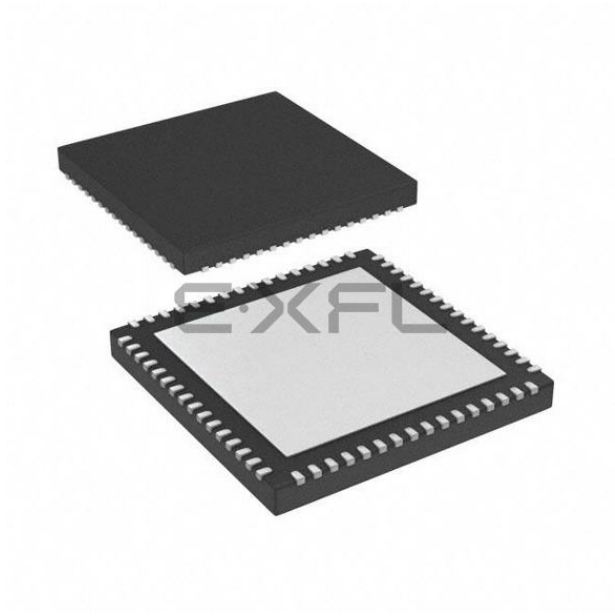


Welcome to E-XFL.COM

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	52
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 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f67j94t-i-mr

3.2 Oscillator Configuration

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The Oscillator Configuration bit settings are in the Configuration registers located in the program memory (refer to **Section 28.1 "Configuration Bits"** for more information). The Primary Oscillator Configuration bits, POSCMD<1:0> (CONFIG3L<1:0>), and Oscillator Configuration bits, FOSC<2:0> (CONFIG2L<2:0>), select the oscillator source that is used at a POR. The FRC Oscillator with Postscaler (FRCDIV) is the default (unprogrammed) selection. The Secondary Oscillator, or one of the internal oscillators, may be chosen by programming these bit locations.

The Configuration bits allow users to choose between 11 different clock modes, as shown in Table 3-1.

TABLE 3-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FOSC<2:0>	Notes
Fast RC Oscillator with Postscaler (FRCDIV)	Internal	11	111	**1, 2**
Fast RC Oscillator divided by 16 (FRC500kHz)	Internal	11	110	**1**
Low-Power RC Oscillator (LPRC)	Internal	11	101	**1**
Secondary (Timer1) Oscillator (SOSC)	Secondary	11	100	**1**
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (MS) with PLL Module (MSPLL)	Primary	01	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (MS)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
Fast RC Oscillator with PLL Module (FRCPLL)	Internal	11	001	**1**
Fast RC Oscillator (FRC)	Internal	11	000	**1**

Note 1: OSC2 pin function is determined by the CLKOEN Configuration bit.

2: Default oscillator mode for an unprogrammed (erased) device.

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
ALRMRPT	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
ALRMVALH	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
ALRMVALL	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
RTCCON2	64-pin	80-pin	100-pin	0000 0000	uuuu uuuu	uuuu uuuu
IOCP	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
IOCN	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PADCFG1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CM1CON	64-pin	80-pin	100-pin	0001 1111	0001 1111	uuuu uuuu
ECCP2AS	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CCPR2H	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR2L	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
ECCP2CON	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
ECCP3AS	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
ECCP3DEL	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CCPR3H	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR3L	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
ECCP3CON	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CCPR8H	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR8L	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP8CON	64-pin	80-pin	100-pin	--00 0000	--00 0000	--uu uuuu
CCPR9H	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR9L	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP9CON	64-pin	80-pin	100-pin	--00 0000	--00 0000	--uu uuuu
CCPR10H	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR10L	64-pin	80-pin	100-pin	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP10CON	64-pin	80-pin	100-pin	--00 0000	--00 0000	--uu uuuu
TMR6	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PR6	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
T6CON	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu uuuu
TMR8	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PR8	64-pin	80-pin	100-pin	1111 1111	1111 1111	uuuu uuuu
T8CON	64-pin	80-pin	100-pin	-000 0000	-000 0000	-uuu 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.

