exits the Reset condition (begins normal operation), the device operating parameters (voltage, frequency, temperature, etc.) must be within their operating ranges; otherwise, the device will not function correctly. The user must ensure that the delay between the time power is first applied, and the time, INTERNAL RESET, becomes inactive, is long enough to get all operating parameters within specification.

TABLE 5-3: 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
POSTINC2	64-pin	80-pin	100-pin	N/A	N/A	N/A
POSTDEC2	64-pin	80-pin	100-pin	N/A	N/A	N/A
PREINC2	64-pin	80-pin	100-pin	N/A	N/A	N/A
PLUSW2	64-pin	80-pin	100-pin	N/A	N/A	N/A
FSR2H	64-pin	80-pin	100-pin	---- xxxx	---- uuuu	---- uuuu
FSR2L	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	64-pin	80-pin	100-pin	---x xxxx	---u uuuu	---u uuuu
TMR0H	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
TMR0L	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
RESERVED	64-pin	80-pin	100-pin	---- ----	---- ----	---- ----
OSCCON	64-pin	80-pin	100-pin	0qqq -qqq	uuuu -uuu	uuuu -uuu
IPR5	64-pin	80-pin	100-pin	-111 -111	-uuu -uuu	-uuu -uuu
IOCF	64-pin	80-pin	100-pin	0000 0000	0000 0000	qqqq qqqq
RCON(4)	64-pin	80-pin	100-pin	0-11 11qq	0-qq qquu	u-qq qquu
TMR1H	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
TMR2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PR2	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
T2CON	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP1STAT	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CMSTAT	64-pin	80-pin	100-pin	---- -xxx	---- -uuu	---- -uuu
ADCBUF0H	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCBUF0L	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1H	64-pin	80-pin	100-pin	0--- -000	u--- -uuu	u--- -uuu
ADCON1L	64-pin	80-pin	100-pin	0000 -000	uuuu -uuu	uuuu -uuu
CVRCONH	64-pin	80-pin	100-pin	---0 0000	---u uuuu	---u uuuu
CVRCONL	64-pin	80-pin	100-pin	0000 ---0	uuuu ---u	uuuu ---u

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

Note 1: 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.

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

4: See Table 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

TABLE 5-3: 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
SSP2CON3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CM2CON	64-pin	80-pin	100-pin	0001 1111	0001 1111	uuuu uuuu
CM3CON	64-pin	80-pin	100-pin	0001 1111	0001 1111	uuuu uuuu
CCPTMRS0	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CCPTMRS1	64-pin	80-pin	100-pin	00-0 -000	00-0 -000	uuuu uuuu
CCPTMRS2	64-pin	80-pin	100-pin	---0 -000	---0 -000	uuuu uuuu
RCSTA2	64-pin	80-pin	100-pin	0000 000x	0000 000x	uuuu uuuu
TXSTA2	64-pin	80-pin	100-pin	0000 0010	0000 0010	uuuu uuuu
BAUDCON2	64-pin	80-pin	100-pin	01x0 0000	01x0 0000	uuuu uuuu
SPBRGH1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCSTA3	64-pin	80-pin	100-pin	0000 000x	0000 000x	uuuu uuuu
TXSTA3	64-pin	80-pin	100-pin	0000 0010	0000 0010	uuuu uuuu
BAUDCON3	64-pin	80-pin	100-pin	01x0 0000	01x0 0000	uuuu uuuu
SPBRGH3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SPBRG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCREG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TXREG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
DSCONL	64-pin	80-pin	100-pin	---- -000	---- -000	--- -uuu
DSCONH	64-pin	80-pin	100-pin	0-0- ---0	u-u- ---u	u-u- ---u
DSWAKEL	64-pin	80-pin	100-pin	0000 0001	uuuu uuuu	uuuu uuuu
DSWAKEH	64-pin	80-pin	100-pin	---- ---0	---- ---u	---- ---q
DSGPR0(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR1(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR2(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR3	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRGH2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SPBRG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCREG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TXREG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PSTR2CON	64-pin	80-pin	100-pin	00-0 0001	00-0 0001	uu-u uuuu
PSTR3CON	64-pin	80-pin	100-pin	00-0 0001	00-0 0001	uu-u uuuu
SSP2STAT	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu

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

Note 1: 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.

