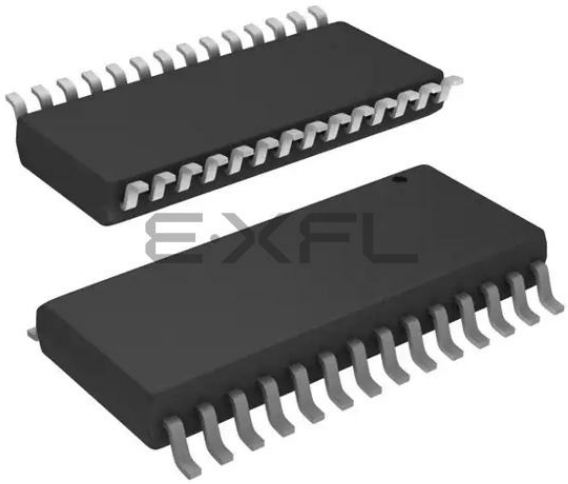


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 2.75V
Data Converters	A/D 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26j13t-i-so

4.6.9 DEEP SLEEP MODE REGISTERS

Deep Sleep mode registers are provided in Register 4-1 through Register 4-6.

REGISTER 4-1: DSCONH: DEEP SLEEP CONTROL HIGH BYTE REGISTER (BANKED F4Dh)

R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DSEN(1)	—	—	—	—	r	DSULPEN	RTCWDIS
bit 7							bit 0

Legend:	r = Reserved 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 **DSEN:** Deep Sleep Enable bit(1)
- 1 = Deep Sleep mode is entered on a SLEEP command
- 0 = Sleep mode is entered on a SLEEP command

bit 6-3 **Unimplemented:** Read as '0'

bit 2 **Reserved:** Maintain as '0'

bit 1 **DSULPEN:** Ultra Low-Power Wake-up Module Enable bit
- 1 = ULPWU module is enabled in Deep Sleep
- 0 = ULPWU module is disabled in Deep Sleep

bit 0 **RTCWDIS:** RTCC Wake-up Disable bit
- 1 = Wake-up from RTCC is disabled
- 0 = Wake-up from RTCC is enabled

Note 1: In order to enter Deep Sleep, Sleep must be executed within 2 instruction cycles after setting DSEN.

REGISTER 4-2: DSCONL: DEEP SLEEP LOW BYTE CONTROL REGISTER (BANKED F4Ch)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0(1)	R/W-0(1)
—	—	—	—	—	ULPWDIS	DSBOR	RELEASE
bit 7							bit 0

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

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **ULPWDIS:** Ultra Low-Power Wake-up Disable bit
- 1 = ULPWU wake-up source is disabled
- 0 = ULPWU wake-up source is enabled (must also set DSULPEN = 1)

bit 1 **DSBOR:** Deep Sleep BOR Event Status bit
- 1 = DSBOREN was enabled and VDD dropped below the DSBOR arming voltage during Deep Sleep, but did not fall below VDSBOR
- 0 = DSBOREN was disabled or VDD did not drop below the DSBOR arming voltage during Deep Sleep

bit 0 **RELEASE:** I/O Pin State Release bit

Upon waking from Deep Sleep, the I/O pins maintain their previous states. Clearing this bit will release the I/O pins and allow their respective TRIS and LAT bits to control their states.

Note 1: This is the value when VDD is initially applied.

TABLE 6-4: REGISTER FILE SUMMARY (PIC18F47J13 FAMILY) (CONTINUED)