6.0 MEMORY ORGANIZATION

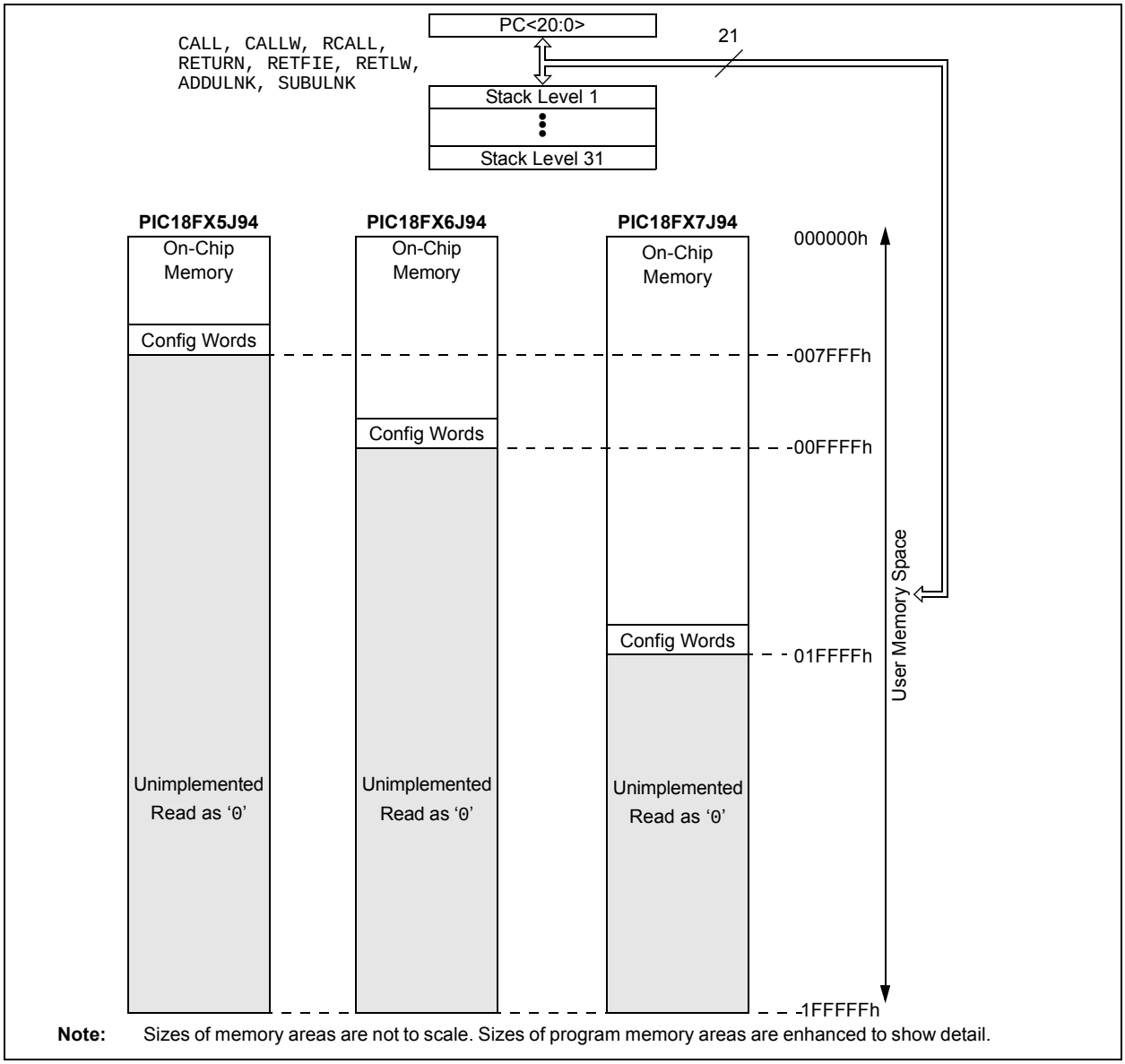
PIC18FXXJ94 devices have these types of memory:

- 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.

TABLE 6-2: REGISTER FILE SUMMARY (CONTINUED)

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F36h	CCPTMRS0	C3TSEL1	C3TSEL0	C2TSEL2	C2TSEL1	C2TSEL0	C1TSEL2	C1TSEL1	C1TSEL0
F35h	CCPTMRS1	C7TSEL1	C7TSEL0	—	C6TSEL0	—	C5TSEL0	C4TSEL1	C4TSEL0
F34h	CCPTMRS2	—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0
F33h	RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
F32h	TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
F31h	BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	IREN	WUE	ABDEN
F30h	SPBRGH1	EUSART1 Baud Rate Generator High Byte							
F2Fh	RCSTA3	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
F2Eh	TXSTA3	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
F2Dh	BAUDCON3	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	IREN	WUE	ABDEN
F2Ch	SPBRGH3	EUSART3 Baud Rate Generator High Byte							
F2Bh	SPBRG3	EUSART3 Baud Rate Generator							
F2Ah	RCREG3	EUSART3 Receive Data FIFO							
F29H	TXREG3	EUSART3 Transmit Data FIFO							
F28h	DSCONL	—	—	—	—	—	ULPWDIS	DSBOR	RELEASE
F27h	DSCONH	DSEN	—	—	—	—	—	—	RTCWDIS
F26h	DSWAKEL	DSFLT	BOR	DSULP	DSWDT	DSRTC	DSMCLR	DSICD	DSPOR
F25h	DSWAKEH	—	—	—	—	—	—	—	DSINT0
F24h	DSGPR0	Deep Sleep General Purpose Register 0							
F23h	DSGPR1	Deep Sleep General Purpose Register 1							
F22h	DSGPR2	Deep Sleep General Purpose Register 2							
F21h	DSGPR3	Deep Sleep General Purpose Register 3							
F20h	SPBRGH2	EUSART2 Baud Rate Generator High Byte							
F1Fh	SPBRG2	EUSART2 Baud Rate Generator							
F1Eh	RCREG2	Receive Data FIFO							
F1Dh	TXREG2	Transmit Data FIFO							
F1Ch	PSTR2CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
F1Bh	PSTR3CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
F1Ah	SSP2STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
F19h	SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
F18h	SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
F17h	SSP2MSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
F16h	TMR5H	Timer5 Register High Byte							
F15h	TMR5L	Timer5 Register Low Byte							
F14h	T5CON	TMR5CS1	TMR5CS0	T5CKPS1	T5CKPS0	SOSCEN	$\overline{T5SYNC}$	RD16	TMR5ON
F13h	T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{T5DONE}$	T5GVAL	T5GSS1	T5GSS0
F12h	CCPR4H	Capture/Compare/PWM Register 4 High Byte							
F11h	CCPR4L	Capture/Compare/PWM Register 4 Low Byte							
F10h	CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0
F0Fh	CCPR5H	Capture/Compare/PWM Register 5 High Byte							
F0Eh	CCPR5L	Capture/Compare/PWM Register 5 Low Byte							
F0Dh	CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0
F0Ch	CCPR6H	Capture/Compare/PWM Register 6 High Byte							
F0Bh	CCPR6L	Capture/Compare/PWM Register 6 Low Byte							
F0Ah	CCP6CON	—	—	DC6B1	DC6B0	CCP6M3	CCP6M2	CCP6M1	CCP6M0
F09h	CCPR7H	Capture/Compare/PWM Register 7 High Byte							
F08h	CCPR7L	Capture/Compare/PWM Register 7 Low Byte							
F07h	CCP7CON	—	—	DC7B1	DC7B0	CCP7M3	CCP7M2	CCP7M1	CCP7M0
F06h	TMR4	Timer4 Register							
F05h	PR4	Timer4 Period Register							
F04h	T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0