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

4: See Table 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

TABLE 5-3: 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
CTMUCON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CTMUCON2	64-pin	80-pin	100-pin	0000 00--	0000 00--	uuuu uu--
CTMUCON3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PMD0	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PMD1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PMD2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PMD3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PMD4	64-pin	80-pin	100-pin	0000 00--	0000 00--	uuuu uu--
MDCON	64-pin	80-pin	100-pin	0010 0--0	0010 0--0	uuuu u--u
MDSRC	64-pin	80-pin	100-pin	0--- xxxx	0--- uuuu	u--- uuuu
MDCARH	64-pin	80-pin	100-pin	0xx- xxxx	0uu- uuuu	uuu- uuuu
MDCARL	64-pin	80-pin	100-pin	0xx- xxxx	0uu- uuuu	uuu- uuuu
ODCON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
ODCON2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TRISK	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
LATK	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTK	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
TRISL	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
LATL	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTL	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
MEMCON	64-pin	80-pin	100-pin	0-00 --00	0-00 --00	u-uu --uu
REFO1CON	64-pin	80-pin	100-pin	0-00 0-00	u-uu u-uu	u-uu u-uu
REFO1CON1	64-pin	80-pin	100-pin	---- 0000	---- uuuu	---- uuuu
REFO1CON2	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
REFO1CON3	64-pin	80-pin	100-pin	-000 0000	-uuu uuuu	-uuu uuuu
REFO2CON	64-pin	80-pin	100-pin	0-00 0-00	u-uu u-uu	u-uu u-uu
REFO2CON1	64-pin	80-pin	100-pin	---- 0000	---- uuuu	---- uuuu
REFO2CON2	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
REFO2CON3	64-pin	80-pin	100-pin	-000 0000	-uuu uuuu	-uuu uuuu
LCDPS	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDREG	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDCON	64-pin	80-pin	100-pin	0000 0000	0000 0000	u-uu uuuu
LCDREF	64-pin	80-pin	100-pin	0-00 0000	u-uu uuuu	u-uu uuuu

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

Note 1: 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.

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

4: See Table 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

TABLE 5-3: 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
LCDREFL	64-pin	80-pin	100-pin	0000 -000	uuuu -uuu	uuuu -uuu
LCDSE7	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE6	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE5	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE4	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE3	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE2	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE1	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDSE0	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA63	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA62	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA61	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA60	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA59	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA58	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA57	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA56	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA55	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA54	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA53	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA52	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA51	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA50	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA49	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA48	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA47	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA46	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA45	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA44	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA43	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA42	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA41	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
LCDDATA40	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu

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

Note 1: 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.

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

4: See Table 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

8.3 Wait States

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

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

8.4 Port Pin Weak Pull-ups

With the exception of the upper address lines, A<19:16>, the pins associated with the External Memory Bus are equipped with weak pull-ups. The pull-ups are controlled by the upper nibble of the PADCFG register (PADCFG<7:4>). They are named RDPU, REPU, RHPU and RJPU, and control pull-ups on PORTD, PORTE, PORTH and PORTJ, respectively. Setting one of these bits enables the corresponding pull-ups for that port. All pull-ups are disabled by default on all device Resets.

In Extended Microcontroller mode, the port pull-ups can be useful in preserving the memory state on the external bus while the bus is temporarily disabled (EBDIS = 1).

8.5 Program Memory Modes and the External Memory Bus

The PIC18FXXJ94 of devices is capable of operating in one of two program memory modes, using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the program memory mode selected, as well as the setting of the EBDIS bit.

In **Microcontroller Mode,** the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted. The Reset value of EBDIS ('0') is ignored and the ABWx pins behave as I/O ports.

In **Extended Microcontroller Mode,** the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function.

If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port functions. When EBDIS = 0, the pins function as the external bus. When EBDIS = 1, the pins function as I/O ports.

If the device fetches or accesses external memory while EBDIS = 1, the pins will switch to the external bus. If the EBDIS bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

If the device is executing out of internal memory when EBDIS = 0, the memory bus address/data and control pins will not be active. They will go to a state where the active address/data pins are tri-state; the $\overline{CE}$, $\overline{OE}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{UB}$ and $\overline{LB}$ signals are '1', and ALE and BA0 are '0'. Note that only those pins associated with the current address width are forced to tri-state; the other pins continue to function as I/O. In the case of 16-bit address width, for example, only AD<15:0> (PORTD and PORTE) are affected; A<19:16> (PORTH<3:0>) continue to function as I/O.

In all external memory modes, the bus takes priority over any other peripherals that may share pins with it. This includes the Parallel Master Port and serial communication modules which would otherwise take priority over the I/O port.

8.6 16-Bit Data Width Modes

In 16-Bit Data Width mode, the external memory interface can be connected to external memories in three different configurations:

- 16-Bit Byte Write
- 16-Bit Word Write
- 16-Bit Byte Select

The configuration to be used is determined by the WM<1:0> bits in the MEMCON register (MEMCON<1:0>). These three different configurations allow the designer maximum flexibility in using both 8-bit and 16-bit devices with 16-bit data.

