**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, USB |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 22 |
| Program Memory Size | 128KB (64K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 3.8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.15V ~ 3.6V |
| Data Converters | A/D 10x10b/12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 28-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f27j53t-i-ss |

**TABLE 1-3: PIC18F2XJ53 PINOUT I/O DESCRIPTIONS (CONTINUED)**

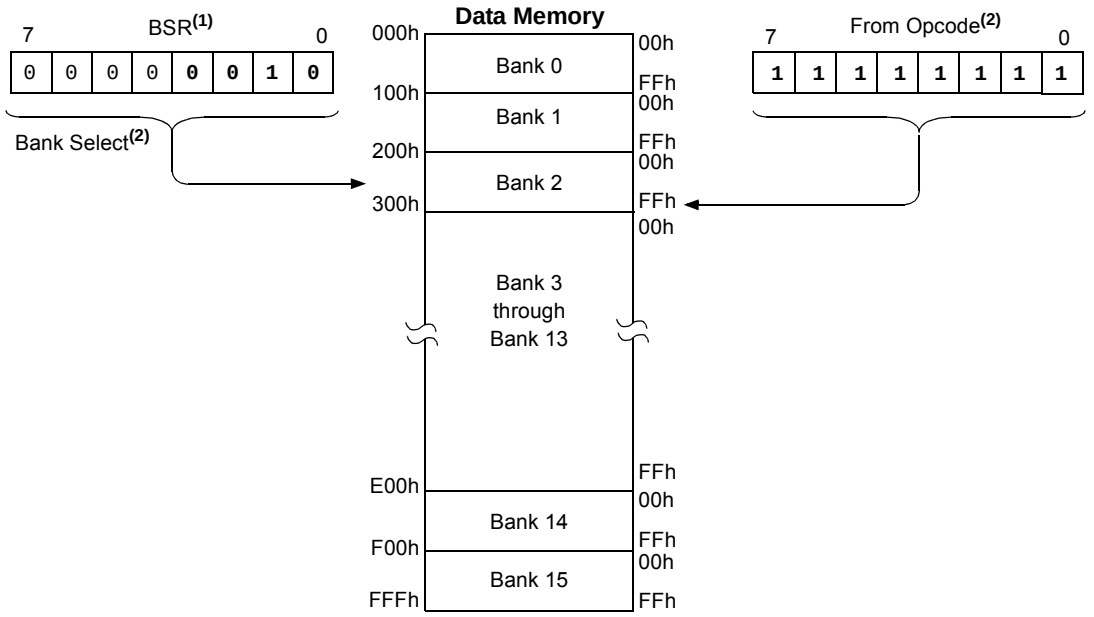
| Pin Name | Pin Number | | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|
| | 28-SPDIP/ SSOP/ SOIC | 28-QFN | | | |
| | | | | | PORTC is a bidirectional I/O port. |
| RC0/T1OSO/T1CKI/RP11 | 11 | 8 | | | |
| RC0 | | | I/O | ST/DIG | Digital I/O. |
| T1OSO | | | O | Analog | Timer1 oscillator output. |
| T1CKI | | | I | ST | Timer1 external digital clock input. |
| RP11 | | | I/O | ST/DIG | Remappable Peripheral Pin 11 input/output. |
| RC1/CCP8/T1OSI/$\overline{UOE}$/RP12 | 12 | 9 | | | |
| RC1 | | | I/O | ST/DIG | Digital I/O. |
| CCP8 | | | I/O | ST/DIG | Capture/Compare/PWM input/output. |
| T1OSI | | | I | Analog | Timer1 oscillator input. |
| $\overline{UOE}$ | | | O | DIG | External USB transceiver NOE output. |
| RP12 | | | I/O | ST/DIG | Remappable Peripheral Pin 12 input/output. |
| RC2/AN11/C2IND/CTPLS/ RP13 | 13 | 10 | | | |
| RC2 | | | I/O | ST/DIG | Digital I/O. |
| AN11 | | | I | Analog | Analog Input 11. |
| C2IND | | | I | Analog | Comparator 2 Input D. |
| CTPLS | | | O | DIG | CTMU pulse generator output. |
| RP13 | | | I/O | ST/DIG | Remappable Peripheral Pin 13 input/output. |
| RC4/D-/VM | 15 | 12 | | | |
| RC4 | | | I | ST | Digital Input. |
| D- | | | I/O | — | USB bus minus line input/output. |
| VM | | | I | ST | External USB transceiver FM input. |
| RC5/D+/VP | 16 | 13 | | | |
| RC5 | | | I | ST | Digital Input. |
| D+ | | | I/O | — | USB bus plus line input/output. |
| VP | | | I | ST | External USB transceiver VP input. |
| RC6/CCP9/TX1/CK1/RP17 | 17(2) | 14(2) | | | |
| RC6 | | | I/O | ST/DIG | Digital I/O. |
| CCP9 | | | I/O | ST/DIG | Capture/Compare/PWM input/output. |
| TX1 | | | O | DIG | EUSART1 asynchronous transmit. |
| CK1 | | | I/O | ST/DIG | EUSART1 synchronous clock (see related RX1/DT1). |
| RP17 | | | I/O | ST/DIG | Remappable Peripheral Pin 17 input/output. |
| RC7/CCP10/RX1/DT1/SDO1/ RP18 | 18(2) | 15(2) | | | Digital I/O. |
| RC7 | | | I/O | ST/DIG | Asynchronous serial receive data input. |
| CCP10 | | | I/O | ST/DIG | Capture/Compare/PWM input/output. |
| RX1 | | | I | ST | Synchronous serial data output/input. |
| DT1 | | | I/O | ST/DIG | SPI data output. |
| SDO1 | | | O | DIG | Remappable Peripheral Pin 18 input/output. |
| RP18 | | | I/O | ST/DIG | |

**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); DIG = Digital output; $I^2C$ = Open-Drain, $I^2C$ specific

**Note 1:** RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

**2:** 5.5V tolerant.

**FIGURE 6-7: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)**




**Note 1:** The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

**2:** The MOVFF instruction embeds the entire 12-bit address in the instruction.

### 6.3.3 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the Access RAM and is composed of GPRs. The upper half is where the device's SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 6-6).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the 'a' parameter in the instruction). When 'a' is equal to '1', the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When 'a' is '0', however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this "forced" addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 6.6.3 "Mapping the Access Bank in Indexed Literal Offset Mode"**.