Legend: — = unimplemented, read as '0'.

7.2.2 TABLE LATCH REGISTER (TABLAT)

The Table Latch (TABLAT) is an 8-bit register mapped into the Special Function Register (SFR) space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

7.2.3 TABLE POINTER REGISTER (TBLPTR)

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the Device ID, the User ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways, based on the table operation. These operations are shown in Table 7-1 and only affect the low-order 21 bits.

7.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into TABLAT.

When a TBLWT is executed, the seven Least Significant bits (LSbs) of the Table Pointer register (TBLPTR<6:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 12 Most Significant bits (MSbs) of the TBLPTR (TBLPTR<21:10>) determine which program memory block of 1024 bytes is written to. For more detail, see **Section 7.5 "Writing to Flash Program Memory"**.

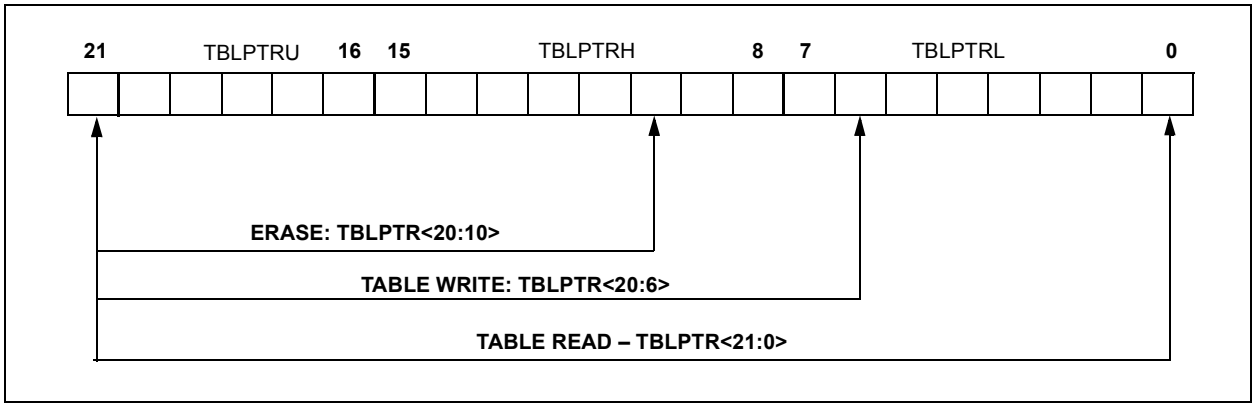
When an erase of program memory is executed, the 12 MSbs of the Table Pointer register point to the 1024-byte block that will be erased. The LSbs are ignored.

Figure 7-3 describes the relevant boundaries of the TBLPTR based on Flash program memory operations.