For all 16-bit modes, the Address Latch Enable (ALE) pin indicates that the address bits, AD<15:0>, are available on the external memory interface bus. Following the address latch, the Output Enable ($\overline{OE}$) signal will enable both bytes of program memory at once to form a 16-bit instruction word. The Chip Enable ($\overline{CE}$ signal) is active at any time that the microcontroller accesses external memory, whether reading or writing; it is inactive (asserted high) whenever the device is in Sleep mode.

In Byte Select mode, JEDEC® standard Flash memories will require BA0 for the byte address line and one I/O line to select between Byte and Word mode. The other 16-bit modes do not need BA0. JEDEC standard static RAM memories will use the $\overline{UB}$ or $\overline{LB}$ signals for byte selection.

11.15.4.1 Control Register Lock

The contents of RPINRx and RPORx registers are constantly monitored in hardware by shadow registers. If an unexpected change in any of the registers occurs (such as cell disturbances caused by ESD or other external events), a Configuration Mismatch Reset will trigger.

11.15.4.2 Configuration Bit Pin Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY Configuration bit (CONFIG5H<0>) blocks the IOLOCK bit from being cleared after it has been set once.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access to the Peripheral Pin Select registers. It is good programming practice to always set the IOLOCK bit (OSCCON2<6>) after all changes have been made to PPS-Lite registers.

11.15.5 CONSIDERATIONS FOR PERIPHERAL PIN SELECTION

The ability to control Peripheral Pin Selection introduces several considerations into application design that should be considered. This is particularly true for several common peripherals which are only available as remappable peripherals.

Before any other application code is executed, the user must initialize the device with the proper peripheral configuration. Since the IOLOCK is not active in the Reset state, the peripherals can be configured, and the IOLOCK bit can be set when configuration is complete.

Choosing the configuration requires the review of all Peripheral Pin Selects and their pin assignments, especially those that will not be used in the application. In all cases, unused pin-selected peripherals should be disabled. Unused peripherals should have their inputs assigned to Vss. I/O pins with unused RPn functions should be configured with the NULL (‘0’) peripheral output.

The assignment of an RPn pin to the peripheral input or output depends on the peripheral and its use in the application. It is good programming practice to map peripherals to pins immediately after Reset. This should be done before any configuration changes to the peripheral itself.

The assignment of a peripheral output to a particular pin does not automatically perform any other configuration of the pin’s I/O circuitry. This means adding a pin-selectable output to a pin may mean inadvertently driving an existing peripheral input when the output is driven. Users must be familiar with the behavior of other fixed peripherals that share a remappable pin. To be safe, fixed digital peripherals that share the same pin should be disabled when not in use.

Configuring a remappable pin for a specific peripheral input does not automatically turn that feature on. The peripheral must be specifically configured for operation and enabled, as if it were tied to a fixed pin.

A final consideration is that Peripheral Pin Select functions neither override analog inputs, nor reconfigure pins with analog functions for digital I/O. If a pin is configured as an analog input on device Reset, it must be explicitly reconfigured as digital I/O when used with a Peripheral Pin Select.

11.15.5.1 Basic Steps to Use Peripheral Pin Selection Lite (PPS-Lite)

1. Disable any fixed digital peripherals on the pins to be used.
2. Switch pins to be used for digital functionality (if they have analog functionality) using the ANCONx registers.
3. Clear the IOLOCK bit (OSCCON<6>) if needed (not needed after a device Reset).
4. Set RPINRx and RPORx registers appropriately.
5. Set the IOLOCK bit (OSCCON<6>).
6. Enable and configure newly mapped PPS-Lite peripherals.

REGISTER 13-5: LCDDATAx: LCD DATA x REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
S(n)Cy	S(n)Cy	S(n)Cy	S(n)Cy	S(n)Cy	S(n)Cy	S(n)Cy	S(n)Cy
bit 7							bit 0

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

bit 7-0 **S(n)Cy:** Pixel On bits

For registers LCDDATA0 through LCDDATA7: n = (0-63), y = 0
For registers LCDDATA8 through LCDDATA15: n = (0-63), y = 1
For registers LCDDATA16 through LCDDATA23: n = (0-63), y = 2
For registers LCDDATA24 through LCDDATA31: n = (0-63), y = 3
For registers LCDDATA32 through LCDDATA39: n = (0-63), y = 4
For registers LCDDATA40 through LCDDATA47: n = (0-63), y = 5
For registers LCDDATA48 through LCDDATA55: n = (0-63), y = 6
For registers LCDDATA56 through LCDDATA63: n = (0-63), y = 7