### 6.3.4 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM, which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upward toward the bottom of the SFR area. GPRs are not initialized by a POR and are unchanged on all other Resets.

**TABLE 6-4: REGISTER FILE SUMMARY (PIC18F47J53 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| FDCh | PREINC2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) | | | | | | | | N/A |
| FDBh | PLUSW2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W | | | | | | | | N/A |
| FDAh | FSR2H | — | — | — | — | Indirect Data Memory Address Pointer 2 High Byte | | | | ---- xxxx |
| FD9h | FSR2L | Indirect Data Memory Address Pointer 2 Low Byte | | | | | | | | xxxx xxxx |
| FD8h | STATUS | — | — | — | N | OV | Z | DC | C | ---x xxxx |
| FD7h | TMR0H | Timer0 Register High Byte | | | | | | | | 0000 0000 |
| FD6h | TMR0L | Timer0 Register Low Byte | | | | | | | | xxxx xxxx |
| FD5h | T0CON | TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 | 1111 1111 |
| FD3h | OSCCON | IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | FLTS | SCS1 | SCS0 | 0110 q000 |
| FD2h | CM1CON | CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 | 0001 1111 |
| FD1h | CM2CON | CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 | 0001 1111 |
| FD0h | RCON | IPEN | — | $\overline{CM}$ | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ | 0-11 11qq |
| FCFh | TMR1H | Timer1 Register High Byte | | | | | | | | xxxx xxxx |
| FCEh | TMR1L | Timer1 Register Low Bytes | | | | | | | | xxxx xxxx |
| FCDh | T1CON | TMR1CS1 | TMR1CS0 | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | RD16 | TMR1ON | 0000 0000 |
| FCCh | TMR2 | Timer2 Register | | | | | | | | 0000 0000 |
| FCBh | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 |
| FCAh | T2CON | — | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 |
| FC9h | SSP1BUF | MSSP1 Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx |
| FC8h | SSP1ADD | MSSP1 Address Register ($I^2C$ Slave Mode). MSSP1 Baud Rate Reload Register ($I^2C$ Master Mode). | | | | | | | | 0000 0000 |
| FC8h | SSP1MSK | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 | ---- ---- |
| FC7h | SSP1STAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 1111 1111 |
| FC6h | SSP1CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 |
| FC5h | SSP1CON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 |
| FC5h | SSP1CON2 | GCEN | ACKSTAT | ADMSK5 | ADMSK4 | ADMSK3 | ADMSK2 | ADMSK1 | SEN | 0000 0000 |
| FC4h | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx |
| FC3h | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx |
| FC2h | ADCON0 | VCFG1 | VCFG0 | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | ADON | 0000 0000 |
| FC1h | ADCON1 | ADFM | ADCAL | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 | 0000 0000 |
| FC0h | WDTCON | REGSLP | LVDSTAT | ULPLVL | VBGOE | DS | ULPEN | ULPSINK | SWDTEN | 1xx0 0000 |
| FBFh | PSTR1CON | CMPL1 | CMPL0 | — | STRSYNC | STRD | STRC | STRB | STRA | 00-0 0001 |
| FBEh | ECCP1AS | ECCP1ASE | ECCP1AS2 | ECCP1AS1 | ECCP1AS0 | PSS1AC1 | PSS1AC0 | PSS1BD1 | PSS1BD0 | 0000 0000 |
| FBDh | ECCP1DEL | P1RSEN | P1DC6 | P1DC5 | P1DC4 | P1DC3 | P1DC2 | P1DC1 | P1DC0 | 0000 0000 |
| FBCh | CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | xxxx xxxx |
| FBBh | CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | xxxx xxxx |
| FBAh | CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 |
| FB9h | PSTR2CON | CMPL1 | CMPL0 | — | STRSYNC | STRD | STRC | STRB | STRA | 00-0 0001 |
| FB8h | ECCP2AS | ECCP2ASE | ECCP2AS2 | ECCP2AS1 | ECCP2AS0 | PSS2AC1 | PSS2AC0 | PSS2BD1 | PSS2BD0 | 0000 0000 |
| FB7h | ECCP2DEL | P2RSEN | P2DC6 | P2DC5 | P2DC4 | P2DC3 | P2DC2 | P2DC1 | P2DC0 | 0000 0000 |
| FB6h | CCPR2H | Capture/Compare/PWM Register 2 High Byte | | | | | | | | xxxx xxxx |
| FB5h | CCPR2L | Capture/Compare/PWM Register 2 Low Byte | | | | | | | | xxxx xxxx |
| FB4h | CCP2CON | P2M1 | P2M0 | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 0000 0000 |
| FB3h | CTMUCONH | CTMUEN | — | CTMUSIDL | TGEN | EDGEN | EDGSEQEN | IDISSEN | CTTRIG | 0-00 0000 |
| FB2h | CTMUCONL | EDG2POL | EDG2SEL1 | EDG2SEL0 | EDG1POL | EDG1SEL1 | EDG1SEL0 | EDG2STAT | EDG1STAT | 0000 00xx |
| FB1h | CTMUICON | ITRIM5 | ITRIM4 | ITRIM3 | ITRIM2 | ITRIM1 | ITRIM0 | IRNG1 | IRNG0 | 0000 0000 |
| FB0h | SPBRG1 | EUSART1 Baud Rate Generator Register Low Byte | | | | | | | | 0000 0000 |
| FAFh | RCREG1 | EUSART1 Receive Register | | | | | | | | 0000 0000 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Implemented only for 44-pin devices (PIC18F46J53, PIC18F47J53, PIC18LF46J53 and PIC18LF47J53).