TABLE 7-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

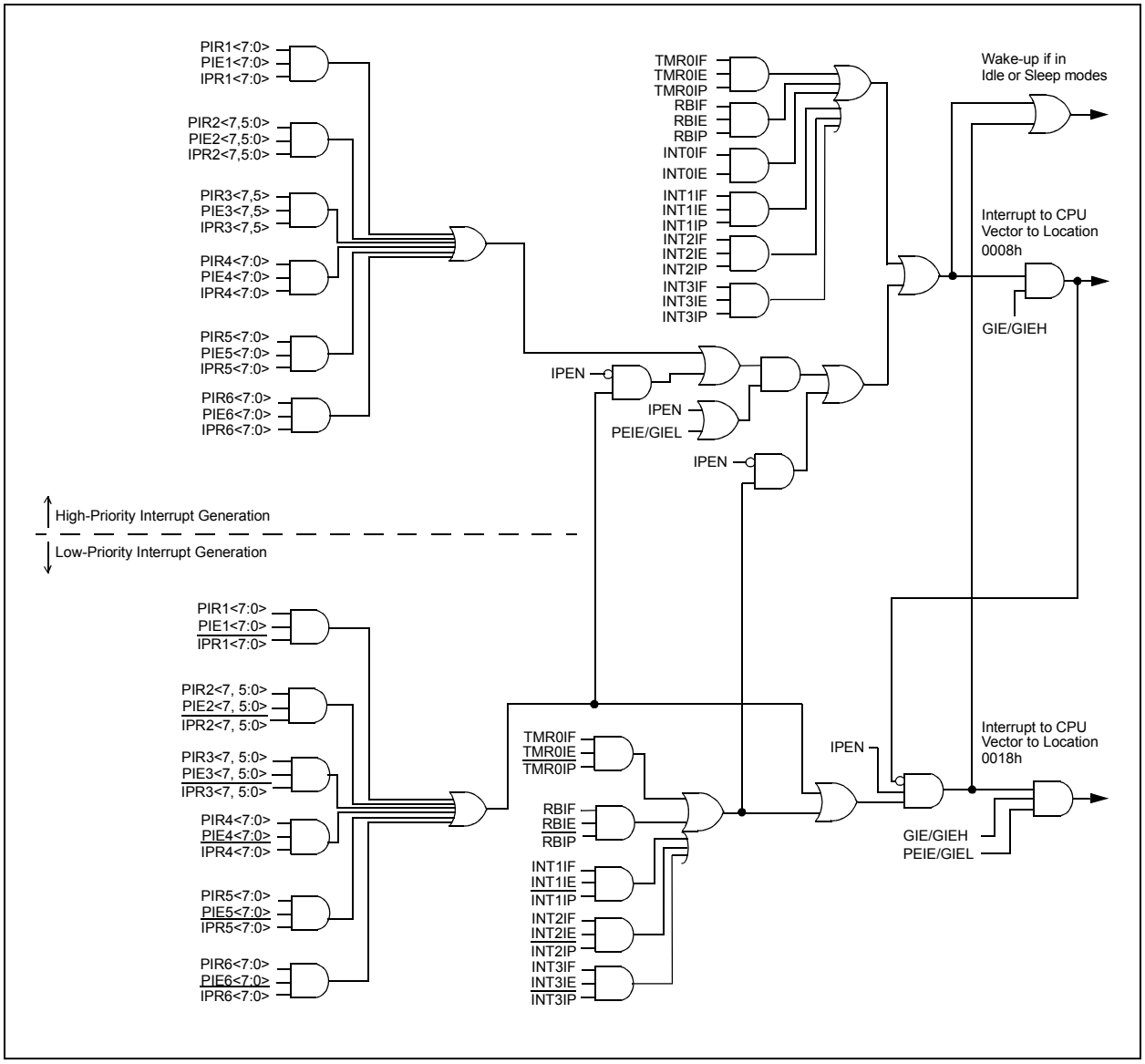
FIGURE 7-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



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

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

FIGURE 10-1: PIC18F97J94 FAMILY INTERRUPT LOGIC



REGISTER 10-23: IOCP: INTERRUPT-ON-CHANGE POSITIVE EDGE 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
IOCP7	IOCP6	IOCP5	IOCP4	IOCP3	IOCP2	IOCP1	IOCP0
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 **IOCP<7:0>:** Interrupt-on-Change Positive Edge Enable bits

1 = Interrupt-on-change is enabled on the pin for a rising edge; associated Status bit and interrupt flag will be set upon detecting an edge

0 = Interrupt-on-change is disabled for the associated pin

REGISTER 10-24: IOCN: INTERRUPT-ON-CHANGE NEGATIVE EDGE 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
IOCN7	IOCN6	IOCN5	IOCN4	IOCN3	IOCN2	IOCN1	IOCN0
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 **IOCN<7:0>:** Interrupt-on-Change Negative Edge Enable bits

1 = Interrupt-on-change is enabled on the pin for a falling edge; associated Status bit and interrupt flag will be set upon detecting an edge

0 = Interrupt-on-change is disabled for the associated pin

REGISTER 10-25: IOCF: INTERRUPT-ON-CHANGE FLAG 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
IOCF7	IOCF6	IOCF5	IOCF4	IOCF3	IOCF2	IOCF1	IOCF0
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 **IOCF<7:0>:** Interrupt-on-Change Flag bits

1 = An enabled change was detected on the associated pin; this is set when IOCP<x> = 1 and a positive edge was detected on the input pin or when IOCN<x> = 1 and a negative edge was detected on the input pin (clear in software to clear the IOCIF bit)

0 = No change was detected or the user cleared the detected change

TABLE 11-7: PORTG FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RG3/RP43/ C3INB/AN17/ SEG31/COM7	RG3	0	O	DIG	LATG<3> data output; not affected by analog input.
		1	I	ST	PORTG<3> data input; disabled when analog input is enabled.
	RP43	x	x	DIG	Reconfigurable Pin 43 for PPS-Lite; TRIS must be set to match input/ output of module.
	C3INB	1	I	ANA	Comparator 3 Input B.
	AN17	1	I	ANA	A/D Input Channel 17. Default input configuration on POR; does not affect digital output.
	SEG31	0	O	ANA	LCD Segment 31 output; disables all other pin functions.
	COM7	x	O	ANA	LCD Common 7 output; disables all other outputs.
RG4/RTCC/ RP44/C3INC/ AN16/SEG26	RG4	0	O	DIG	LATG<4> data output; not affected by analog input.
		1	I	ST	PORTG<4> data input; disabled when analog input is enabled.
	RTCC	x	O	DIG	RTCC output.
	RP44	x	x	DIG	Reconfigurable Pin 44 for PPS-Lite; TRIS must be set to match input/ output of module.
	C3INC	1	I	ANA	Comparator 3 Input C.
	AN16	1	I	ANA	A/D Input Channel 16. Default input configuration on POR; does not affect digital output.
	SEG26	0	O	ANA	LCD Segment 26 output; disables all other pin functions.
RG6	RG6	0	O	DIG	LATG<6> data output.
		1	I	ST	PORTG<6> data input.
RG7	RG7	0	O	DIG	LATG<7> data output.
		1	I	ST	PORTG<7> data input.

