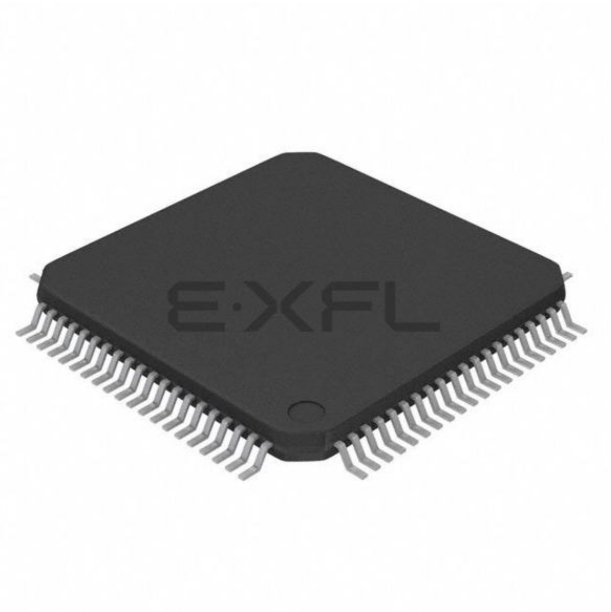
Microchip Technology - PIC18F86J94T-I/PT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	64KB (32K 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	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f86j94t-i-pt

TABLE 2: 64-PIN ALLOCATION TABLE (PIC18F6XJ94) (CONTINUED)

I/O	64-Pin TQFP/QFN	ADC	Comparator	HLVD	CTMU	USB	LCD	MSSP	PSP	Interrupt	REFO	PPS-Lite[1]	Pull-up	Basic
RE6	60	—	—	—	—	—	COM3	—	—	—	—	RP34	Y	—
RE7	59	—	—	—	—	—	LCDBIAS0	—	—	—	—	RP31	Y	—
RF2	16	AN7	C2INB	—	CTMUI	—	SEG20	—	—	—	—	RP36	Y	—
RF3	15	—	—	—	—	D-	—	—	—	—	—	—	Y	—
RF4	14	—	—	—	—	D+	—	—	—	—	—	—	Y	—
RF5	13	AN10	C1INB/ CVREF	—	—	—	SEG23	—	—	—	—	RP35	Y	—
RF6	12	AN11	C1INA	—	—	—	SEG24	—	—	—	—	RP40	Y	—
RF7	11	AN5	—	—	—	—	SEG25	—	—	—	—	RP38	Y	—
RG0	3	AN8	—	—	—	—	COM4/ SEG28	—	—	—	—	RP46	Y	—
RG1	4	AN19	—	—	—	—	COM5/ SEG29	—	—	—	—	RP39	Y	—
RG2	5	AN18	C3INA	—	—	—	COM6/ SEG30	—	—	—	—	RP42	Y	—
RG3	6	AN17	C3INB	—	—	—	COM7/ SEG31	—	—	—	—	RP43	Y	—
RG4	8	AN16	C3INC	—	—	—	SEG26	—	—	—	—	RP44	Y	—
RG5/$\overline{\text{MCLR}}$	7	—	—	—	—	—	—	—	—	—	—	—	Y	$\overline{\text{MCLR}}$
AVDD	19	AVDD	—	—	—	—	—	—	—	—	—	—	—	—
AVSS	20	AVSS	—	—	—	—	—	—	—	—	—	—	—	—
VBAT	18	—	—	—	—	—	—	—	—	—	—	—	—	VBAT
VCAP/ VDDCORE	10	—	—	—	—	—	—	—	—	—	—	—	—	VCAP/ VDDCORE
VDD	26, 38, 57	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	9, 25, 41, 56	—	—	—	—	—	—	—	—	—	—	—	—	VSS
VUSB3V3	17	—	—	—	—	—	—	—	—	—	—	—	—	VUSB3V3

Note 1: The peripheral inputs and outputs that support PPS have no default pins.