1 = Pixel on
0 = Pixel off

TABLE 13-2: LCDDATA REGISTERS AND BITS FOR SEGMENT AND COM COMBINATIONS

COM Lines	Segments							
	0 to 7	8 to 15	16 to 23	24 to 31	32 to 39	40 to 47	48 to 55	56 to 63
0	LCDDATA0 S00C0:S07C0	LCDDATA1 S08C0:S15C0	LCDDATA2 S16C0:S23C0	LCDDATA3 S24C0:S31C0	LCDDATA4 S32C0:S39C0	LCDDATA5 S40C0:S47C0	LCDDATA6 S48C0:S55C0	LCDDATA7 S56C0:S63C0
1	LCDDATA8 S00C1:S07C1	LCDDATA9 S08C1:S15C1	LCDDATA10 S16C1:S23C1	LCDDATA11 S24C1:S31C1	LCDDATA12 S32C1:S39C1	LCDDATA13 S40C1:S47C1	LCDDATA14 S48C1:S55C1	LCDDATA15 S56C1:S63C1
2	LCDDATA16 S00C2:S07C2	LCDDATA17 S08C2:S15C2	LCDDATA18 S16C2:S23C2	LCDDATA19 S24C2:S31C2	LCDDATA20 S32C2:S39C2	LCDDATA21 40C2:S47C2	LCDDATA22 S48C2:S55C2	LCDDATA23 S56C2:S63C2
3	LCDDATA24 S00C3:S07C3	LCDDATA25 S08C3:S15C3	LCDDATA26 S16C3:S23C3	LCDDATA27 S24C3:S31C3	LCDDATA28 S32C3:S39C3	LCDDATA29 S40C3:S47C3	LCDDATA30 S48C3:S55C3	LCDDATA31 S56C3:S63C3
4	LCDDATA32 S00C4:S07C4	LCDDATA33 S08C4:S15C4	LCDDATA34 S16C4:S23C4	LCDDATA35 S24C4:S31C4	LCDDATA36 S32C4:S39C4	LCDDATA37 S40C4:S47C4	LCDDATA38 S48C4:S55C4	LCDDATA39 S56C4:S63C4
5	LCDDATA40 S00C5:S07C5	LCDDATA41 S08C5:S15C5	LCDDATA42 S16C5:S23C5	LCDDATA43 S24C5:S31C5	LCDDATA44 S32C5:S39C5	LCDDATA45 S40C5:S47C5	LCDDATA46 S48C5:S55C5	LCDDATA47 S56C5:S63C5
6	LCDDATA48 S00C6:S07C6	LCDDATA49 S08C6:S15C6	LCDDATA50 S16C6:S23C6	LCDDATA51 S24C6:S31C6	LCDDATA52 S32C6:S39C6	LCDDATA53 S40C6:S47C6	LCDDATA54 S48C6:S55C6	LCDDATA55 S56C6:S63C6
7	LCDDATA56 S00C7:S07C7	LCDDATA57 S08C7:S15C7	LCDDATA58 S16C7:S23C7	LCDDATA59 S24C7:S31C7	LCDDATA60 S32C7:S39C7	LCDDATA61 S40C7:S47C7	LCDDATA62 S48C7:S55C7	LCDDATA63 S56C7:S63C7

13.6 BIAS CONFIGURATIONS

PIC18F97J94 family devices have four distinct circuit configurations for LCD bias generation:

- M0: Regulator with Boost
- M1: Regulator without Boost
- M2: Resistor Ladder with Software Contrast
- M3: Resistor Ladder with Hardware Contrast

13.6.1 M0 (REGULATOR WITH BOOST)

In M0 operation, the LCD charge pump feature is enabled. This allows the regulator to generate voltages up to +3.6V to the LCD (as measured at LCDBIAS3).

M0 uses a flyback capacitor connected between VLCAP1 and VLCAP2, as well as filter capacitors on LCDBIAS0 through LCDBIAS3, to obtain the required voltage boost (Figure 13-6). The output voltage (VBIAS) is the difference of the potential between LCDBIAS3 and LCDBIAS0. It is set by the BIAS<2:0> bits which adjust the offset between LCDBIAS0 and VSS. The flyback capacitor (CFLY) acts as a charge storage element for large LCD loads. This mode is useful in those cases where the voltage requirements of the LCD are higher than the microcontroller's VDD. It also permits software control of the display's contrast, by adjustment of bias voltage, by changing the value of the BIAS bits.

M0 supports static and 1/3 bias types. Generation of the voltage levels for 1/3 bias is handled automatically, but must be configured in software.

M0 is enabled by selecting a valid regulator clock source (CLKSEL<1:0> set to any value except '00') and setting the CPEN bit. If static bias type is required, the MODE13 bit must be cleared.