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

TABLE 11-9: PORTJ FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RJ6/SEG37/$\overline{LB}$	RJ6	0	O	DIG	LATJ<6> data output.
		1	I	ST	PORTJ<6> data input.
	SEG37	0	O	ANA	LCD Segment 37 output; disables all other pin functions.
	$\overline{LB}$	x	O	DIG	External Memory Bus Lower Byte ($\overline{LB}$) signal.
RJ7/SEG36/$\overline{UB}$	RJ7	0	O	DIG	LATJ<7> data output.
		1	I	ST	PORTJ<7> data input.
	SEG36	0	O	ANA	LCD Segment 36 output; disables all other pin functions.
	$\overline{UB}$	x	O	DIG	External Memory Bus Upper Byte ($\overline{UB}$) signal.

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

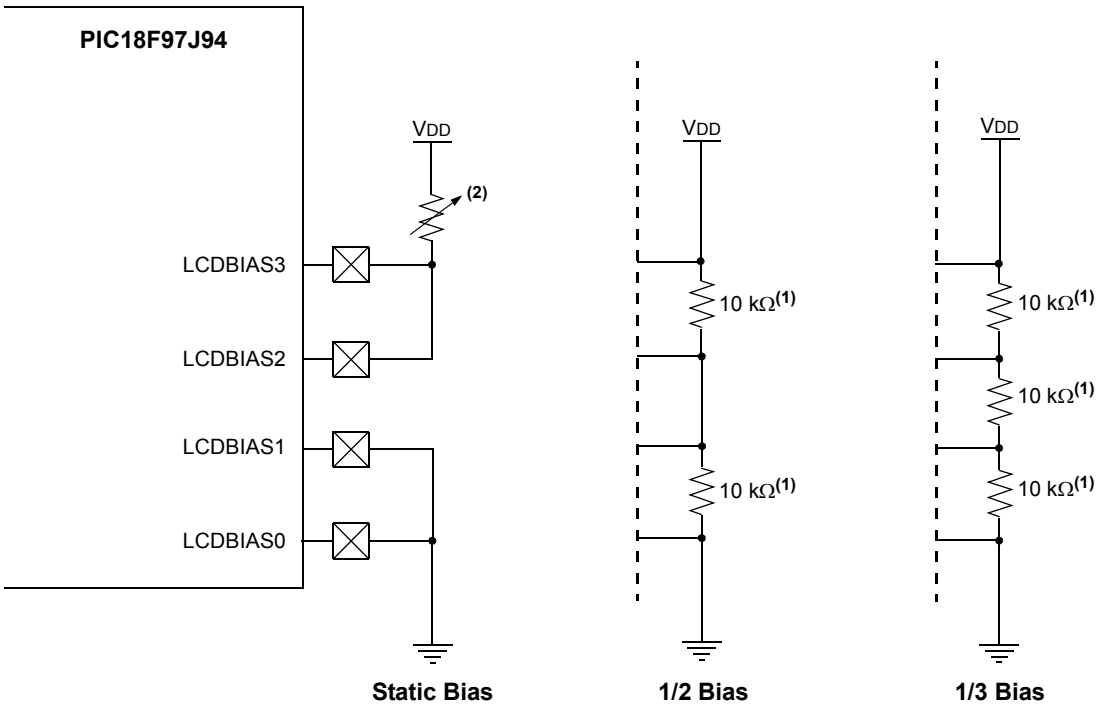
13.6.4 M3 (HARDWARE CONTRAST)

In M3, the LCD regulator is completely disabled. Like M2, LCD bias levels are tied to VDD and are generated using an external divider. The difference is that the internal voltage reference is also disabled and the bottom of the ladder is tied to ground (VSS); see Figure 13-8. The value of the resistors, and the difference between VSS and VDD, determine the contrast range; no software adjustment is possible. This configuration is also used where the LCD's current requirements exceed the capacity of the charge pump and software contrast control is not needed.

Depending on the bias type required, resistors are connected between some or all of the pins. A potentiometer can also be connected between LCDBIAS3 and VDD to allow for hardware controlled contrast adjustment.

M3 is selected by clearing the CLKSEL<1:0> and CPEN bits.

FIGURE 13-8: RESISTOR LADDER CONNECTIONS FOR M3 CONFIGURATION



Bias Level at Pin	Bias Type		
	Static	1/2 Bias	1/3 Bias
LCDBIAS0	AVSS	AVSS	AVSS
LCDBIAS1	AVSS	1/2 VDD	1/3 VDD
LCDBIAS2	VDD	1/2 VDD	2/3 VDD
LCDBIAS3	VDD	VDD	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.