TABLE 1-4: PIC18FXXJ94 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	100	80	64			
SEG19/AN0/AN1-/RP0/RA0	37	30	24			
SEG19				O	Analog	SEG19 output for LCD.
AN0				I	Analog	Analog Input 0.
AN1-				I	Analog	A/D negative input channel.
RP0				I/O	ST/DIG	Remappable Peripheral Pin 0 input/output.
RA0				I/O	ST/DIG	General purpose I/O pin.
SEG18/AN1/RP1/RA1	36	29	23			
SEG18				O	Analog	SEG18 output for LCD.
AN1				I	Analog	Analog Input 1.
RP1				I/O	ST/DIG	Remappable Peripheral Pin 1 input/output.
RA1				I/O	ST/DIG	General purpose I/O pin.
SEG21/VREF-/AN2/RP2/RA2	34	28	22			
SEG21				O	Analog	SEG21 output for LCD.
VREF-				I	Analog	A/D reference voltage (low) input.
AN2				I	Analog	Analog Input 2.
RP2				I/O	ST/DIG	Remappable Peripheral Pin 2 input/output.
RA2				I/O	ST/DIG	General purpose I/O pin.
VREF+/AN3/RP3/RA3	33	27	21			
VREF+				I	Analog	A/D reference voltage (high) input.
AN3				I	Analog	Analog Input 3.
RP3				I/O	ST/DIG	Remappable Peripheral Pin 3 input/output.
RA3				I/O	ST/DIG	General purpose I/O pin.
SEG14/AN6/RP4/RA4	43	34	28			
SEG14				O	Analog	SEG14 output for LCD.
AN6				I	Analog	Analog Input 6.
RP4				I/O	ST/DIG	Remappable Peripheral Pin 4 input/output.
RA4				I/O	ST/DIG	General purpose I/O pin.
SEG15/AN4/LVDIN/C1INA/ C2INA/C3INA/RP5/RA5	42	33	27			
SEG15				O	Analog	SEG15 output for LCD.
AN4				I	Analog	Analog Input 4.
LVDIN				I	Analog	High/Low-Voltage Detect (HLVD) input.
C1INA				I	Analog	Comparator 1 Input A.
C2INA				I	Analog	Comparator 2 Input A.
C3INA				I	Analog	Comparator 3 Input A.
RP5				I/O	ST/DIG	Remappable Peripheral Pin 5 input/output.
RA5				I/O	ST/DIG	General purpose I/O pin.

Legend:	TTL	= TTL compatible input	CMOS	= CMOS compatible input or output
	ST	= Schmitt Trigger input with CMOS levels	Analog	= Analog input
	I	= Input	O	= Output
	P	= Power	OD	= Open-Drain (no P diode to VDD)
	I^2C	= I^2C/SMBus		

TABLE 1-4: PIC18FXXJ94 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	100	80	64			
VLCAP1/RP8/CTED13/INT0/RB0	73	58	48			
VLCAP1				I	Analog	LCD Drive Charge Pump Capacitor Input 1.
RP8				I/O	ST/DIG	Remappable Peripheral Pin 8 input/output.
CTED13				I	ST	CTMU Edge 13 input.
INT0				I	ST	External Interrupt 0.
RB0				I/O	ST/DIG	General purpose I/O pin.
VLCAP2/RP9/RB1	72	57	47			
VLCAP2				I	Analog	LCD Drive Charge Pump Capacitor Input 2.
RP9				I/O	ST/DIG	Remappable Peripheral Pin 9 input/output.
RB1				I/O	ST/DIG	General purpose I/O pin.
SEG9/RP14/CTED1/RB2	70	56	46			
SEG9				O	Analog	SEG9 output for LCD.
RP14				I/O	ST/DIG	Remappable Peripheral Pin 14 input/output.
CTED1				I	ST	CTMU Edge 1 input.
RB2				I/O	ST/DIG	General purpose I/O pin.
SEG10/RP7/CTED2/RB3	69	55	45			
SEG10				O	Analog	SEG10 output for LCD.
RP7				I/O	ST/DIG	Remappable Peripheral Pin 7 input/output.
CTED2				I	ST	CTMU Edge 2 input.
RB3				I/O	ST/DIG	General purpose I/O pin.
SEG11/RP12/CTED3/RB4	68	54	44			
SEG11				O	Analog	SEG11 output for LCD.
RP12				I/O	ST/DIG	Remappable Peripheral Pin 12 input/output.
CTED3				I	ST	CTMU Edge 3 input.
RB4				I/O	ST/DIG	General purpose I/O pin.
SEG8/RP13/CTED4/RB5	67	53	43			
SEG8				O	Analog	SEG8 output for LCD.
RP13				I/O	ST/DIG	Remappable Peripheral Pin 13 input/output.
CTED4				I	ST	CTMU Edge 4 input.
RB5				I/O	ST/DIG	General purpose I/O pin.
PGC/CTED5/RB6	65	52	42			
PGC				I/O	ST/DIG	In-Circuit Debugger and ICSP™ programming clock pin.
CTED5				I	ST	CTMU Edge Input.
RB6				I/O	ST/DIG	General purpose I/O pin.
PGD/CTED6/RB7	58	47	37			
PGD				I/O	ST/DIG	In-Circuit Debugger and ICSP™ programming data pin.
CTED6				I	ST	CTMU Edge 6 input.
RB7				I/O	ST/DIG	General purpose I/O pin.