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
F07h	CCPR8L	Capture/Compare/PWM Register 8 Low Byte								xxxx xxxx
F06h	CCP8CON	—	—	DC8B1	DC8B0	CCP8M3	CCP8M2	CCP8M1	CCP8M0	--00 0000
F05h	CCPR9H	Capture/Compare/PWM Register 9 High Byte								xxxx xxxx
F04h	CCPR9L	Capture/Compare/PWM Register 9 Low Byte								xxxx xxxx
F03h	CCP9CON	—	—	DC9B1	DC9B0	CCP9M3	CCP9M2	CCP9M1	CCP9M0	--00 0000
F02h	CCPR10H	Capture/Compare/PWM Register 10 High Byte								xxxx xxxx
F01h	CCPR10L	Capture/Compare/PWM Register 10 Low Byte								xxxx xxxx
F00h	CCP10CON	—	—	DC10B1	DC10B0	CCP10M3	CCP10M2	CCP10M1	CCP10M0	--00 0000
EFFh	RPINR24	—	—	—	PWM Fault Input (FLT0) to Input Pin Mapping bits					---1 1111
EFEh	RPINR23	—	—	—	SPI2 Slave Select Input ($\overline{SS2}$) to Input Pin Mapping bits					---1 1111
EFDh	RPINR22	—	—	—	SPI2 Clock Input (SCK2) to Input Pin Mapping bits					---1 1111
EFCh	RPINR21	—	—	—	SPI2 Data Input (SDI2) to Input Pin Mapping bits					---1 1111
EFBh	—	—	—	—	—	—	—	—	—	(3)
EFAh	—	—	—	—	—	—	—	—	—	(3)
EF9h	—	—	—	—	—	—	—	—	—	(3)
EF8h	RPINR17	—	—	—	EUSART2 Clock Input (CK2) to Input Pin Mapping bits					---1 1111
EF7h	RPINR16	—	—	—	EUSART2 RX2/DT2 to Input Pin Mapping bits					---1 1111
EF6h	—	—	—	—	—	—	—	—	—	(3)
EF5h	—	—	—	—	—	—	—	—	—	(3)
EF4h	RPINR14	—	—	—	Timer5 Gate Input (T5G) to Input Pin Mapping bits					---1 1111
EF3h	RPINR13	—	—	—	Timer3 Gate Input (T3G) to Input Pin Mapping bits					---1 1111
EF2h	RPINR12	—	—	—	Timer1 Gate Input (T1G) to Input Pin Mapping bits					---1 1111
EF1h	—	—	—	—	—	—	—	—	—	(3)
EF0h	—	—	—	—	—	—	—	—	—	(3)
EEFh	—	—	—	—	—	—	—	—	—	(3)
EEEh	—	—	—	—	—	—	—	—	—	(3)
EEDh	—	—	—	—	—	—	—	—	—	(3)
EECh	—	—	—	—	—	—	—	—	—	(3)
EEBh	—	—	—	—	—	—	—	—	—	(3)
EEAh	RPINR9	—	—	—	ECCP3 Input Capture (IC3) to Input Pin Mapping bits					---1 1111
EE9h	RPINR8	—	—	—	ECCP2 Input Capture (IC2) to Input Pin Mapping bits					---1 1111
EE8h	RPINR7	—	—	—	ECCP1 Input Capture (IC1) to Input Pin Mapping bits					---1 1111
EE7h	RPINR15	—	—	—	Timer5 External Clock Input (T5CKI) to Input Pin Mapping bits					---1 1111
EE6h	RPINR6	—	—	—	Timer3 External Clock Input (T3CKI) to Input Pin Mapping bits					---1 1111
EE5h	—	—	—	—	—	—	—	—	—	(3)
EE4h	RPINR4	—	—	—	Timer0 External Clock Input (T0CKI) to Input Pin Mapping bits					---1 1111
EE3h	RPINR3	—	—	—	External Interrupt (INT3) to Input Pin Mapping bits					---1 1111
EE2h	RPINR2	—	—	—	External Interrupt (INT2) to Input Pin Mapping bits					---1 1111
EE1h	RPINR1	—	—	—	External Interrupt (INT1) to Input Pin Mapping bits					---1 1111
EE0h	—	—	—	—	—	—	—	—	—	(3)
EDFh	—	—	—	—	—	—	—	—	—	(3)
EDEh	—	—	—	—	—	—	—	—	—	(3)
EDDh	—	—	—	—	—	—	—	—	—	(3)
EDCh	—	—	—	—	—	—	—	—	—	(3)
EDBh	—	—	—	—	—	—	—	—	—	(3)
EDAh	—	—	—	—	—	—	—	—	—	(3)
ED9h	—	—	—	—	—	—	—	—	—	(3)
ED8h	RPOR24(2)	—	—	—	Remappable Pin RP24 Output Signal Select bits					---0 0000
ED7h	RPOR23(2)	—	—	—	Remappable Pin RP23 Output Signal Select bits					---0 0000

Note 1: Applicable for 28-pin devices (PIC18F26J13, PIC18F27J13, PIC18LF26J13 and PIC18LF27J13).
2: Applicable for 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).
3: Value on POR, BOR.

7.5.2 FLASH PROGRAM MEMORY WRITE SEQUENCE (WORD PROGRAMMING)

The PIC18F47J13 Family of devices has a feature that allows programming a single word (two bytes). This feature is enabled when the WPROG bit is set. If the memory location is already erased, the following sequence is required to enable this feature:

1. Load the Table Pointer register with the address of the data to be written. (It must be an even address.)
2. Write the 2 bytes into the holding registers by performing table writes. (Do not post-increment on the second table write.)
3. Set the WREN bit (EECON1<2>) to enable writes and the WPROG bit (EECON1<5>) to select Word Write mode.
4. Disable interrupts.
5. Write 55h to EECON2.
6. Write 0AAh to EECON2.
7. Set the WR bit; this will begin the write cycle.
8. The CPU will stall for the duration of the write for TIW (see parameter D133A).
9. Re-enable interrupts.