2: A potentiometer for manual contrast adjustment is optional; it may be omitted entirely.

REGISTER 15-1: TxCON: TIMERx CONTROL 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
TMRxCS1	TMRxCS0	TxCKPS1	TxCKPS0	SOSCEN	$\overline{\text{TxSYNC}}$	RD16	TMRxON
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 **TMRxCS<1:0>:** Timerx Clock Source Select bits

11 = Timerx Clock source is INTOSC
10 = Timerx clock source depends on the SOSCEN bit:
SOSCEN = 0:
External clock from the TxCKI pin (on the rising edge).
SOSCEN = 1:
Depending on the SOSCSEL fuses, either a crystal oscillator on the SOSCI/SOSCO pins or an external clock from the SCLKI pin.
01 = Timerx clock source is the system clock (FOSC)**(1)**
00 = Timerx clock source is the instruction clock (FOSC/4)

bit 5-4 **TxCKPS<1:0>**: Timerx Input Clock Prescale Select bits

11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **SOSCEN**: SOSC Oscillator Enable bit

1 = SOSC/SCLKI are enabled for Timerx (based on the SOSCSEL fuses)
0 = SOSC/SCLKI are disabled for Timerx and TxCKI is enabled

bit 2 **$\overline{\text{TxSYNC}}$:** Timerx External Clock Input Synchronization Control bit
(Not usable if the device clock comes from Timer1/3/5.)
When TMRxCS<1:0> = 10:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMRxCS<1:0> = 0x:
This bit is ignored; Timer1/3/5 uses the internal clock.

bit 1 **RD16:** 16-Bit Read/Write Mode Enable bit

1 = Enables register read/write of Timerx in one 16-bit operation
0 = Enables register read/write of Timerx in two 8-bit operations

bit 0 **TMRxON:** Timerx On bit

1 = Enables Timerx
0 = Stops Timerx

Note 1: The FOSC clock source should not be selected if the timer will be used with the ECCP capture/compare features.

REGISTER 19-2: CCPTMRS1: CCP TIMER SELECT REGISTER 1

R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C7TSEL1	C7TSEL0	—	C6TSEL0	—	C5TSEL0	C4TSEL1	C4TSEL0
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 **C7TSEL<1:0>:** CCP7 Timer Selection bits
00 =CCP7 is based off of TMR1/TMR2
01 =CCP7 is based off of TMR5/TMR4
10 =CCP7 is based off of TMR5/TMR6
11 =CCP7 is based off of TMR5/TMR8

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **C6TSEL0:** CCP6 Timer Selection bit
0 = CCP6 is based off of TMR1/TMR2
1 = CCP6 is based off of TMR5/TMR2

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **C5TSEL0:** CCP5 Timer Selection bit
0 = CCP5 is based off of TMR1/TMR2
1 = CCP5 is based off of TMR5/TMR4

bit 1-0 **C4TSEL<1:0>:** CCP4 Timer Selection bits
00 =CCP4 is based off of TMR1/TMR2
01 =CCP4 is based off of TMR3/TMR4
10 =CCP4 is based off of TMR3/TMR6
11 =Reserved; do not use

20.4.4.1 DMACON1

The DMACON1 register is used to select the main operating mode of the SPI DMA module. The SSCON1 and SSCON0 bits are used to control the slave select pin.

When MSSP1 is used in SPI Master mode with the SPI DMA module, SSDMA can be controlled by the DMA module as an output pin. If MSSP1 will be used to communicate with an SPI slave device that needs the SSx pin to be toggled periodically, the SPI DMA hardware can automatically be used to de-assert SSx between each byte, every two bytes or every four bytes.

Alternatively, user firmware can manually generate slave select signals with normal general purpose I/O pins, if required by the slave device(s).

When the TXINC bit is set, the TXADDR register will automatically increment after each transmitted byte. Automatic transmit address increment can be disabled by clearing the TXINC bit. If the automatic transmit address increment is disabled, each byte which is output on SDO will be the same (the contents of the SRAM pointed to by the TXADDR register) for the entire DMA transaction.

When the RXINC bit is set, the RXADDR register will automatically increment after each received byte. Automatic receive address increment can be disabled by clearing the RXINC bit. If RXINC is disabled in Full-Duplex or Half-Duplex Receive modes, all incoming data bytes on SDI will overwrite the same memory location pointed to by the RXADDR register. After the SPI DMA transaction has completed, the last received byte will reside in the memory location pointed to by the RXADDR register.

The SPI DMA module can be used for either half-duplex receive only communication, half-duplex transmit only communication or full-duplex simultaneous transmit and receive operations. All modes are available for both SPI master and SPI slave configurations. The DUPLEX0 and DUPLEX1 bits can be used to select the desired operating mode.

The behavior of the DLYINTEN bit varies greatly depending on the SPI operating mode. For example behavior for each of the modes, see Figure 20-3 through Figure 20-6.

SPI Slave mode, DLYINTEN = 1: In this mode, an SSP1IF interrupt will be generated during a transfer if the time between successful byte transmission events is longer than the value set by the DLYCYC<3:0> bits in the DMACON2 register. This interrupt allows slave firmware to know that the master device is taking an unusually large amount of time between byte transmissions. For example, this information may be useful for implementing application defined communication protocols, involving time-outs if the bus remains Idle for too long. When DLYINTEN = 1, the DLYLVL<3:0> interrupts occur normally according to the selected setting.