Legend:	TTL = TTL compatible input	CMOS = CMOS compatible input or output
	ST = Schmitt Trigger input with CMOS levels	Analog = Analog input
	I = Input	O = Output
	P = Power	OD = Open-Drain (no P diode to VDD)
	I^2C = I^2C/SMBus	

4.0 POWER-MANAGED MODES

All PIC18F97J94 Family devices offer a number of built-in strategies for reducing power consumption. These strategies can be particularly useful in applications, which are both power-constrained (such as battery operation), yet require periods of full-power operation for timing-sensitive routines (such as serial communications).

Aside from their low-power architecture, these devices include an expanded range of dedicated hardware features that allow the microcontroller to reduce power consumption to even lower levels when long-term hibernation is required, and still be able to resume operation on short notice.

The device has four power-saving features:

- Instruction-Based Power-Saving Modes
- Hardware-Based Power Reduction Features
- Microcontroller Clock Manipulation
- Selective Peripheral Control

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical or timing-sensitive application features. However, it is more convenient to discuss the strategies separately.

4.1 Overview of Power-Saving Modes

In addition to full-power operation, otherwise known as Run mode, PIC18F97J94 Family devices offer three instruction-based, power-saving modes and one hardware-based mode. In descending order of power consumption, they are:

- Idle
- Sleep (including retention Sleep)
- Deep Sleep (with and without retention)
- VBAT (with and without RTCC)

By powering down all four modes, different functional areas of the microcontroller allow progressive reductions of operating and Idle power consumption. In addition, three of the modes can be tailored for more power reduction at a trade-off of some operating features. Table 4-1 lists all of the operating modes (including Run mode, for comparison) in order of increasing power savings and summarizes how the microcontroller exits the different modes.

4.4.10 CONTROL BIT SUMMARY FOR SLEEP MODES

Table 4-5 shows the settings for the bits relevant to Deep Sleep modes.

TABLE 4-5: BIT SETTINGS FOR ALL DEEP SLEEP MODES

Instruction-Based Mode	DSEN (DSCONH<7>)	Retention Regulator			DSWDTEN (CONFIG8H<0>)
		$\overline{\text{RETEN}}$ (CONFIG7L<0>)	SRETEN (RCON4<4>)	State	
Retention Deep Sleep	1	0	1	Enabled	0
Deep Sleep	1	1	x	Disabled	x

4.4.11 WAKE-UP DELAYS

The Reset delays associated with wake-up from Deep Sleep and Retention Deep Sleep modes, in different oscillator modes, are provided in Table 4-6 and Table 4-7, respectively.

Note: The PMSLP bit (RCON4<0>) allows the voltage regulator to be maintained during Sleep modes.

TABLE 4-6: DELAY TIMES FOR EXITING FROM DEEP SLEEP MODE

Clock Source		Exit Delay	Oscillator Delay	Notes
EC		TDSWU	—	
ECPLL		TDSWU	TLOCK	**1, 3**
MS, HS		TDSWU	TOST	**1, 2**
MSPLL, HSPLL		TDSWU	TOST + TLOCK	**1, 2, 3**
SOSC	(Off during Sleep)	TDSWU	TOST	**1, 2**
	(On during Sleep)	TDSWU	—	**1**
FRC, FRCDIV		TDSWU	TFRC	**1, 4**
LPRC	(Off during Sleep)	TDSWU	TLPRC	**1, 4**
	(On during Sleep)	TDSWU	—	**1**
FRCPLL		TDSWU	TFRC + TLOCK	**1, 3, 4**

Note 1: TDSWU = Deep Sleep wake-up delay.

2: TOST = Oscillator Start-up Timer; a delay of 1024 oscillator periods before the oscillator clock is released to the system.

3: TLOCK = PLL lock time.

4: TFRC and TLPRC are RC Oscillator start-up times.

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).