13.6.2 M1 (REGULATOR WITHOUT BOOST)

M1 operation is similar to M0, but does not use the LCD charge pump. It can provide VBIAS up to the voltage level supplied directly to LCDBIAS3. It can be used in cases where VDD for the application is expected to never drop below a level that can provide adequate contrast for the LCD. The connection of external components is very similar to M0, except that LCDBIAS3 must be tied directly to VDD (Figure 13-6).

Note:	When the device is put to Sleep while operating in mode M0 or M1, make sure that the bias capacitors are fully discharged to get the lowest Sleep current.

The BIAS<2:0> bits can still be used to adjust contrast in software by changing the VBIAS. As with M0, changing these bits changes the offset between LCDBIAS0 and VSS. In M1, this is reflected in the change between the LCDBIAS0 and the voltage tied to LCDBIAS3. Thus, if VDD should change, VBIAS will also change; where in M0, the level of VBIAS is constant.

Like M0, M1 supports static and 1/3 bias types. Generation of the voltage levels for 1/3 bias is handled automatically but must be configured in software. M1 is enabled by selecting a valid regulator clock source (CLKSEL<1:0> set to any value except '00') and clearing the CPEN bit. If 1/3 bias type is required, the MODE13 bit should also be set.

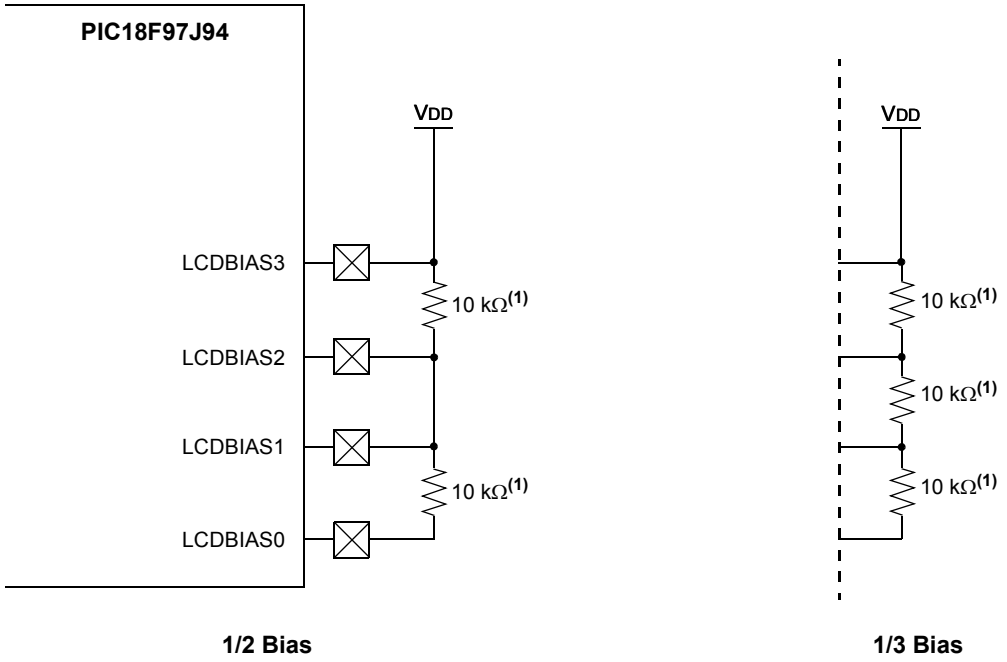
13.6.3 M2 (EXTERNAL RESISTOR LADDER WITH SOFTWARE CONTRAST)

M2 operation also uses the LCD regulator but disables the charge pump. The regulator's internal voltage reference remains active as a way to regulate contrast. It is used in cases where the current requirements of the LCD exceed the capacity of the regulator's charge pump.

In this configuration, the LCD bias voltage levels are created by an external resistor voltage divider, connected across LCDBIAS0 through LCDBIAS3, with the top of the divider tied to VDD (Figure 13-7). The potential at the bottom of the ladder is determined by the LCD regulator's voltage reference, tied internally to LCDBIAS0. The bias type is determined by the voltages on the LCDBIAS pins, which are controlled by the configuration of the resistor ladder. Most applications, using M2, will use a 1/3 or 1/2 bias type. While static bias can also be used, it offers extremely limited contrast range and additional current consumption over other bias generation modes.

Like M1, the LCDBIAS bits can be used to control contrast, limited by the level of VDD supplied to the device. Also, since there is no capacitor required across VLCAP1 and VLCAP2, these pins are available as digital I/O ports, RG2 and RG3. M2 is selected by clearing the CLKSEL<1:0> bits and setting the CPEN bit.