EXAMPLE 7-4: SINGLE-WORD WRITE TO FLASH PROGRAM MEMORY

```
                MOVLW   CODE_ADDR_UPPER         ; Load TBLPTR with the base address
                MOVWF   TBLPTRU
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW           ; The table pointer must be loaded with an even
                                                  address
                MOVWF   TBLPTRL

                MOVLW   DATA0                   ; LSB of word to be written
                MOVWF   TABLAT
                TBLWT*+
                MOVLW   DATA1                   ; MSB of word to be written
                MOVWF   TABLAT
                TBLWT*                          ; The last table write must not increment the table
                                                  pointer! The table pointer needs to point to the
                                                  MSB before starting the write operation.


PROGRAM_MEMORY
                BSF     EECON1, WPROG           ; enable single word write
                BSF     EECON1, WREN            ; enable write to memory
                BCF     INTCON, GIE             ; disable interrupts
                MOVLW   0x55
    Required    MOVWF   EECON2                  ; write 55h
    Sequence    MOVLW   0xAA
                MOVWF   EECON2                  ; write AAh
                BSF     EECON1, WR              ; start program (CPU stall)
                BSF     INTCON, GIE             ; re-enable interrupts
                BCF     EECON1, WPROG           ; disable single word write
                BCF     EECON1, WREN            ; disable write to memory
```

9.0 INTERRUPTS

Devices of the PIC18F47J13 Family have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are 19 registers, which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3, PIR4, PIR5
- PIE1, PIE2, PIE3, PIE4, PIE5
- IPR1, IPR2, IPR3, IPR4, IPR5

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address, 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit, which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit, which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine (ISR), the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

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

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

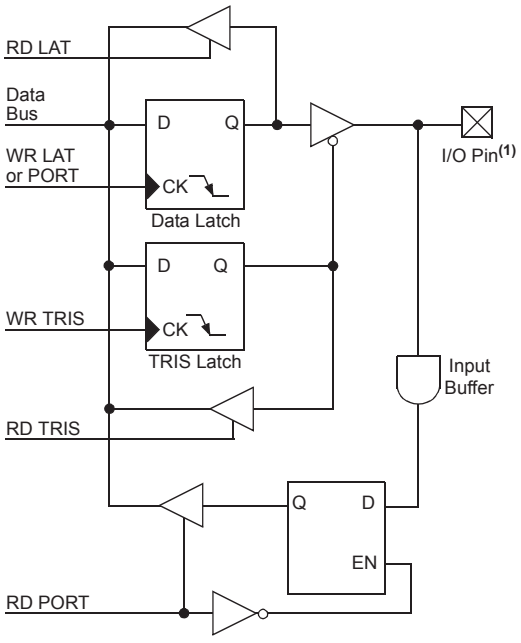
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Data Latch)

The Data Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

Figure 10-1 displays a simplified model of a generic I/O port, without the interfaces to other peripherals.

FIGURE 10-1: GENERIC I/O PORT OPERATION



Note 1: I/O pins have diode protection to VDD and VSS.

10.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

10.1.1 PIN OUTPUT DRIVE

The output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. PORTB and PORTC are designed to drive higher loads, such as LEDs. All other ports are designed for small loads, typically indication only. Table 10-1 summarizes the output capabilities. Refer to **Section 30.0 "Electrical Characteristics"** for more details.

TABLE 10-1: OUTPUT DRIVE LEVELS

Port	Drive	Description
PORTA	Minimum	Intended for indication.
PORTD		
PORTE		
PORTB	High	Suitable for direct LED drive levels.
PORTC		

10.1.2 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V; a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided. Table 10-2 summarizes the input capabilities. Refer to **Section 30.0 "Electrical Characteristics"** for more details.

TABLE 10-2: INPUT VOLTAGE LEVELS

Port or Pin	Tolerated Input	Description
PORTA<7:0>	VDD	Only VDD input levels are tolerated.
PORTB<3:0>		
PORTC<2:0>		
PORTE<2:0>		
PORTB<7:4>	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTC<7:3>		
PORTD<7:0>		

10.7.6 PERIPHERAL PIN SELECT REGISTERS

The PIC18F47J13 Family of devices implements a total of 37 registers for remappable peripheral configuration of 44-pin devices. The 28-pin devices have 31 registers for remappable peripheral configuration.

Note: Input and output register values can only be changed if IOLOCK (PPSCON<0>) = 0. See Example 10-7 for a specific command sequence.