**2:** Implemented only for 28-pin devices (PIC18F26J53, PIC18F27J53, PIC18LF26J53 and PIC18LF27J53).

**3:** Implemented only for devices with 128 Kbyte of program memory (PIC18F27J53, PIC18F47J53, PIC18LF27J53 and PIC18LF47J53).

**REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2 (ACCESS FF1h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | INTEDG3 | TMR0IP | INT3IP | RBIP |
| 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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

**NOTES:**

## 15.0 TIMER3/5 MODULE

The Timer3/5 timer/counter modules incorporate these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMRxH and TMRxL)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on ECCP Special Event Trigger

**Note:** Throughout this section, generic references are used for register and bit names that are the same – except for an 'x' variable that indicates the item's association with the Timer3 or Timer5 module. For example, the control register is named TxCON, and refers to T3CON and T5CON.

A simplified block diagram of the Timer3/5 module is shown in Figure 15-1.

The Timer3/5 module is controlled through the TxCON register (Register 15-1). It also selects the clock source options for the ECCP modules. (For more information, see **Section 19.1.1 "ECCP Module and Timer Resources"**.)

The FOSC clock source should not be used with the ECCP capture/compare features. If the timer will be used with the capture or compare features, always select one of the other timer clocking options.

**REGISTER 16-1: TxCON: TIMER4/6/8 CONTROL REGISTER (ACCESS F76h, BANKED F1Eh, BANKED F1Bh)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TxOUTPS3 | TxOUTPS2 | TxOUTPS1 | TxOUTPS0 | TMRxON | TxCKPS1 | TxCKPS0 |
| 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 **Unimplemented:** Read as '0'

bit 6-3 **TxOUTPS<3:0>:** Timerx Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMRxON**: Timerx On bit
1 = Timerx is on
0 = Timerx is off

bit 1-0 **TxCKPS<1:0>:** Timerx Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

## 16.2 Timer4/6/8 Interrupt

The Timer4/6/8 modules have 8-bit Period registers, PRx, that are both readable and writable. Timer4/6/8 increment from 00h until they match PR4/6/8 and then reset to 00h on the next increment cycle. The PRx registers are initialized to FFh upon Reset.

## 16.3 Output of TMRx

The outputs of TMRx (before the postscaler) are used only as a PWM time base for the ECCP modules. They are not used as baud rate clocks for the MSSP modules as is the Timer2 output.

**FIGURE 16-1: TIMER4 BLOCK DIAGRAM**

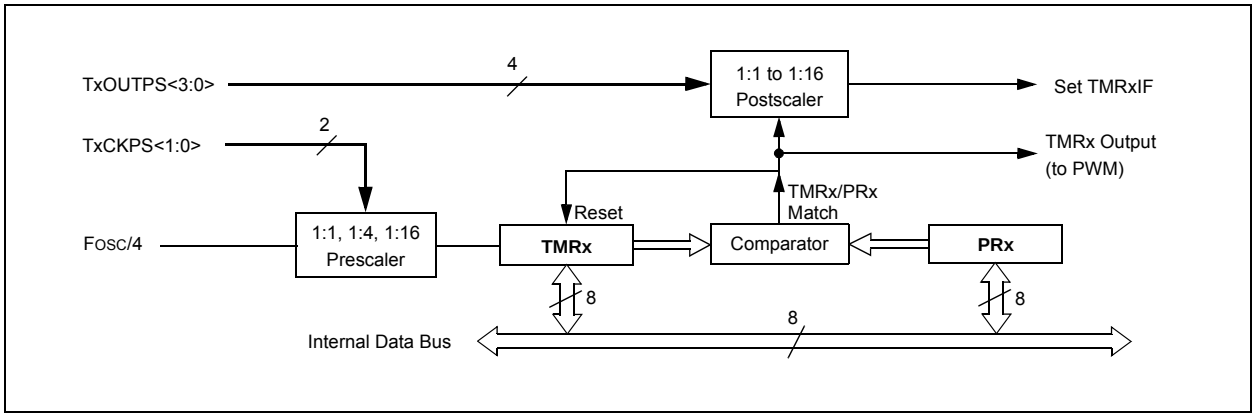


**TABLE 17-2: DAY TO MONTH ROLLOVER SCHEDULE**

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

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

### 17.2.4 LEAP YEAR

Since the year range on the RTCC module is 2000 to 2099, the leap year calculation is determined by any year divisible by '4' in the above range. Only February is effected in a leap year.

February will have 29 days in a leap year and 28 days in any other year.

### 17.2.5 GENERAL FUNCTIONALITY

All Timer registers containing a time value of seconds or greater are writable. The user configures the time by writing the required year, month, day, hour, minutes and seconds to the Timer registers, via register pointers (see **Section 17.2.8 "Register Mapping"**).

The timer uses the newly written values and proceeds with the count from the required starting point.

The RTCC is enabled by setting the RTCEN bit (RTCCFG<7>). If enabled, while adjusting these registers, the timer still continues to increment. However, any time the MINSEC register is written to, both of the timer prescalers are reset to '0'. This allows fraction of a second synchronization.

The Timer registers are updated in the same cycle as the write instruction's execution by the CPU. The user must ensure that when RTCEN = 1, the updated registers will not be incremented at the same time. This can be accomplished in several ways:

- By checking the RTCSYNC bit (RTCCFG<4>)
- By checking the preceding digits from which a carry can occur
- By updating the registers immediately following the seconds pulse (or alarm interrupt)

The user has visibility to the half-second field of the counter. This value is read-only and can be reset only by writing to the lower half of the SECONDS register.

### 17.2.6 SAFETY WINDOW FOR REGISTER READS AND WRITES

The RTCSYNC bit indicates a time window during which the RTCC Clock Domain registers can be safely read and written without concern about a rollover. When RTCSYNC = 0, the registers can be safely accessed by the CPU.