FIGURE 13-7: RESISTOR LADDER CONNECTIONS FOR M2 CONFIGURATION



1/2 Bias **1/3 Bias**

Bias Level at Pin	Bias Type	
	1/2 Bias	1/3 Bias
LCDBIAS0	(Internal Low Reference Voltage)	(Internal Low Reference Voltage)
LCDBIAS1	1/2 VBIAS	1/3 VBIAS
LCDBIAS2	1/2 VBIAS	2/3 VBIAS
LCDBIAS3	VBIAS (up to VDD)	VBIAS (up to VDD)

Note 1: These values are provided for design guidance only; they should be optimized for the application by the designer based on the actual LCD specifications.

20.5.20.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDAx pin has been deasserted and allowed to float high, SDAx is sampled low after the BRG has timed out.

b) After the SCLx pin is deasserted, SCLx is sampled low before SDAx goes high.

The Stop condition begins with SDAx asserted low. When SDAx is sampled low, the SCLx pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD<6:0> and counts down to 0. After the BRG times out, SDAx is sampled. If SDAx is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 20-33). If the SCLx pin is sampled low before SDAx is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 20-34).

FIGURE 20-33: BUS COLLISION DURING A STOP CONDITION (CASE 1)

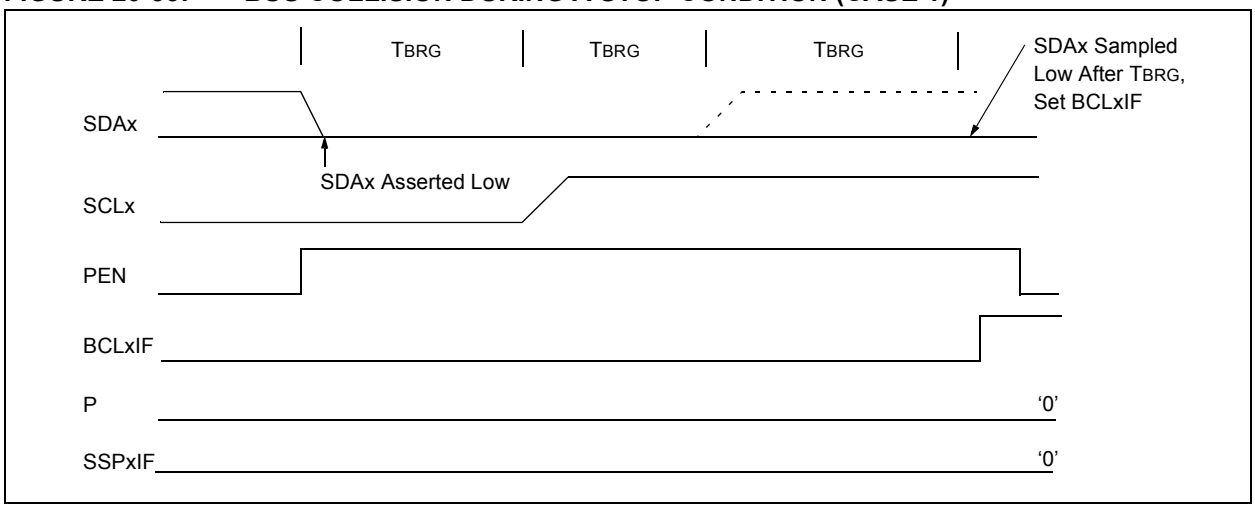
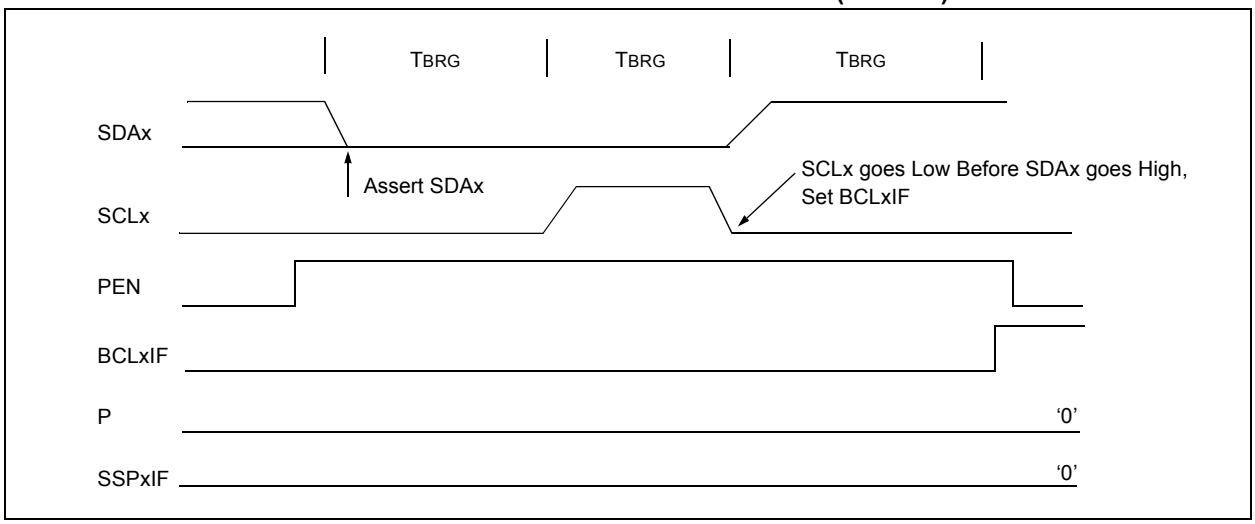