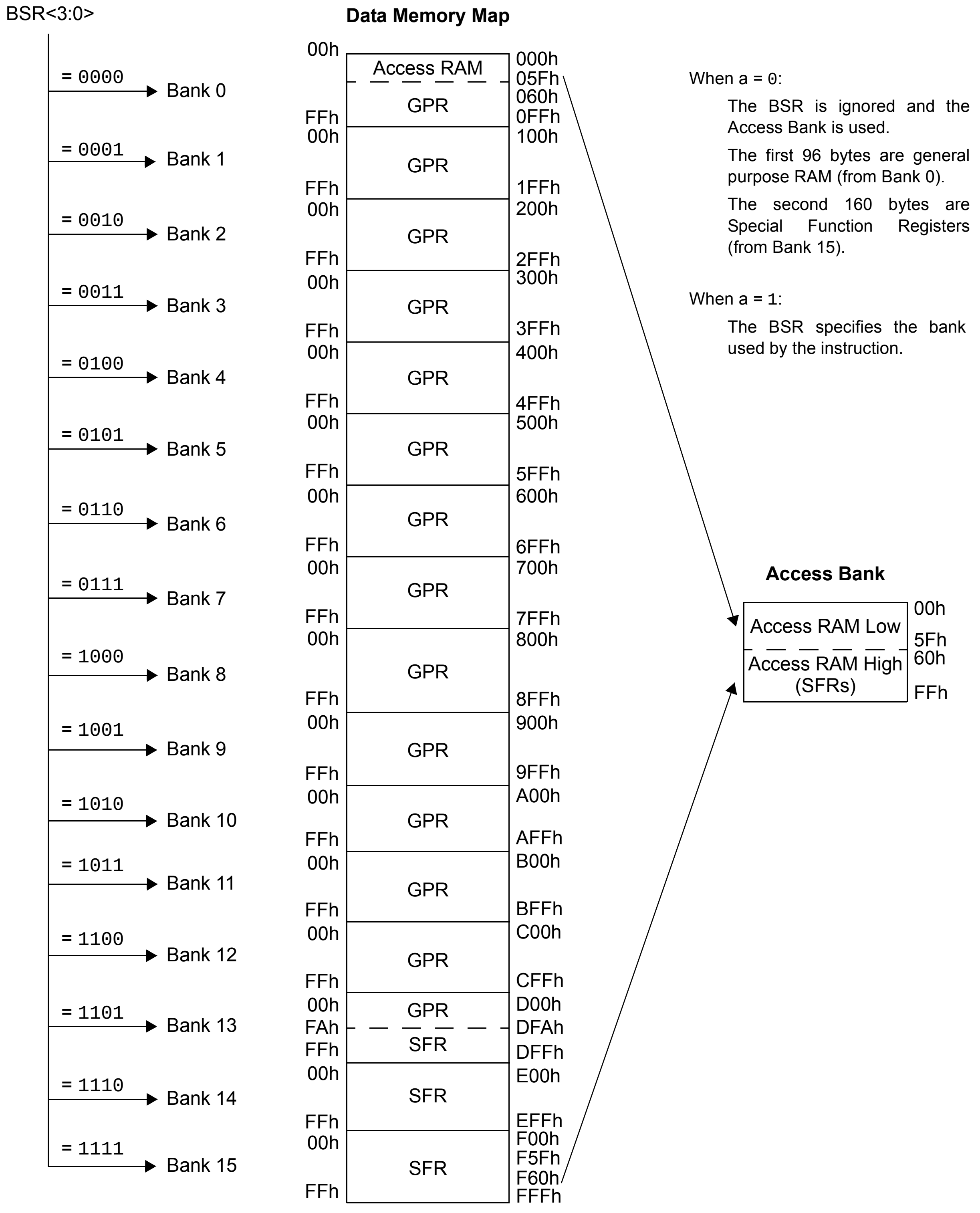
3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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 VBAT is always powered, the DSGPx register values will remain unchanged after the first POR.

FIGURE 6-6: DATA MEMORY MAP FOR PIC18F97J94 FAMILY DEVICES



When a = 0:

The BSR is ignored and the Access Bank is used.

The first 96 bytes are general purpose RAM (from Bank 0).

The second 160 bytes are Special Function Registers (from Bank 15).

When a = 1:

The BSR specifies the bank used by the instruction.

Note 1: Addresses, DFAh through F5Fh, are also SFRs, but are not part of the Access RAM. Users must always use the complete address, or load the proper BSR value, to access these registers.

REGISTER 10-18: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR5GIP	LCDIP	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP
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 **TMR5GIP:** TMR5 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **LCDIP:** LCD Ready Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RC2IP:** EUSART2 Receive Priority Flag bit
1 = High priority
0 = Low priority