Whether RTCSYNC = 1 or 0, the user should employ a firmware solution to ensure that the data read did not fall on a rollover boundary, resulting in an invalid or partial read. This firmware solution would consist of reading each register twice and then comparing the two values. If the two values matched, then, a rollover did not occur.

### 17.2.7 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RTCCFG<5>) must be set.

To avoid accidental writes to the RTCC Timer register, it is recommended that the RTCWREN bit (RTCCFG<5>) be kept clear at any time other than while writing to it. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN. For that reason, it is recommended that users follow the code example in Example 17-1.

**EXAMPLE 17-1: SETTING THE RTCWREN BIT**

```
movlb   0x0F            ;RTCCFG is banked
bcf     INTCON, GIE     ;Disable interrupts
movlw   0x55
movwf   EECON2
movlw   0xAA
movwf   EECON2
bsf     RTCCFG,RTCWREN
```

### 17.2.8 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Timer registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH<15:8> and RTCVALL<7:0>) uses the RTCPTR bits (RTCCFG<1:0>) to select the required Timer register pair.

By reading or writing to the RTCVALH register, the RTCC Pointer value (RTCPTR<1:0>) decrements by 1 until it reaches '00'. Once it reaches '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

## 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 | All Resets |
|---|---|---|---|---|---|---|---|---|---|
| RTCCFG | RTCEN | — | RTCWREN | RTCSYNC | HALFSEC | RTCOE | RTCPTR1 | RTCPTR0 | 0000 |
| RTCCAL | CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 | 0000 |
| PADCFG1 | — | — | — | — | — | RTSECSEL1 | RTSECSEL0 | PMPTTL | 0000 |
| ALRMCFG | ALRMEN | CHIME | AMASK3 | AMASK2 | AMASK1 | AMASK0 | ALRMPTR1 | ALRMPTR0 | 0000 |
| ALRMRPT | ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 | 0000 |
| PIR3 | SSP2IF | BCL2IF | RC2IF | TX2IF | TMR4IF | CTMUIF | TMR3GIF | RTCCCIF | 0000 |
| PIE3 | SSP2IE | BCL2IE | RC2IE | TX2IE | TMR4IE | CTMUIE | TMR3GIE | RTCCCIE | 0000 |
| IPR3 | SSP2IP | BCL2IP | RC2IP | TX2IP | TMR4IP | CTMUIP | TMR3GIP | RTCCCIP | 0000 |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal for 44-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 | All Resets |
|---|---|---|---|---|---|---|---|---|---|
| RTCVALH | RTCC Value Register Window High Byte, Based on RTCPTR<1:0> | | | | | | | | xxxx |
| RTCVALL | RTCC Value Register Window Low Byte, Based on RTCPTR<1:0> | | | | | | | | xxxx |
| RTCCFG | RTCEN | — | RTCWREN | RTCSYNC | HALFSEC | RTCOE | RTCPTR1 | RTCPTR0 | 0000 |
| ALRMCFG | ALRMEN | CHIME | AMASK3 | AMASK2 | AMASK1 | AMASK0 | ALRMPTR1 | ALRMPTR0 | 0000 |
| ALRMVALH | Alarm Value Register Window High Byte, Based on ALRMPTR<1:0> | | | | | | | | xxxx |
| ALRMVALL | Alarm Value Register Window Low Byte, Based on ALRMPTR<1:0> | | | | | | | | xxxx |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal for 44-pin devices.

**TABLE 17-7: ALARM VALUE REGISTERS**

| File Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|---|---|---|---|---|---|---|---|---|---|
| ALRMRPT | ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 | 0000 |
| ALRMVALH | Alarm Value Register Window High Byte, Based on ALRMPTR<1:0> | | | | | | | | xxxx |
| ALRMVALL | Alarm Value Register Window Low Byte, Based on ALRMPTR<1:0> | | | | | | | | xxxx |
| RTCCAL | CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 | 0000 |
| RTCVALH | RTCC Value Register Window High Byte, Based on RTCPTR<1:0> | | | | | | | | xxxx |
| RTCVALL | RTCC Value Register Window Low Byte, Based on RTCPTR<1:0> | | | | | | | | xxxx |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal for 44-pin devices.

### 18.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

The open-drain output option is controlled by the CCPxOD bits (ODCON1<7:0> and ODCON2<3:2>). Setting the appropriate bit configures the pin for the corresponding module for open-drain operation.

## 18.2 Capture Mode

In Capture mode, the CCPR4H:CCPR4L register pair captures the 16-bit value of the TMR1 or TMR3 register when an event occurs on the CCP4 pin, RB4. An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

The event is selected by the mode select bits, CCP4M<3:0> (CCP4CON<3:0>). When a capture is made, the interrupt request flag bit, CCP4IF (PIR4<1>), is set. (It must be cleared in software.) If another capture occurs before the value in register, CCPR4, is read, the old captured value is overwritten by the new captured value.

Figure 18-1 shows the Capture mode block diagram.