REGISTER 10-5: PPSCON: PERIPHERAL PIN SELECT INPUT REGISTER 0 (BANKED EBFh)(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IOLOCK
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 7-1 **Unimplemented:** Read as '0'

bit 0 **IOLOCK:** I/O Lock Enable bit
1 = I/O lock is active, RPORx and RPINRx registers are write-protected
0 = I/O lock is not active, pin configurations can be changed

Note 1: Register values can only be changed if IOLOCK (PPSCON<0>) = 0.

REGISTER 10-6: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1 (BANKED EE1h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	INTR1R4	INTR1R3	INTR1R2	INTR1R1	INTR1R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 **INTR1R<4:0>:** Assign External Interrupt 1 (INT1) to the Corresponding RPn Pin bits

REGISTER 10-7: RPINR2: PERIPHERAL PIN SELECT INPUT REGISTER 2 (BANKED EE2h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	INTR2R4	INTR2R3	INTR2R2	INTR2R1	INTR2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 **INTR2R<4:0>:** Assign External Interrupt 2 (INT2) to the Corresponding RPn Pin bits

REGISTER 11-5: PMEH: PARALLEL PORT ENABLE REGISTER HIGH BYTE (BANKED F57h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
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 **PTEN<15:14>:** PMCS Port Enable bits
1 = PMA<15:14> function as either PMA<15:14> or PMCS
0 = PMA<15:14> function as port I/O

bit 5-0 **PTEN<13:8>:** PMP Address Port Enable bits
1 = PMA<13:8> function as PMP address lines
0 = PMA<13:8> function as port I/O

Note 1: This register is only available on 44-pin devices.

REGISTER 11-6: PMEL: PARALLEL PORT ENABLE REGISTER LOW BYTE (BANKED F56h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
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 7-2 **PTEN<7:2>:** PMP Address Port Enable bits
1 = PMA<7:2> function as PMP address lines
0 = PMA<7:2> function as port I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
1 = PMA<1:0> function as either PMA<1:0> or PMALH and PMALL
0 = PMA<1:0> pads functions as port I/O

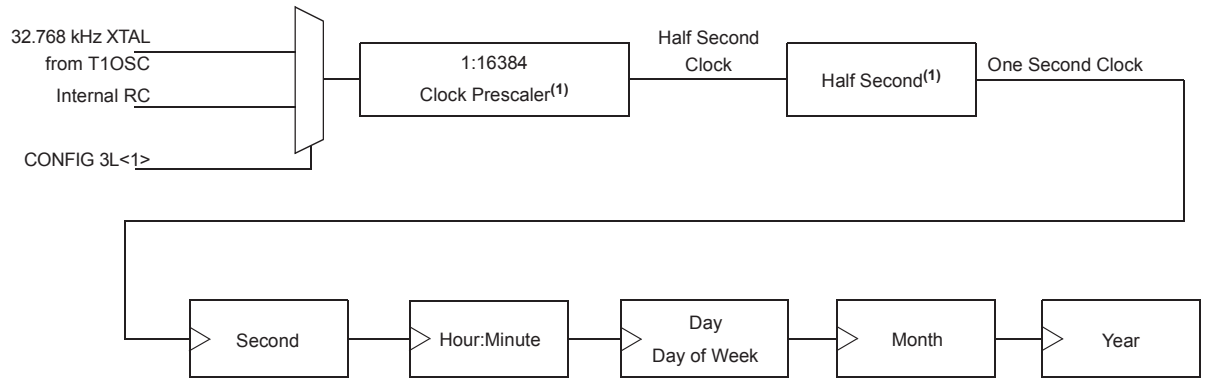
Note 1: This register is only available on 44-pin devices.

17.2.2 CLOCK SOURCE

As mentioned earlier, the RTCC module is intended to be clocked by an external Real-Time Clock (RTC) crystal oscillating at 32.768 kHz, but can also be clocked by the INTRC. The RTCC clock selection is decided by the RTCOSC bit (CONFIG3L<1>).

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

FIGURE 17-4: CLOCK SOURCE MULTIPLEXING



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

17.2.2.1 Real-Time Clock Enable

The RTCC module can be clocked by an external, 32.768 kHz crystal (Timer1 oscillator or T1CKI input) or the INTRC oscillator, which can be selected in CONFIG3L<1>.

If the Timer1 oscillator will be used as the clock source for the RTCC, make sure to enable it by setting T1CON<3> (T1OSCEN). The selected RTC clock can be brought out to the RTCC pin by the RTSECSEL<1:0> bits in the PADCFG1 register.

17.2.3 DIGIT CARRY RULES

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

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

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

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

TABLE 17-1: DAY OF WEEK SCHEDULE

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

19.4 PWM (Enhanced Mode)

The Enhanced PWM mode can generate a PWM signal on up to four different output pins with up to 10 bits of resolution. It can do this through four different PWM Output modes:

- Single PWM mode
- Half-Bridge PWM mode
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the PxM bits of the CCPxCON register must be set appropriately.