bit 4 **TX2IP:** EUSART2 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

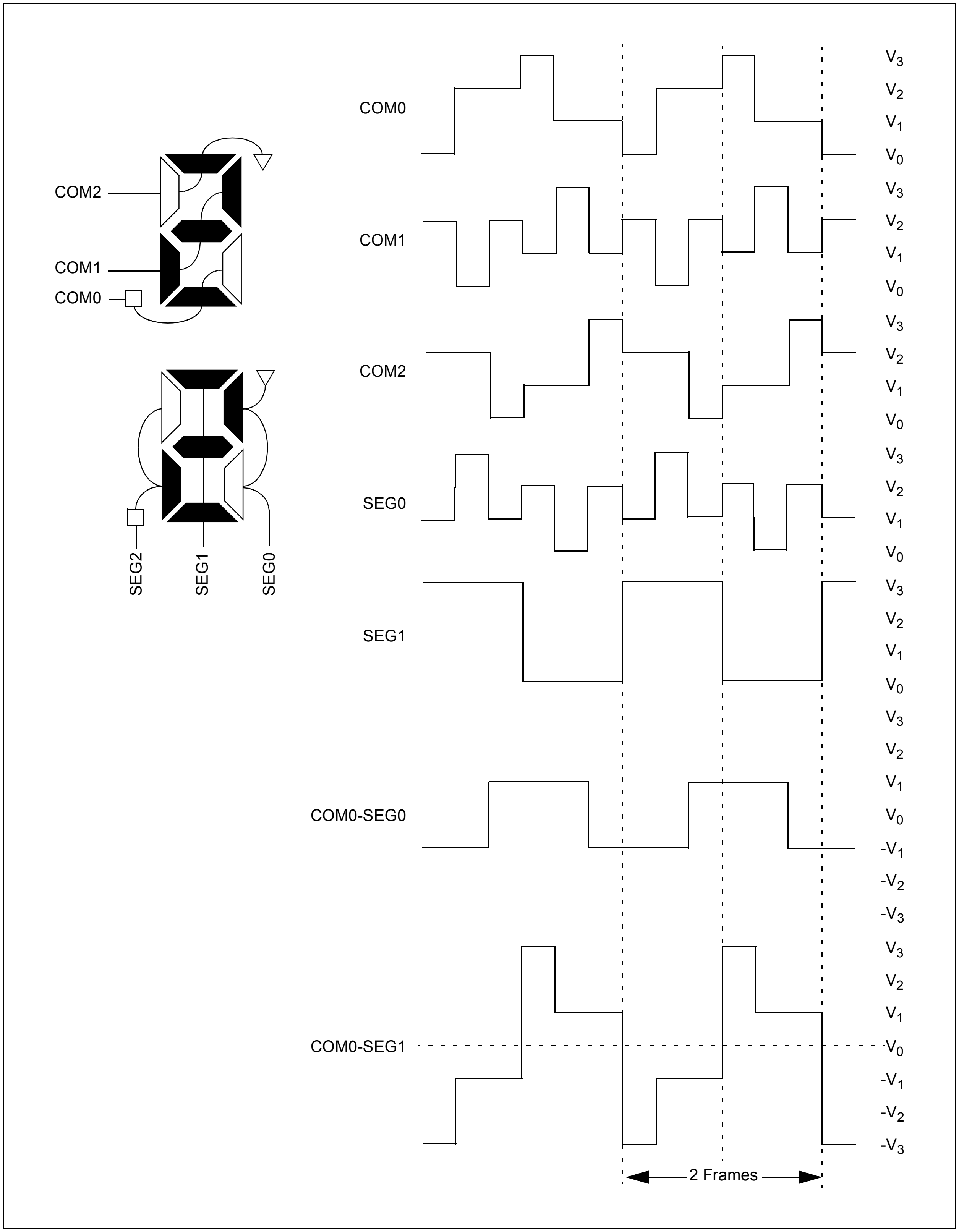
bit 3 **CTMUIP:** CTMU Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP2IP:** CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP1IP:** ECCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RTCCIP:** RTCC Interrupt Priority bit
1 = High priority
0 = Low priority

FIGURE 13-17: TYPE-B WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE



17.6 Register Maps

Table 17-5, Table 17-6 and Table 17-7 summarize the registers associated with the RTCC module.

TABLE 17-5: RTCC CONTROL REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTCCON1	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0
RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
RTCCON2	PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLKSEL1	RTCCLKSEL0	RTCSECSEL1	RTCSECSEL
ALRMCFG	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
ALRMRPT	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
PMD3	DSMMD	CTMUMD	ADCMD	RTCCMD	LCDMD	PSPMD	REFO1MD	REFO2MD

Legend: — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for 80-pin devices.