SPI Slave mode, DLYINTEN = 0: In this mode, the time-out based interrupt is disabled. No additional SSP1IF interrupt events will be generated by the SPI DMA module, other than those indicated by the INTLVL<3:0> bits in the DMACON2 register. In this mode, always set DLYCYC<3:0> = 0000.

SPI Master mode, DLYINTEN = 0: The DLYCYC<3:0> bits in the DMACON2 register determine the amount of additional inter-byte delay, which is added by the SPI DMA module during a transfer; the Master mode SS1 output feature may be used.

SPI Master mode, DLYINTEN = 1: The amount of hardware overhead is slightly reduced in this mode, and the minimum inter-byte delay is 8 TCY for FOSC/4, 9 TCY for FOSC/16 and 15 TCY for FOSC/64. This mode can potentially be used to obtain slightly higher effective SPI bandwidth. In this mode, the SS1 control feature cannot be used and should always be disabled (DMACON1<7:6> = 00). Additionally, the interrupt generating hardware (used in Slave mode) remains active. To avoid extraneous SSP1IF interrupt events, set the DMACON2 Delay bits, DLYCYC<3:0> = 1111, and ensure that the SPI serial clock rate is no slower than FOSC/64.

In SPI Master modes, the DMAEN bit is used to enable the SPI DMA module and to initiate an SPI DMA transaction. After user firmware sets the DMAEN bit, the DMA hardware will begin transmitting and/or receiving data bytes according to the configuration used. In SPI Slave modes, setting the DMAEN bit will finish the initialization steps needed to prepare the SPI DMA module for communication (which must still be initiated by the master device).

To avoid possible data corruption, once the DMAEN bit is set, user firmware should not attempt to modify any of the MSSP2 or SPI DMA related registers, with the exception of the INTLVLx bits in the DMACON2 register.

If user firmware wants to halt an ongoing DMA transaction, the DMAEN bit can be manually cleared by the firmware. Clearing the DMAEN bit while a byte is currently being transmitted will not immediately halt the byte in progress. Instead, any byte currently in progress will be completed before the MSSP1 and SPI DMA modules go back to their Idle conditions. If user firmware clears the DMAEN bit, the TXADDR, RXADDR and DMABC registers will no longer update, and the DMA module will no longer make any additional read or writes to SRAM; therefore, state information can be lost.

REGISTER 20-7: SSPxCON1: MSSPx CONTROL REGISTER 1 (I^2C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
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 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPxBUF register was attempted while the I^2C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

In Transmit mode:

This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCKx Release Control bit

In Slave mode:

1 = Releases clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)

1111 = I^2C Slave mode: 10-bit address with Start and Stop bit interrupts enabled

1110 = I^2C Slave mode: 7-bit address with Start and Stop bit interrupts enabled

1011 = I^2C Firmware Controlled Master mode (slave Idle)

1001 = Load SSPxMSK register at SSPxADD SFR address(3,4)

1000 = I^2C Master mode: Clock = FOSC/(4 * (SSPxADD + 1))

0111 = I^2C Slave mode: 10-bit address(3,4)

0110 = I^2C Slave mode: 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.

2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

3: When SSPM<3:0> = 1001, any reads or writes to the SSPxADD SFR address actually accesses the SSPxMSK register.

4: This mode is only available when 7-Bit Address Masking mode is selected (MSSPMSK Configuration bit is '1').

REGISTER 22-16: ADCHIT0H: A/D SCAN COMPARE HIT REGISTER 0 HIGH (HIGH WORD)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8
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 **CHH<15:8>:** A/D Compare Hit bits

If CM<1:0> = 11:

1 = A/D Result Buffer n has been written with data or a match has occurred

0 = A/D Result Buffer n has not been written with data

For all other values of CM<1:0>:

1 = A match has occurred on A/D Result Channel n

0 = No match has occurred on A/D Result Channel n

REGISTER 22-17: ADCHIT0L: A/D SCAN COMPARE HIT REGISTER 0 LOW (LOW WORD)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0
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 **CHH<7:0>:** A/D Compare Hit bits

If CM<1:0> = 11:

1 = A/D Result Buffer n has been written with data or a match has occurred

0 = A/D Result Buffer n has not been written with data

For all other values of CM<1:0>:

1 = A match has occurred on A/D Result Channel n

0 = No match has occurred on A/D Result Channel n

27.5.1.1 Bus Activity Detect Interrupt Bit (ACTVIF)

The ACTVIF bit cannot be cleared immediately after the USB module wakes up from Suspend mode or while the USB module is suspended. A few clock cycles are required to synchronize the internal hardware state machine before the ACTVIF bit can be cleared by firmware. Clearing the ACTVIF bit before the internal hardware is synchronized may not have an effect on the value of ACTVIF. Additionally, if the USB module uses the clock from the 96 MHz PLL source, then after clearing the SUSPND bit, the USB module may not be immediately operational while waiting for the 96 MHz PLL to lock. The application code should clear the ACTVIF flag as provided in Example 27-1.