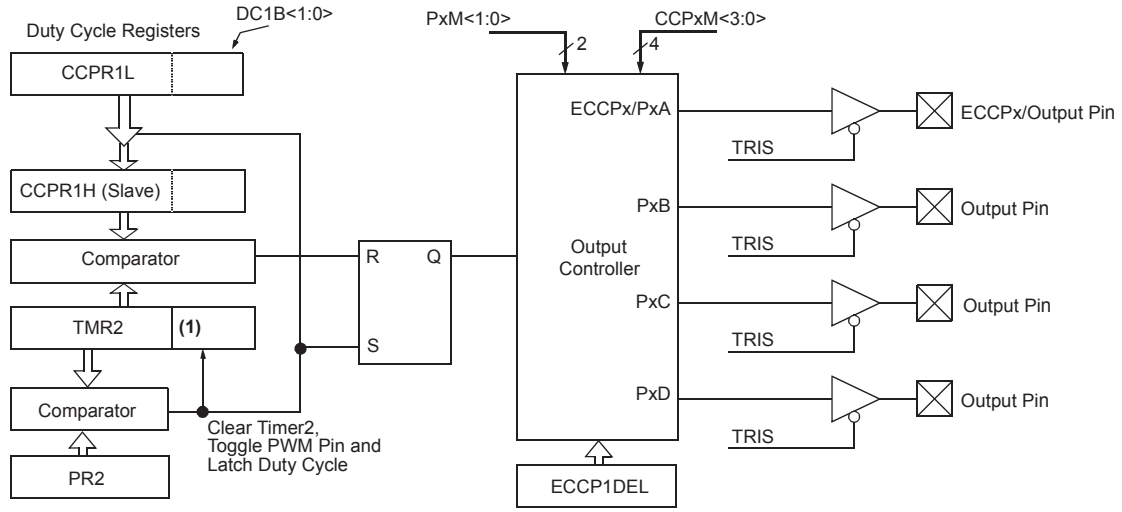
The PWM outputs are multiplexed with I/O pins and are designated: PxA, PxB, PxC and PxD. The polarity of the PWM pins is configurable and is selected by setting the CCPxM bits in the CCPxCON register appropriately.

Table 19-1 provides the pin assignments for each Enhanced PWM mode.

Figure 19-3 provides an example of a simplified block diagram of the Enhanced PWM module.

Note:	To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal.

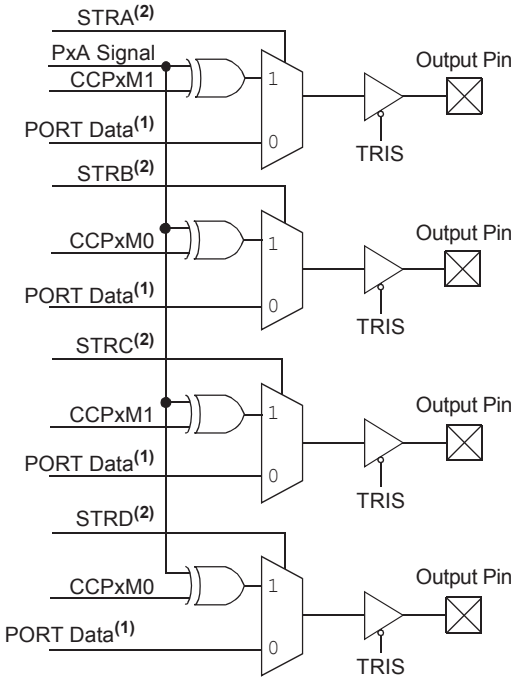
FIGURE 19-3: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE EXAMPLE



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

Note 1:	The TRIS register value for each PWM output must be configured appropriately.
2:	Any pin not used by an Enhanced PWM mode is available for alternate pin functions.

FIGURE 19-16: SIMPLIFIED STEERING BLOCK DIAGRAM



Note 1: Port outputs are configured as displayed when the CCPxCON register bits, PxM<1:0> = 00 and CCP1M<3:2> = 11.

2: Single PWM output requires setting at least one of the STR<D:A> bits.

19.4.7.1 Steering Synchronization

The STRSYNC bit of the PSTRxCON register gives the user two choices for when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRxCON register. In this case, the output signal at the Px<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figure 19-17 and Figure 19-18 illustrate the timing diagrams of the PWM steering depending on the STRSYNC setting.