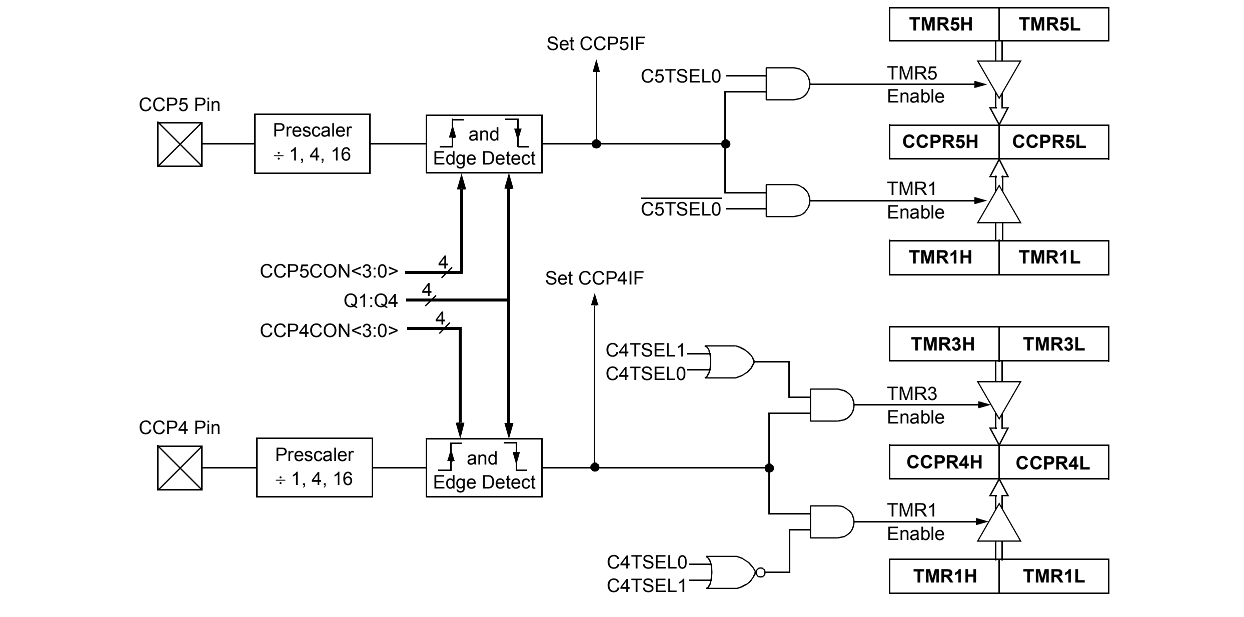
TABLE 17-6: RTCC VALUE REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTCVALH	RTCC Value High Register Window based on RTCPTR<1:0>							
RTCVALL	RTCC Value Low Register Window based on RTCPTR<1:0>							

TABLE 17-7: ALARM VALUE REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ALRMVALH	Alarm Value High Register Window based on ALRMPTR<1:0>							
ALRMVALL	Alarm Value Low Register Window based on ALRMPTR<1:0>							

FIGURE 19-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



Note: This block diagram uses CCP4 and CCP5, and their appropriate timers as an example. For details on all of the CCP modules and their timer assignments, see Table 19-2 and Table 19-3.

19.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP4IE bit (PIE4<1>) clear to avoid false interrupts and should clear the flag bit, CCP4IF, following any such change in operating mode.

19.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCP4M<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Doing that will also not clear the prescaler counter – meaning the first capture may be from a non-zero prescaler.