FIGURE 20-34: BUS COLLISION DURING A STOP CONDITION (CASE 2)



REGISTER 22-7: ADCON2L: A/D CONTROL REGISTER 2 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS(1)	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM(1)	ALTS
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 **BUFS:** Buffer Fill Status bit(1)
- 1 = A/D is filling the upper half of the buffer; user should access data in the lower half
- 0 = A/D is filling the lower half of the buffer; user should access data in the upper half

bit 6-2 **SMPI<4:0>:** Interrupt Sample Increment Rate Select bits

Selects the number of sample/conversions per each interrupt.
- 11111 = Interrupt/address increment at the completion of conversion for each 32nd sample
- 11110 = Interrupt/address increment at the completion of conversion for each 31st sample
- • • •
- 00001 = Interrupt/address increment at the completion of conversion for every other sample
- 00000 = Interrupt/address increment at the completion of conversion for each sample

bit 1 **BUFM:** Buffer Fill Mode Select bit(1)
- 1 = A/D buffer is two, 13-word buffers, starting at ADC1BUF0 and ADC1BUF12, and sequential conversions fill the buffers alternately (Split mode)
- 0 = A/D buffer is a single, 26-word buffer and fills sequentially from ADC1BUF0 (FIFO mode)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
- 1 = Uses channel input selects for Sample A on first sample and Sample B on next sample
- 0 = Always uses channel input selects for Sample A

Note 1: These bits are only applicable when the buffer is used in FIFO mode (BUFREGEN = 0). In addition, BUFS is only used when BUFM = 1.

24.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 32-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram of the module is shown in Figure 24-1. The resistor ladder is segmented to provide a range of CV_{REF} values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device V_{DD}/V_{SS} or an external voltage reference.

24.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCONH register (Register 24-1). The comparator voltage reference provides a range of output voltage with 32 levels.

The CVR<4:0> selection bits (CVRCONH<4:0>) offer a range of output voltages. Equation 24-1 shows how the comparator voltage reference is computed.

EQUATION 24-1:

If CVRSS = 1:

$$CV_{REF} = (V_{REF-} + \frac{CVR<4:0>}{32}) \bullet (V_{REF+} - V_{REF-})$$

If CVRSS = 0:

$$CV_{REF} = (AV_{SS} + \frac{CVR<4:0>}{32}) \bullet (AV_{DD} - AV_{SS})$$

The comparator voltage reference supply can come from either V_{DD} and V_{SS}, or the external V_{REF+} and V_{REF-} that are multiplexed with RA3 and RA2. The voltage source is selected by the CVRPSS<1:0> bits (CVRCONL<5:4>).

The settling time of the comparator voltage reference must be considered when changing the CV_{REF} output (see Table 30-13 in **Section 30.0 "Electrical Specifications"**).

REGISTER 24-1: CVRCONH: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER HIGH

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CVR4	CVR3	CVR2	CVR1	CVR0
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 4-0 **CVR<4:0>:** Comparator V_{REF} Value Selection $0 \le CVR<4:0> \le 31$ bits