FIGURE 19-17: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)

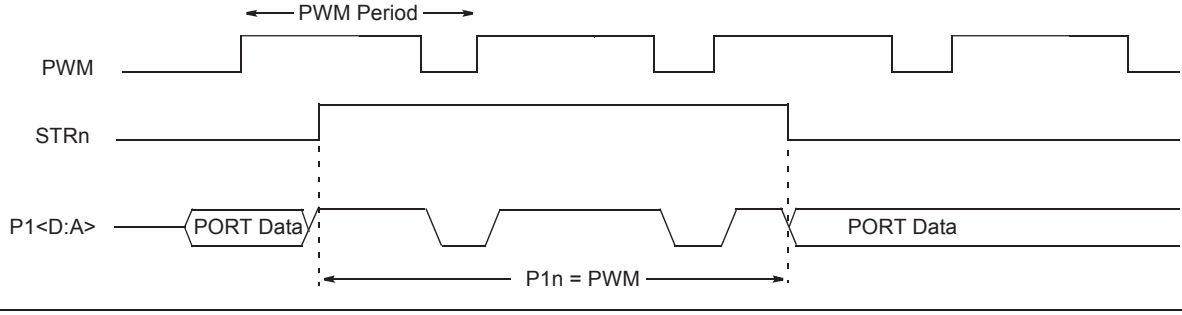
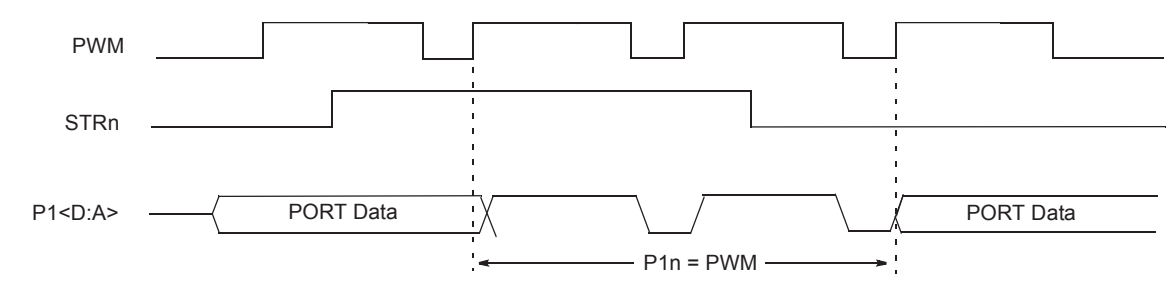


FIGURE 19-18: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)



REGISTER 20-2: SSPxCON1: MSSPx CONTROL REGISTER 1 (SPI MODE) (1, ACCESS FC6h; 2, F72h)

R/W-0	R/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
bit 7							bit 0

Legend:	C = Clearable 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 **WCOL:** Write Collision Detect bit
- 1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)

SPI Slave mode:
- 1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
- 0 = No overflow

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(2)
- 1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(3)
- 0101 = SPI Slave mode, clock = SCKx pin; $\overline{SSx}$ pin control disabled, $\overline{SSx}$ can be used as I/O pin
- 0100 = SPI Slave mode, clock = SCKx pin; $\overline{SSx}$ pin control enabled
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0010 = SPI Master mode, clock = FOSC/64
- 0001 = SPI Master mode, clock = FOSC/16
- 1010 = SPI Master mode, clock = FOSC/8
- 0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.

2: When enabled, this pin must be properly configured as input or output.

3: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

REGISTER 20-3: DMACON1: DMA CONTROL REGISTER 1 (ACCESS F88h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSCON1	SSCON0	TXINC	RXINC	DUPLEX1	DUPLEX0	DLYINTEN	DMAEN
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 **SSCON<1:0>:** $\overline{\text{SSDMA}}$ Output Control bits (Master modes only)
11 = $\overline{\text{SSDMA}}$ is asserted for the duration of 4 bytes; DLYINTEN is always reset low
01 = $\overline{\text{SSDMA}}$ is asserted for the duration of 2 bytes; DLYINTEN is always reset low
10 = $\overline{\text{SSDMA}}$ is asserted for the duration of 1 byte; DLYINTEN is always reset low
00 = $\overline{\text{SSDMA}}$ is not controlled by the DMA module; the DLYINTEN bit is software programmable

bit 5 **TXINC:** Transmit Address Increment Enable bit
Allows the transmit address to increment as the transfer progresses.
1 = The transmit address is to be incremented from the initial value of TXADDR<11:0>
0 = The transmit address is always set to the initial value of TXADDR<11:0>

bit 4 **RXINC:** Receive Address Increment Enable bit
Allows the receive address to increment as the transfer progresses.
1 = The received address is to be incremented from the initial value of RXADDR<11:0>
0 = The received address is always set to the initial value of RXADDR<11:0>

bit 3-2 **DUPLEX<1:0>:** Transmit/Receive Operating Mode Select bits
10 = SPI DMA operates in Full-Duplex mode; data is simultaneously transmitted and received
01 = DMA operates in Half-Duplex mode; data is transmitted only
00 = DMA operates in Half-Duplex mode; data is received only

bit 1 **DLYINTEN:** Delay Interrupt Enable bit
Enables the interrupt to be invoked after the number of T_{CY} cycles specified in DLYCYC<2:0> has elapsed from the latest completed transfer.
1 = The interrupt is enabled; SSCON<1:0> must be set to '00'
0 = The interrupt is disabled

bit 0 **DMAEN:** DMA Operation Start/Stop bit
This bit is set by the users' software to start the DMA operation. It is reset back to zero by the DMA engine when the DMA operation is completed or aborted.
1 = DMA is in session
0 = DMA is not in session

21.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs and so on.