Example 19-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 19-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF  CCP4CON     ; Turn CCP module off
MOVLW NEW_CAPT_PS ; Load WREG with the
                  ; new prescaler mode
                  ; value and CCP ON
MOVWF CCP4CON     ; Load CCP4CON with
                  ; this value
```

REGISTER 20-6: SSPxSTAT: MSSPx STATUS REGISTER (I^2C MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{\text{A}}$	P[1]	S[1]	R/$\overline{\text{W}}$[2,3]	UA	BF
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 **SMP:** Slew Rate Control bit

In Master or Slave mode:
1 = Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

In Master or Slave mode:
1 = Enables SMBus-specific inputs
0 = Disables SMBus-specific inputs

bit 5 **D/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ bit

In Master mode:
Reserved.

In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit[1]

1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

bit 3 **S:** Start bit[1]

1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last

bit 2 **R/$\overline{\text{W}}$:** Read/$\overline{\text{Write}}$ Information bit[2,3]

In Slave mode:
1 = Read
0 = Write

In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-Bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:
1 = SSPxBUF is full
0 = SSPxBUF is empty

In Receive mode:
1 = SSPxBUF is full (does not include the $\overline{\text{ACK}}$ and Stop bits)
0 = SSPxBUF is empty (does not include the $\overline{\text{ACK}}$ and Stop bits)

Note 1: This bit is cleared on Reset and when SSPEN is cleared.

2: This bit holds the R/$\overline{\text{W}}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{\text{ACK}}$ bit.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

21.2 EUSARTx Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTAx<4>). In this mode, the EUSARTx uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated 8-bit/16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSARTx transmits and receives the LSb first. The EUSARTx's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH and BRG16 bits (TXSTAx<2> and BAUDCONx<3>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

When operating in Asynchronous mode, the EUSARTx module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-Bit Break Character Transmit
- Auto-Baud Rate Detection

21.2.1 EUSARTx ASYNCHRONOUS TRANSMITTER

The EUSARTx transmitter block diagram is shown in Figure 21-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREGx register (if available).

Once the TXREGx register transfers the data to the TSR register (occurs in one TCY), the TXREGx register is empty and the TXxIF flag bit is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxIE. TXxIF will be set regardless of the state of TXxIE; it cannot be cleared in software. TXxIF is also not cleared immediately upon loading TXREGx, but becomes valid in the second instruction cycle following the load instruction. Polling TXxIF immediately following a load of TXREGx will return invalid results.

While TXxIF indicates the status of the TXREGx register, another bit, TRMT (TXSTAx<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

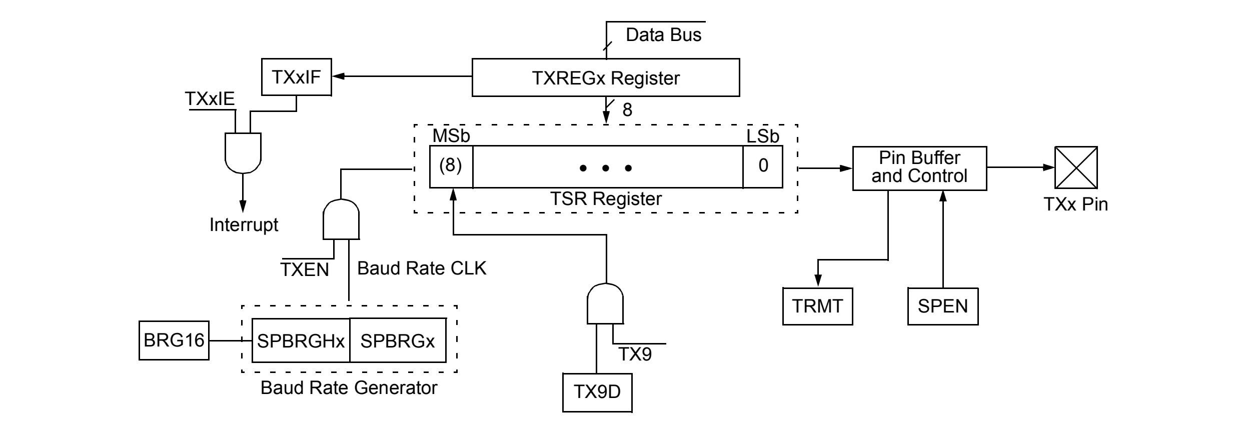
Note 1: The TSR register is not mapped in data memory, so it is not available to the user.

2: Flag bit, TXxIF, is set when enable bit, TXEN, is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, TXxIE.
4. If 9-bit transmission is desired, set transmit bit, TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TXxIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Load data to the TXREGx register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 21-3: EUSARTx TRANSMIT BLOCK DIAGRAM



REGISTER 22-11: ADCON5L: A/D CONTROL REGISTER 5 LOW

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	CM1	CM0
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 7-4 **Unimplemented:** Read as '0'

bit 3-2 **WM<1:0>:** Write Mode bits

11 = Reserved
10 = Auto-compare only (conversion results are not saved, but interrupts are generated when a valid match occurs, as defined by the CM<1:0> and ASINTMD<1:0> bits)
01 = Convert and save (conversion results are saved to locations as determined by the register bits when a match occurs, as defined by the CMx bits)
00 = Legacy operation (conversion data is saved to a location determined by the buffer register bits)

bit 1-0 **CM<1:0>:** Compare Mode bits

11 = Outside Window mode (valid match occurs if the conversion result is outside of the window, defined by the corresponding buffer pair)
10 = Inside Window mode (valid match occurs if the conversion result is inside the window, defined by the corresponding buffer pair)
01 = Greater Than mode (valid match occurs if the result is greater than the value in the corresponding buffer register)
00 = Less Than mode (valid match occurs if the result is less than the value in the corresponding buffer register)

26.1 CTMU Registers

The control registers for the CTMU are:

- CTMUCON1
- CTMUCON2
- CTMUCON3
- CTMUCON4

The CTMUCON1 and CTMUCON3 registers (Register 26-1 and Register 26-3) contain control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUCON2 register (Register 26-2) has bits for selecting the current source range and current source trim.

REGISTER 26-1: CTMUCON1: CTMU CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 4 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 3 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 2 **ESGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 1 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 0 **CTTRIG:** CTMU Special Event Trigger bit
1 = CTMU Special Event Trigger is enabled
0 = CTMU Special Event Trigger is disabled

EXAMPLE 26-3: CTMU CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 25                         //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                  //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                     //for unsigned conversion 10 sig bits
#define ADREF 3.3                        //Vdd connected to A/D Vr+
#define RCAL .027                        //R value is 4200000 (4.2M)
                                         //scaled so that result is in
                                         //1/100th of uA


int main(void)
{
    int i;
    int j = 0;                           //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

//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
    for(j=0;j<10;j++)
    {
        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
        PIR1bits.ADIF = 0;               //Clear A/D Interrupt Flag
        VTot += Vread;                   //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);         //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

REGISTER 28-8: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-1	R/WO-1	R/WO-1
WAIT(1)	BW	ABW1	ABW0	EASHFT	—	CINASEL	T5GSEL
bit 7							bit 0

Legend:	P = Programmable bit	WO = Write-Once bit	
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 **WAIT:** External Bus Wait Enable bit(1)

1 = Wait selections from WAIT<1:0> (MEMCON<5:4>) are unavailable and the device will not wait
0 = Wait is programmed by WAIT<1:0> (MEMCON<5:4>)

bit 6 **BW:** Data Bus Width Select bit

1 = 16-Bit External Bus mode
0 = 8-Bit External Bus mode

bit 5-4 **ABW<1:0>:** External Memory Bus Configuration bits

00 =Extended Microcontroller Mode – 20-Bit Address mode
01 =Extended Microcontroller Mode – 16-Bit Address mode
10 =Extended Microcontroller Mode – 12-Bit Address mode
11 =Microcontroller Mode – External bus is disabled

bit 3 **EASHFT:** External Address Bus Shift Enable bit

1 = Address shifting is enabled – External address bus is shifted to start at 000000h
0 = Address shifting is disabled – External address bus reflects the PC value

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **CINASEL:** CxINA Gate Select bit

1 = C1INA and C3INA are on their default pin locations
0 = C1INA and C3INA are all remapped to pin, RA5

bit 0 **T5GSEL:** TMR5 Gate Select bit

1 = TMR5 gate is driven by the T5G input
0 = TMR5 gate is driven by the T3G input

Note 1: This bit was previously referred to as ‘WAIT’, but a set condition actually indicates the case where the EMB does not wait and the name was therefore changed to reflect this. No change in functionality or polarity occurred, only a change in the name of the register bit.

CLRF — Clear f

Syntax: CLRF f {,a}

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

Operation: 000h → f,
1 → Z

Status Affected: Z

Encoding:

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

Description: Clears the contents of the specified register.

If ‘a’ is ‘`0`’, the Access Bank 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: `CLRF FLAG_REG,1`

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT — Clear Watchdog Timer

Syntax: CLRWDT

Operands: None

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

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

Encoding:

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

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: `CLRWDT`

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
$\overline{\text{TO}}$ = 1
$\overline{\text{PD}}$ = 1

NEGF — Negate f

Syntax: NEGF f {,a}

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

Operation: $(\overline{f}) + 1 \rightarrow f$

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

Encoding:

`0110`	`110a`	`ffff`	`ffff`

Description: Location ‘f’ is negated using two’s complement. The result is placed in the 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 $f \leq 95$ (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: `NEGF REG, 1`