### 18.2.1 CCP PIN CONFIGURATION

In Capture mode, the appropriate CCPx pin should be configured as an input by setting the corresponding TRIS direction bit.

| | |
|---|---|
| **Note:** | If RB4 is configured as a CCP4 output, a write to the port causes a capture condition. |

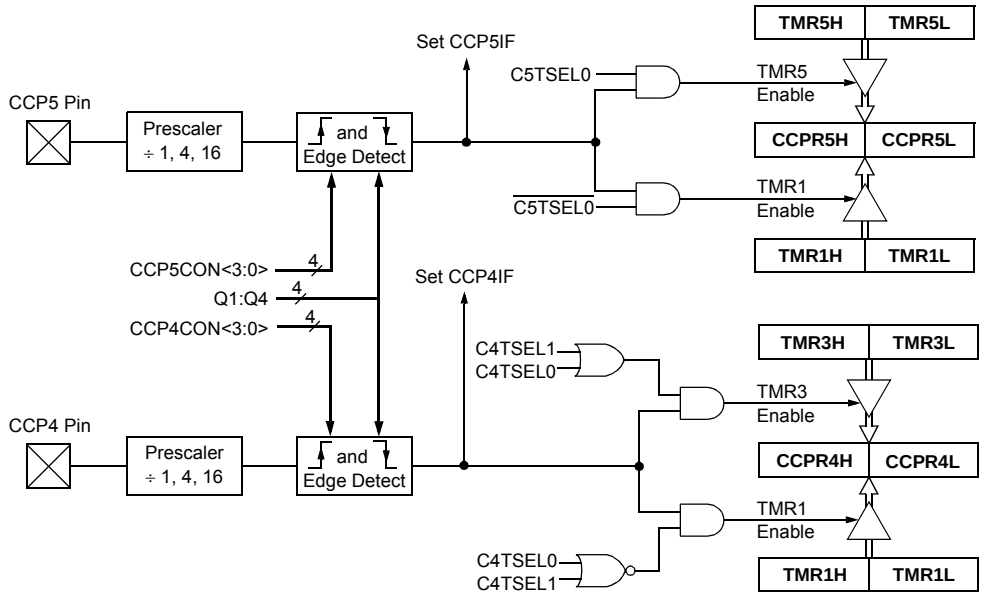
### 18.2.2 TIMER1/3/5 MODE SELECTION

For the available timers (1/3/5) to be used for the capture feature, the used timers must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work.

The timer to be used with each CCP module is selected in the CCPTMRSx registers. (See **Section 18.1.1 "CCP Modules and Timer Resources"**.)

Details of the timer assignments for the CCP modules are given in Table 18-2 and Table 18-3.

**FIGURE 18-1: 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 18-2 and Table 18-3.

### 18.2.3 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.

### 18.2.4 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 also will not clear the prescaler counter – meaning the first capture may be from a non-zero prescaler.

Example 18-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

**EXAMPLE 18-1: 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
```

## 18.3 Compare Mode

In Compare mode, the 16-bit CCPR4 register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCP4 pin can be:

- Driven high
- Driven low
- Toggled (high-to-low or low-to-high)
- Unchanged (that is, reflecting the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCP4M<3:0>). At the same time, the interrupt flag bit, CCP4IF, is set.

Figure 18-2 gives the Compare mode block diagram

### 18.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

| | |
|---|---|
| **Note:** | Clearing the CCP4CON register will force the RB4 compare output latch (depending on device configuration) to the default low level. This is not the PORTB I/O data latch. |

### 18.3.2 TIMER1/3/5 MODE SELECTION

If the CCP module is using the compare feature in conjunction with any of the Timer1/3/5 timers, the timers must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the compare operation may not work.

| | |
|---|---|
| **Note:** | Details of the timer assignments for the CCP modules are given in Table 18-2 and Table 18-3. |

### 18.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCP4M<3:0> = 1010), the CCP4 pin is not affected. Only a CCP interrupt is generated, if enabled, and the CCP4IE bit is set.

### 18.3.4 SPECIAL EVENT TRIGGER

Both CCP modules are equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCP4M<3:0> = 1011).

For either CCP module, the Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module’s time base. This allows the CCPRx registers to serve as a programmable period register for either timer.

The Special Event Trigger for CCP4 cannot start an A/D conversion.

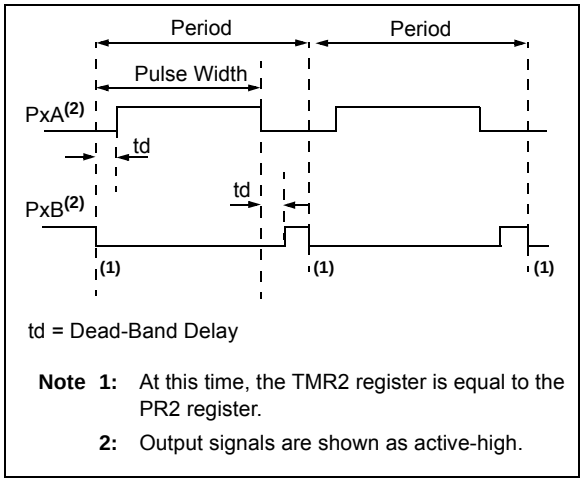
| | |
|---|---|
| **Note:** | The Special Event Trigger of ECCP1 can start an A/D conversion, but the A/D Converter must be enabled. For more information, see **Section 19.0 “Enhanced Capture/Compare/PWM (ECCP) Module”**. |

### 19.4.6 PROGRAMMABLE DEAD-BAND DELAY MODE

In half-bridge applications, where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on and the other turned off), both switches may be on for a short period until one switch completely turns off. During this brief interval, a very high current (shoot-through current) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable, dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. For an illustration, see Figure 19-14. The lower seven bits of the associated ECCPxDEL register (Register 19-5) sets the delay period in terms of microcontroller instruction cycles ($T_{CY}$ or 4 $T_{OSC}$).

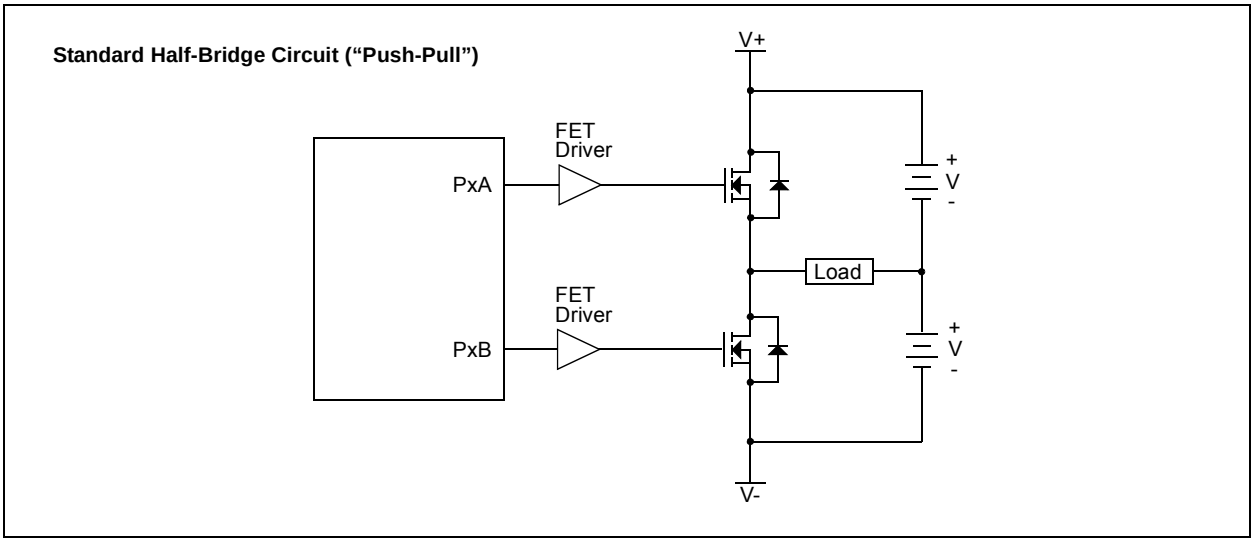
**FIGURE 19-14: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**




**Note 1:** At this time, the TMR2 register is equal to the PR2 register.

**2:** Output signals are shown as active-high.

**FIGURE 19-15: EXAMPLE OF HALF-BRIDGE APPLICATIONS**



**EXAMPLE 20-2: 512-BYTE SPI MASTER MODE `Init` AND TRANSFER**