Note:	Only one ACTVIF interrupt is generated when resuming from the USB bus Idle condition. If user firmware clears the ACTVIF bit, the bit will not immediately become set again, even when there is continuous bus traffic. Bus traffic must cease long enough to generate another IDLEIF condition before another ACTVIF interrupt can be generated.

EXAMPLE 27-1: CLEARING ACTVIF BIT (UIR<2>)

Assembly:

```
    BCF     UCON, SUSPND
LOOP:
    BTFSS   UIR, ACTVIF
    BRA     DONE
    BCF     UIR, ACTVIF
    BRA     LOOP
DONE:
```

C:

```
UCONbits.SUSPND = 0;
while (UIRbits.ACTVIF) { UIRbits.ACTVIF = 0; }
```

REGISTER 28-13: CONFIG8L: CONFIGURATION REGISTER 8 LOW (BYTE ADDRESS 30000Eh)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-1	U-1	U-1
DSWDTPS4	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	—	—	—
bit 7							bit 0

Legend:	P = Programmable bit	WO = Write-Once bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-3 **DSWDTPS<4:0>:** Deep Sleep Watchdog Timer Postscale Select bits
The DS WDT prescaler is 32; this creates an approximate base time unit of 1 ms.
11111 = 1:2^36 (25.7 days)
11110 = 1:2^35 (12.8 days)
11101 = 1:2^34 (6.4 days)
11100 = 1:2^33 (77.0 hours)
11011 = 1:2^32 (38.5 hours)
11010 = 1:2^31 (19.2 hours)
11001 = 1:2^30 (9.6 hours)
11000 = 1:2^29 (4.8 hours)
10111 = 1:2^28 (2.4 hours)
10110 = 1:2^27 (72.2 minutes)
10101 = 1:2^26 (36.1 minutes)
10100 = 1:2^25 (18.0 minutes)
10011 = 1:2^24 (9.0 minutes)
10010 = 1:2^23 (4.5 minutes)
10001 = 1:2^22 (135.3s)
10000 = 1:2^21 (67.7s)
01111 = 1:2^20 (33.825s)
01110 = 1:2^19 (16.912s)
01101 = 1:2^18 (8.456s)
01100 = 1:2^17 (4.228s)
01011 = 1:65536 (2.114s)
01010 = 1:32768 (1.057s)
01001 = 1:16384 (528.5 ms)
01000 = 1:8192 (264.3 ms)
00111 = 1:4096 (132.1 ms)
00110 = 1:2048 (66.1 ms)
00101 = 1:1024 (33 ms)
00100 = 1:512 (16.5 ms)
00011 = 1:256 (8.3 ms)
00010 = 1:128 (4.1 ms)
00001 = 1:64 (2.1 ms)
00000 = 1:32 (1 ms)

bit 2-0 **Unimplemented:** Read as '1'

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

Syntax: BTFSC f, b {,a}

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

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

Status Affected: None

Encoding:

1011	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction.

If 'a' is '0', the Access Bank 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	No operation

If skip:

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

If skip and followed by 2-word instruction:

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

Example:
```
HERE    BTFSC   FLAG, 1, 0
FALSE   :
TRUE    :
```

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

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

Syntax: BTFSS f, b {,a}

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

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

Status Affected: None

Encoding:

1010	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction.

If 'a' is '0', the Access Bank 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	No operation

If skip:

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

If skip and followed by 2-word instruction:

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

Example:
```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

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

RETFIE **Return from Interrupt**

Syntax: RETFIE {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
1 → GIE/GIEH or PEIE/GIEL;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: GIE/GIEH, PEIE/GIEL.

Encoding:

0000	0000	0001	000s

Description: Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low-priority Global Interrupt Enable bit. If 's' = 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	POP PC from stack Set GIEH or GIEL
No operation	No operation	No operation	No operation

Example: RETFIE 1

After Interrupt
PC = TOS
W = WS
BSR = BSRS
STATUS = STATUSS
GIE/GIEH, PEIE/GIEL = 1

RETLW **Return Literal to W**

Syntax: RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W,
(TOS) → PC,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	1100	kkkk	kkkk

Description: W is loaded with the 8-bit literal 'k'. The Program Counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	POP PC from stack, write to W
No operation	No operation	No operation	No operation

Example:

```
    CALL TABLE ; W contains table
               ; offset value
               ; W now has
               ; table value
    :
TABLE
    ADDWF PCL  ; W = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
    :
    :
    RETLW kn   ; End of table
```

Before Instruction
W = 07h
After Instruction
W = value of kn

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	X	/XX	XXX
Device	Tape and Reel Option	Temperature Range	Package	Pattern

Device:	PIC18F97J94, PIC18F96J94, PIC18F95J94, PIC18F87J94, PIC18F86J94, PIC18F85J94, PIC18F67J94, PIC18F66J94, PIC18F65J94 VDD range 2.0 to 3.6V
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel(1)
Temperature Range:	I = -40°C to +85°C (Industrial)
Package:	PT = TQFP (Thin Quad Flatpack) PF = TQFP (100-Pin Thin Quad, 14x14x1 Body)
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC18F97J94-I/PT = Industrial temp., TQFP package, QTP pattern #301.

b) PIC18F87J94-I/PT = Industrial temp., TQFP package.

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.