$CV_{REF} = V_{NEGSRC} + (CVR<4:0>/32) \bullet (V_{POSSRC} - V_{NEGSRC})$

EXAMPLE 26-1: SETUP FOR CTMU CALIBRATION ROUTINES

```
#include "p18cxxx.h"
/*****************************************************************************/
/*Setup CTMU *****************************************************************/
/*****************************************************************************/
void setup(void)

{ //CTMUCON - CTMU Control register

    CTMUCON = 0x00;              //make sure CTMU is disabled
    CTMUCON3 = 0x90;
    //CTMU continues to run when emulator is stopped,CTMU continues
    //to run in idle mode,Time Generation mode disabled, Edges are blocked
    //No edge sequence order, Analog current source not grounded, trigger
    //output disabled, Edge2 polarity = positive level, Edge2 source =
    //source 0, Edge1 polarity = positive level, Edge1 source = source 0,
    // Set Edge status bits to zero


    //CTMUCON1 - CTMU Current Control Register
    CTMUCON1 = 0x01;             //0.55uA, Nominal - No Adjustment

/*****************************************************************************/
//Setup AD converter;
/*****************************************************************************/

 TRISBbits.TRISB0=0;
    TRISAbits.TRISA2=1;                     //set channel 2 as an input
    ANCON1bits.ANSEL2=1;                    // Configured AN2 as an analog channel
    ADCON1Hbits.FORM=0b00;                  // Result format 1= Right justified
    ADCON1Lbits.SSRC=0b0111;
    ADCON3Hbits.SAMC=0b00111;               // Acquisition time 7 = 20TAD 2 = 4TAD 1=2TAD
    ADCON3Lbits.ADCS=0x3F;                  // Clock conversion bits 6= FOSC/64 2=FOSC/32
                                            // ADCON1
    ADCON2Hbits.PVCFG=0b00;                 // Vref+ = AVdd
    ADCON2Hbits.NVCFG0=0;                   // Vref- = AVss
    ADCHS0Lbits.CHONA=0b000;
    ADCHS0Lbits.CHOSA=0b00010;              // Select ADC channel
    ADCON1Hbits.ADON=1; // Turn on ADC

}
```

EXAMPLE 26-4: CTMU ROUTINE FOR CAPACITIVE TOUCH SWITCH

```
#include "p18cxxx.h"

#define COUNT 500                       //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define OPENSW 1000                     //Un-pressed switch value
#define TRIP 300                        //Difference between pressed
                                        //and un-pressed switch
#define HYST 65                         //amount to change
                                        //from pressed to un-pressed
#define PRESSED 1
#define UNPRESSED 0

int main(void)
{
    unsigned int Vread;                 //storage for reading
    unsigned int switchState;
    int i;

    //assume CTMU and A/D have been setup correctly
    //see Example 25-1 for CTMU & A/D setup
    setup();

    CTMUCONbits.CTMUEN = 1;             //Enable the CTMU

    CTMUCONbits.IDISSEN = 1;            //drain charge on the circuit
    DELAY;                              //wait 125us
    CTMUCONbits.IDISSEN = 0;            //end drain of circuit

    CTMUCON3bits.EDG1STAT = 1;          //Begin charging the circuit
                                        //using CTMU current source
    DELAY;                              //wait for 125us
    CTMUCON3bits.EDG1STAT = 0;          //Stop charging circuit

    PIR1bits.ADIF = 0;                  //make sure A/D Int not set
    ADCON1Lbits.SAMP=1;;                //and begin A/D conv.
    while(!PIR1bits.ADIF);              //Wait for A/D convert complete

    Vread = ADRES;                      //Get the value from the A/D

    if(Vread < OPENSW - TRIP)
    {
        switchState = PRESSED;
    }
    else if(Vread > OPENSW - TRIP + HYST)
    {
        switchState = UNPRESSED;
    }
}
```

BCF **Bit Clear f**

Syntax: BCF f, b {,a}

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

Operation: $0 \rightarrow f<b>$

Status Affected: None

Encoding:

1001	bbba	ffff	ffff

Description: Bit 'b' in register 'f' is cleared.

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 register 'f'

Example: `BCF FLAG_REG, 7, 0`

Before Instruction
FLAG_REG = C7h
After Instruction
FLAG_REG = 47h

BN **Branch if Negative**

Syntax: BN n

Operands: $-128 \le n \le 127$

Operation: if Negative bit is '1',
$(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0110	nnnn	nnnn

Description: If the Negative bit is '1', then the program will branch.

The 2's complement number '2n' is added 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 then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

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

If No Jump:

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

Example:
```
HERE    BN   Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Negative = 1;
PC = address (Jump)
If Negative = 0;
PC = address (HERE + 2)

DECFSZ **Decrement f, Skip if 0**

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

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

Operation: (f) – 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

0010	11da	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a 2-cycle instruction.

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(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	Write to destination

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    DECFSZ  CNT, 1, 1
        GOTO    LOOP
CONTINUE
```
Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT – 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT ≠ 0;
PC = Address (HERE + 2)

DCFSNZ **Decrement f, Skip if Not 0**

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

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

0100	11da	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a 2-cycle instruction.

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(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	Write to destination

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    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```
Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP – 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

MOVFF **Move f to f**

Syntax: 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	$ffff_s$
2nd word (destin.)	1111	ffff	ffff	$ffff_d$

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

Words: 2

Cycles: 2

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 = 33h
REG2 = 11h
After Instruction
REG1 = 33h
REG2 = 33h

MOVLB **Move Literal to Low Nibble in BSR**

Syntax: 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). The value of BSR<7:4> always remains '0' regardless of the value of k_7:k_4.

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 = 02h
After Instruction
BSR Register = 05h

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