```
                                ;For this example, let's use RP5(RB2) for SCK2,
                                ;RP4(RB1) for SDO2, and RP3(RB0) for SDI2

                                ;Let’s use SPI master mode, CKE = 0, CKP = 0,
                                ;without using slave select signalling.

InitSPIPins:
   movlb     0x0F               ;Select bank 15, for access to ODCON3 register
   bcf       ODCON3, SPI2OD     ;Let’s not use open drain outputs in this example

   bcf       LATB, RB2          ;Initialize our (to be) SCK2 pin low (idle).
   bcf       LATB, RB1          ;Initialize our (to be) SDO2 pin to an idle state
   bcf       TRISB, RB1         ;Make SDO2 output, and drive low
   bcf       TRISB, RB2         ;Make SCK2 output, and drive low (idle state)
   bsf       TRISB, RB0         ;SDI2 is an input, make sure it is tri-stated

                                ;Now we should unlock the PPS registers, so we can
                                ;assign the MSSP2 functions to our desired I/O pins.

   movlb     0x0E               ;Select bank 14 for access to PPS registers
   bcf       INTCON, GIE        ;I/O Pin unlock sequence will not work if CPU
                                ;services an interrupt during the sequence
   movlw     0x55               ;Unlock sequence consists of writing 0x55
   movwf     EECON2             ;and 0xAA to the EECON2 register.
   movlw     0xAA
   movwf     EECON2
   bcf       PPSCON, IOLOCK     ;We may now write to RPINRx and RPORx registers
   bsf       INTCON, GIE        ;May now turn back on interrupts if desired

   movlw     0x03               ;RP3 will be SDI2
   movwf     RPINR21            ;Assign the SDI2 function to pin RP3

   movlw     0x0A               ;Let’s assign SCK2 output to pin RP4
   movwf     RPOR4              ;RPOR4 maps output signals to RP4 pin
   movlw     0x04               ;SCK2 also needs to be configured as an input on the
                                same pin
   movwf     RPINR22            ;SCK2 input function taken from RP4 pin
   movlw     0x09               ;0x09 is SDO2 output
   movwf     RPOR5              ;Assign SDO2 output signal to the RP5 (RB2) pin
   movlb     0x0F               ;Done with PPS registers, bank 15 has other SFRs

InitMSSP2:
   clrf      SSP2STAT           ;CKE = 0, SMP = 0 (sampled at middle of bit)
   movlw     b'00000000'        ;CKP = 0, SPI Master mode, Fosc/4
   movwf     SSP2CON1           ;MSSP2 initialized
   bsf       SSP2CON1, SSPEN    ;Enable the MSSP2 module