The Enhanced USART module implements additional features, including Automatic Baud Rate Detection and calibration, automatic wake-up on Sync Break reception, and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

All members of the PIC18F47J13 Family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/CCP9/PMA5/TX1/CK1/RP17 and RC7/CCP10/PMA4/RX1/DT1/RP18), and remapped (RPn1/TX2/CK2 and RPn2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
  - SPEN bit (RCSTA1<7>) must be set (= 1)
  - TRISC<7> bit must be set (= 1)
  - TRISC<6> bit must be cleared (= 0) for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode
- For EUSART2:
  - SPEN bit (RCSTA2<7>) must be set (= 1)
  - TRIS bit for RPn2/RX2/DT2 = 1
  - TRIS bit for RPn1/TX2/CK2 = 0 for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The TXx/CKx I/O pins have an optional open-drain output capability. By default, when this pin is used by the EUSART as an output, it will function as a standard push-pull CMOS output. The TXx/CKx I/O pins' open-drain, output feature can be enabled by setting the corresponding UxOD bit in the ODCON2 register. For more details, see **Section 20.3.3 "Open-Drain Output Option"**.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are covered in detail in Register 21-1, Register 21-2 and Register 21-3, respectively.

Note:	Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

REGISTER 21-2: RCSTAx: RECEIVE STATUS AND CONTROL REGISTER (1, ACCESS FACh; 2, FC9h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-Bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-Bit (RX9 = 1):
1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and the ninth bit can be used as a parity bit
Asynchronous mode 8-Bit (RX9 = 0):
Don't care.

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be cleared by reading the RCREGx register and receiving the next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit, CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data
This can be an address/data bit or a parity bit and must be calculated by user firmware.

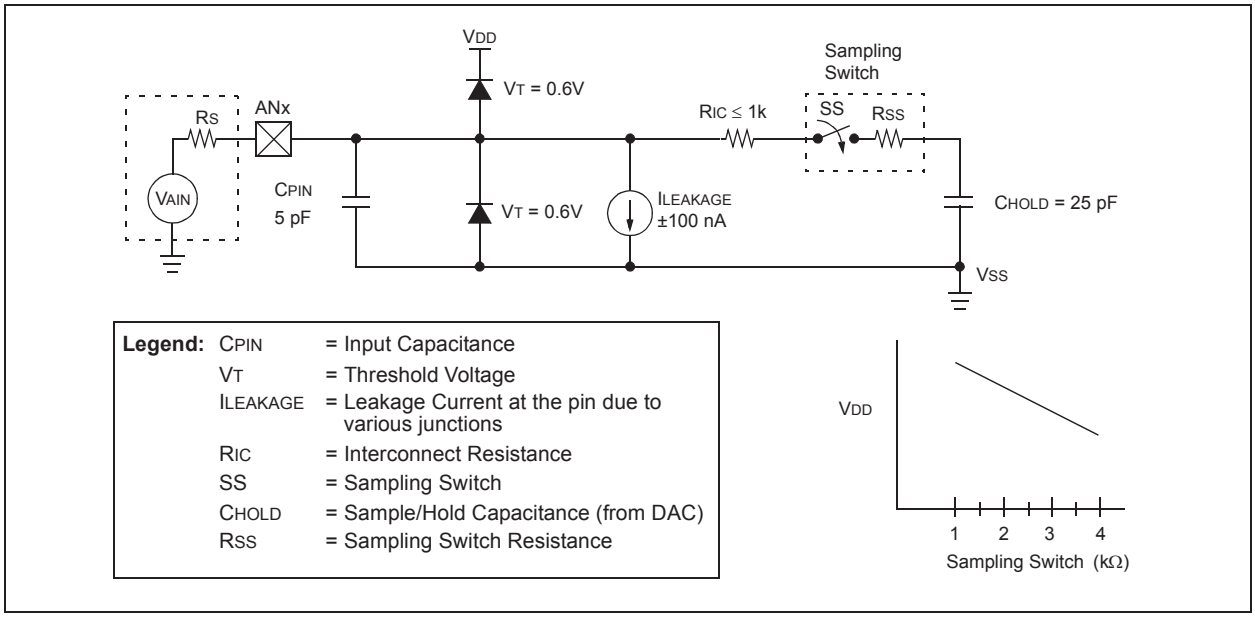
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine acquisition time, see **Section 22.1 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