```
Before Instruction
    REG    =   0011 1010 [3Ah]
After Instruction
    REG    =   1100 0110 [C6h]
```

NOP — No Operation

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

`0000`
`1111`
	`0000`
	`xxxx`
		`0000`
		`xxxx`
			`0000`
			`xxxx`

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

29.2.2 EXTENDED INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \leq k \leq 63$
$f \in [\, 0, 1, 2 \,]$

Operation: $\mathrm{FSR}(f) + k \rightarrow \mathrm{FSR}(f)$

Status Affected: None

Encoding:

`1110`	`1000`	`ffkk`	`kkkk`

Description: The 6-bit literal ‘k’ is added to the contents of the FSR specified by ‘f’.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ADDFSR 2, 23h`

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDULNK **Add Literal to FSR2 and Return**

Syntax: ADDULNK k

Operands: $0 \leq k \leq 63$

Operation: $\mathrm{FSR2} + k \rightarrow \mathrm{FSR2}$,
$(\mathrm{TOS}) \rightarrow \mathrm{PC}$

Status Affected: None

Encoding:

`1110`	`1000`	`11kk`	`kkkk`

Description: The 6-bit literal ‘k’ is added to the contents of FSR2. A `RETURN` is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a `NOP` is performed during the second cycle.

This may be thought of as a special case of the `ADDFSR` instruction, where f = 3 (binary ‘`11`’); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: `ADDULNK 23h`

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).