InitSPIDMA:
   movlw     b'00111010'        ;Full duplex, RX/TXINC enabled, no SSCON
   movwf     DMACON1            ;DLYINTEN is set, so DLYCYC3:DLYCYC0 = 1111
   movlw     b'11110000'        ;Minimum delay between bytes, interrupt
   movwf     DMACON2            ;only once when the transaction is complete
```

**REGISTER 21-3: BAUDCONx: BAUD RATE CONTROL REGISTER (ACCESS F7Eh, F7Ch)**

| R/W-0 | R-1 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN |
| 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 **ABDOVF:** Auto-Baud Acquisition Rollover Status bit
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)
0 = No BRG rollover has occurred

bit 6 **RCIDL:** Receive Operation Idle Status bit
1 = Receive operation is Idle
0 = Receive operation is active

bit 5 **RXDTP:** Data/Receive Polarity Select bit
Asynchronous mode:
1 = Receive data (RXx) is inverted (active-low)
0 = Receive data (RXx) is not inverted (active-high)
Synchronous mode:
1 = Data (DTx) is inverted (active-low)
0 = Data (DTx) is not inverted (active-high)

bit 4 **TXCKP:** Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TXx) is a low level
0 = Idle state for transmit (TXx) is a high level
Synchronous mode:
1 = Idle state for clock (CKx) is a high level
0 = Idle state for clock (CKx) is a low level

bit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx
0 = 8-bit Baud Rate Generator – SPBRGx only (Compatible mode), SPBRGHx value is ignored

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = EUSART will continue to sample the RXx pin – interrupt is generated on the falling edge; bit is cleared in hardware on the following rising edge
0 = RXx pin is not monitored or the rising edge detected
Synchronous mode:
Unused in this mode.

bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character; requires reception of a Sync field (55h); cleared in hardware upon completion
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode.

**REGISTER 23-5: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER (BD0STAT THROUGH BD63STAT), CPU MODE**

| R/W-x | R/W-x | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| UOWN(1) | DTS(2) | —(3) | —(3) | DTSEN | BSTALL | BC9 | BC8 |
| 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 **UOWN:** USB Own bit(1)
- 0 = The microcontroller core owns the BD and its corresponding buffer

bit 6 **DTS:** Data Toggle Synchronization bit(2)
- 1 = Data 1 packet
- 0 = Data 0 packet

bit 5-4 **Unimplemented:** These bits should always be programmed to '0'(3)

bit 3 **DTSEN:** Data Toggle Synchronization Enable bit
- 1 = Data toggle synchronization is enabled; data packets with incorrect Sync value will be ignored except for a SETUP transaction, which is accepted even if the data toggle bits do not match
- 0 = No data toggle synchronization is performed

bit 2 **BSTALL:** Buffer Stall Enable bit
- 1 = Buffer stall enabled; STALL handshake issued if a token is received that would use the BD in the given location (UOWN bit remains set, BD value is unchanged)
- 0 = Buffer stall disabled

bit 1-0 **BC<9:8>:** Byte Count 9 and 8 bits

The byte count bits represent the number of bytes that will be transmitted for an IN token or received during an OUT token. Together with BC<7:0>, the valid byte counts are 0-1023.

**Note 1:** This bit must be initialized by the user to the desired value prior to enabling the USB module.

**2:** This bit is ignored unless DTSEN = 1.

**3:** If these bits are set, USB communication may not work. Hence, these bits should always be maintained as '0'.

### 29.1.1 STANDARD INSTRUCTION SET

**ADDLW** **ADD Literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: $(W) + k \rightarrow W$

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

Encoding:

| 0000 | 1111 | kkkk | kkkk |
|---|---|---|---|

Description: The contents of W are added to the 8-bit literal ‘k’ and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal ‘k’ | Process Data | Write to W |

Example: ADDLW 15h

Before Instruction
W = 10h
After Instruction
W = 25h

**ADDWF** **ADD W to f**

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

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

Operation: $(W) + (f) \rightarrow$ dest

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

Encoding:

| 0010 | 01da | ffff | ffff |
|---|---|---|---|

Description: Add W to 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 29.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register ‘f’ | Process Data | Write to destination |

Example: ADDWF REG, 0, 0

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

**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).

## 31.2 DC Characteristics: Power-Down and Supply Current PIC18F47J53 Family (Industrial) (Continued)

| PIC18LF47J53 Family | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18F47J53 Family** | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **Param. No.** | **Device** | **Typ.** | **Max.** | **Units** | **Conditions** | | |
| | PIC18LFXXJ53 | 0.18 | 0.70 | mA | -40°C | VDD = 2.0V, VDDCORE = 2.0V | FOSC = 4 MHz **PRI_IDLE** mode, EC Oscillator |
| | | 0.18 | 0.70 | mA | +25°C | | |
| | | 0.19 | 0.75 | mA | +85°C | | |
| | PIC18LFXXJ53 | 0.23 | 0.90 | mA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | |
| | | 0.23 | 0.90 | mA | +25°C | | |