1. Configure the A/D module:
   - Configure the required ADC pins as analog pins using ANCON0, ANCON1
   - Set voltage reference using ADCON0
   - Select A/D input channel (ADCON0)
   - Select A/D acquisition time (ADCON1)
   - Select A/D conversion clock (ADCON1)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time (if required).
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear bit, ADIF, if required.
7. For the next conversion, go to Step 1 or Step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum Wait of 2 TAD is required before the next acquisition starts.

FIGURE 22-2: ANALOG INPUT MODEL



REGISTER 27-8: CONFIG4H: CONFIGURATION REGISTER 4 HIGH (BYTE ADDRESS 300007h)

U-1	U-1	U-1	U-1	U-0	U-0	R/WO-1	R/WO-1
—	—	—	—	—	—	WPEND	WPDIS
bit 7							bit 0

Legend:			
R = Readable bit	WO = Write-Once 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:** Program the corresponding Flash Configuration bit to '1'

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **WPEND:** Write-Protect Disable bit
1 = Flash pages, WPFP<6:0> to (Configuration Words page), are erase/write-protected
0 = Flash pages 0 to WPFP<6:0> are erase/write-protected

bit 0 **WPDIS:** Write-Protect Disable bit
1 = WPFP<5:0>, WPEND and WPCFG bits are ignored; all Flash memory may be erased or written
0 = WPFP<5:0>, WPEND and WPCFG bits are enabled; erase/write-protect is active for the selected region(s)

REGISTER 27-9: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F47J13 FAMILY DEVICES (BYTE ADDRESS 3FFFFEh)

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

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

bit 7-5 **DEV<2:0>:** Device ID bits
These bits are used with DEV<10:3> bits in Device ID Register 2 to identify the part number. See Register 27-10.

bit 4-0 **REV<4:0>:** Revision ID bits
These bits are used to indicate the device revision.

ANDWF **AND W with f**

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

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

0001	01da	ffff	ffff

Description: The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

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

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 28.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ANDWF REG, 0, 0`

Before Instruction
W = 17h
REG = C2h
After Instruction
W = 02h
REG = C2h

BC **Branch if Carry**

Syntax: BC n

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

Operation: if Carry bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0010	nnnn	nnnn

Description: If the Carry 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 BC 5`

Before Instruction
PC = address (HERE)
After Instruction
If Carry = 1;
PC = address (HERE + 12)
If Carry = 0;
PC = address (HERE + 2)

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 (default).

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 28.2.3 "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 (default).

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 28.2.3 "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:

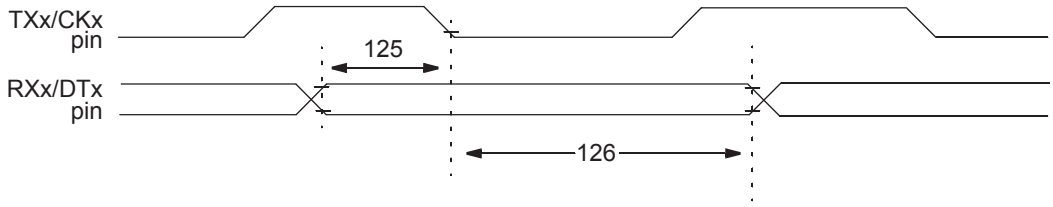
```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

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

TABLE 30-29: EUSARTx SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TCKH2DTV	Sync XMIT (Master and Slave) Clock High to Data Out Valid	—	40	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	TDTRF	Data Out Rise Time and Fall Time	—	20	ns	

FIGURE 30-22: EUSARTx SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



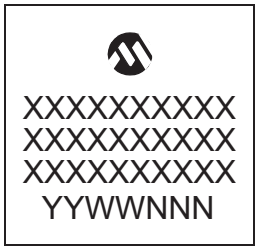
Note: Refer to Figure 30-4 for load conditions.

TABLE 30-30: EUSARTx SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	Sync RCV (Master and Slave) Data Hold before CKx ↓ (DTx hold time)	10	—	ns	
126	TCKL2DTL	Data Hold after CKx ↓ (DTx hold time)	15	—	ns	

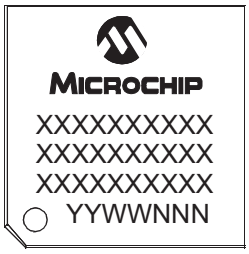
Package Marking Information (Continued)

44-Lead QFN



Example

44-Lead TQFP



Example