| | | 0.24 | 1.00 | mA | +85°C | | |
| | PIC18FXXJ53 | 0.23 | 0.80 | mA | -40°C | VDD = 2.15V, VDDCORE = 10 μF Capacitor | |
| | | 0.24 | 0.80 | mA | +25°C | | |
| | | 0.25 | 0.85 | mA | +85°C | | |
| | PIC18FXXJ53 | 0.32 | 1.00 | mA | -40°C | VDD = 3.3V, VDDCORE = 10 μF Capacitor | |
| | | 0.31 | 1.00 | mA | +25°C | | |
| | | 0.32 | 1.00 | mA | +85°C | | |
| | PIC18LFXXJ53 | 2.74 | 7.00 | mA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 48 MHz **PRI_IDLE** mode, EC Oscillator |
| | | 2.77 | 6.50 | mA | +25°C | | |
| | | 2.80 | 6.50 | mA | +85°C | | |
| | PIC18FXXJ53 | 3.48 | 11.00 | mA | -40°C | VDD = 3.3V, VDDCORE = 10 μF Capacitor | |
| | | 3.42 | 10.00 | mA | +25°C | | |
| | | 3.44 | 10.00 | mA | +85°C | | |

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (USB module, WDT, etc.). The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD/VSS;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

**3:** Low-power Timer1 with standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

**4:** This is the module differential current when the USB module is enabled and clocked at 48 MHz, but with no USB cable attached. When the USB cable is attached or data is being transmitted, the current consumption may be much higher (see **Section 23.6.4 "USB Transceiver Current Consumption"**). During USB Suspend mode (USBEN = 1, SUSPND = 1, bus in Idle state), the USB module current will be dominated by the D+ or D- pull-up resistor. The integrated pull-up resistors use "resistor switching" according to the `resistor_ecn` supplement to the USB 2.0 Specifications, and therefore, may be as low as 900Ω during Idle conditions.

## 31.2 DC Characteristics: Power-Down and Supply Current PIC18F47J53 Family (Industrial) (Continued)

| PIC18LF47J53 Family | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18F47J53 Family** | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **Param. No.** | **Device** | **Typ.** | **Max.** | **Units** | **Conditions** | | |
| | PIC18LFXXJ53 | 9 | 45 | µA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 32 kHz(3) **SEC_RUN** mode, (SOSCSEL<1:0> = 01) |
| | | 9 | 45 | µA | +25°C | | |
| | | 12 | 61 | µA | +85°C | | |
| | PIC18FXXJ53 | 24 | 95 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | |
| | | 28 | 95 | µA | +25°C | | |
| | | 35 | 105 | µA | +85°C | | |
| | PIC18FXXJ53 | 27 | 110 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | |
| | | 31 | 110 | µA | +25°C | | |
| | | 35 | 150 | µA | +85°C | | |
| | PIC18LFXXJ53 | 2.5 | 31 | µA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 32 kHz(3) **SEC_IDLE** mode, (SOSCSEL<1:0> = 01) |
| | | 3.0 | 31 | µA | +25°C | | |
| | | 6.1 | 50 | µA | +85°C | | |
| | PIC18FXXJ53 | 19 | 87 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | |
| | | 24 | 89 | µA | +25°C | | |
| | | 31 | 97 | µA | +85°C | | |
| | PIC18FXXJ53 | 21 | 100 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | |
| | | 25 | 100 | µA | +25°C | | |
| | | 31 | 140 | µA | +85°C | | |

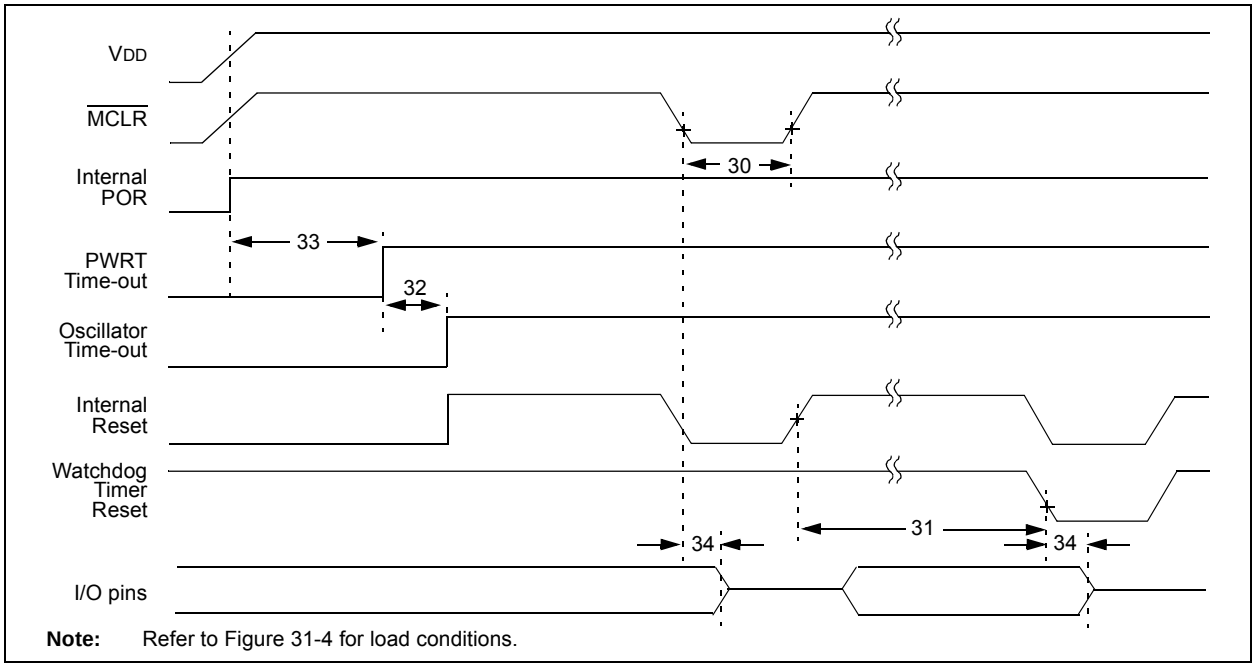
**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (USB module, WDT, etc.). The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD/VSS;
$\overline{\text{MCLR}}$ = VDD; WDT disabled unless otherwise specified.

**3:** Low-power Timer1 with standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

**4:** This is the module differential current when the USB module is enabled and clocked at 48 MHz, but with no USB cable attached. When the USB cable is attached or data is being transmitted, the current consumption may be much higher (see **Section 23.6.4 "USB Transceiver Current Consumption"**). During USB Suspend mode (USBEN = 1, SUSPND = 1, bus in Idle state), the USB module current will be dominated by the D+ or D- pull-up resistor. The integrated pull-up resistors use "resistor switching" according to the resistor_ecn supplement to the USB 2.0 Specifications, and therefore, may be as low as 900Ω during Idle conditions.

**FIGURE 31-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note:** Refer to Figure 31-4 for load conditions.

**TABLE 31-14: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

| Param. No. | Symbol | Characteristic | Min. | Typ. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2 | — | — | μs | |
| 31 | TWDT | Watchdog Timer Time-out Period (no postscaler) | 2.67 | 4.0 | 5.53 | ms | |
| 32 | TOST | Oscillator Start-up Timer Period | 1024 TOSC | — | 1024 TOSC | — | TOSC = OSC1 period |
| 33 | TPWRT | Power-up Timer Period | — | 1.0 | — | ms | |
| 34 | TIOZ | I/O High-Impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 3 TCY + 2 | μs | **(Note 1)** |
| 36 | TIRVST | Time for Internal Reference Voltage to become Stable | — | 20 | — | μs | |
| 37 | TLVD | High/Low-Voltage Detect Pulse Width | — | 200 | — | μs | |
| 38 | TCSD | CPU Start-up Time | — | 200 | — | μs | **(Note 2)** |

**Note 1:** The maximum TIOZ is the lesser of (3 TCY + 2 μs) or 700 μs.

**2:** $\overline{\text{MCLR}}$ rising edge to code execution, assuming TPWRT (and TOST if applicable) has already expired.

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.



*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

**QUALITY MANAGEMENT SYSTEM**
**CERTIFIED BY DNV**
**═ ISO/TS 16949 ═**

**Trademarks**



ISBN: 978-1-5224-0936-6