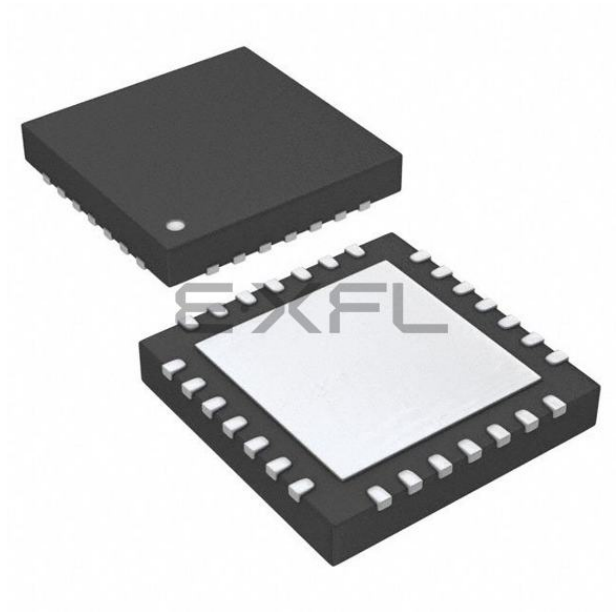


Welcome to E-XFL.COM

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)	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-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf27j53t-i-ml

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F26J53
- PIC18F27J53
- PIC18F46J53
- PIC18F47J53
- PIC18LF26J53
- PIC18LF27J53
- PIC18LF46J53
- PIC18LF47J53

This family introduces a new line of low-voltage Universal Serial Bus (USB) microcontrollers with the main traditional advantage of all PIC18 microcontrollers, namely, high computational performance and a rich feature set at an extremely competitive price point. These features make the PIC18F47J53 family a logical choice for many high-performance applications, where cost is a primary consideration.

1.1 Core Features

1.1.1 XLP TECHNOLOGY

All of the devices in the PIC18F47J53 family incorporate a range of features that can significantly reduce power consumption during operation. Key features are:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal RC oscillator, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operational requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the users to incorporate power-saving ideas into their application's software design.
- **Deep Sleep:** The 2.5V internal core voltage regulator on F parts can be shutdown to cut power consumption to as low as 15 nA (typical). Certain features can remain operating during Deep Sleep, such as the Real-Time Clock Calendar.
- **Ultra Low Power Wake-Up:** Waking from Sleep or Deep Sleep modes after a period of time can be done without an oscillator/clock source, saving power for applications requiring periodic activity.

1.1.2 UNIVERSAL SERIAL BUS (USB)

Devices in the PIC18F47J53 family incorporate a fully-featured USB communications module with a built-in transceiver that is compliant with the USB Specification Revision 2.0. The module supports both low-speed and full-speed communication for all supported data transfer types.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F47J53 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes, using crystals or ceramic resonators.
- Two External Clock modes, offering the option of a divide-by-4 clock output.
- An internal oscillator block, which provides an 8 MHz clock and an INTRC source (approximately 31 kHz, stable over temperature and V_{DD}), as well as a range of six user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of eight clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier available to the high-speed crystal, and external and internal oscillators, providing a clock speed up to 48 MHz.
- Dual clock operation, allowing the USB module to run from a high-frequency oscillator while the rest of the microcontroller is clocked at a different frequency.

The internal oscillator block provides a stable reference source that gives the PIC18F47J53 family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset (POR), or wake-up from Sleep mode, until the primary clock source is available.

1.1.4 EXPANDED MEMORY

The PIC18F47J53 family provides ample room for application code, from 64 Kbytes to 128 Kbytes of code space. The Flash cells for program memory are rated to last in excess of 10000 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The Flash program memory is readable and writable during normal operation. The PIC18F47J53 family also provides plenty of room for dynamic application data with up to 3.8 Kbytes of data RAM.

REGISTER 4-6: DSWAKEL: DEEP SLEEP WAKE LOW BYTE REGISTER (BANKED F4Ah)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1
DSFLT	—	DSULP	DSWDT	DSRTC	DSMCLR	—	DSPOR
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 **DSFLT:** Deep Sleep Fault Detected bit
1 = A Deep Sleep Fault was detected during Deep Sleep
0 = A Deep Sleep Fault was not detected during Deep Sleep

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

bit 5 **DSULP:** Ultra Low-Power Wake-up Status bit
1 = An ultra low-power wake-up event occurred during Deep Sleep
0 = An ultra low-power wake-up event did not occur during Deep Sleep

bit 4 **DSWDT:** Deep Sleep Watchdog Timer Time-out bit
1 = The Deep Sleep Watchdog Timer timed out during Deep Sleep
0 = The Deep Sleep Watchdog Timer did not time out during Deep Sleep

bit 3 **DSRTC:** Real-Time Clock and Calendar Alarm bit
1 = The Real-Time Clock/Calendar triggered an alarm during Deep Sleep
0 = The Real-Time Clock /Calendar did not trigger an alarm during Deep Sleep

bit 2 **DSMCLR:** $\overline{\text{MCLR}}$ Event bit
1 = The $\overline{\text{MCLR}}$ pin was asserted during Deep Sleep
0 = The $\overline{\text{MCLR}}$ pin was not asserted during Deep Sleep

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

bit 0 **DSPOR:** Power-on Reset Event bit
1 = The V_{DD} supply POR circuit was active and a POR event was detected(1)
0 = The V_{DD} supply POR circuit was not active, or was active, but did not detect a POR event

Note 1: Unlike the other bits in this register, this bit can be set outside of Deep Sleep.

5.0 RESET

The PIC18F47J53 family of devices differentiates among various kinds of Reset:

a) Power-on Reset (POR)
b) $\overline{\text{MCLR}}$ Reset during normal operation
c) $\overline{\text{MCLR}}$ Reset during power-managed modes
d) Watchdog Timer (WDT) Reset (during execution)
e) Configuration Mismatch (CM)
f) Brown-out Reset (BOR)
g) RESET Instruction
h) Stack Full Reset
i) Stack Underflow Reset
j) Deep Sleep Reset

This section discusses Resets generated by $\overline{\text{MCLR}}$, POR and BOR, and covers the operation of the various start-up timers.

For information on WDT Resets, see **Section 28.2 "Watchdog Timer (WDT)"**. For Stack Reset events, see **Section 6.1.4.4 "Stack Full and Underflow Resets"** and for Deep Sleep mode, see **Section 4.6 "Deep Sleep Mode"**.

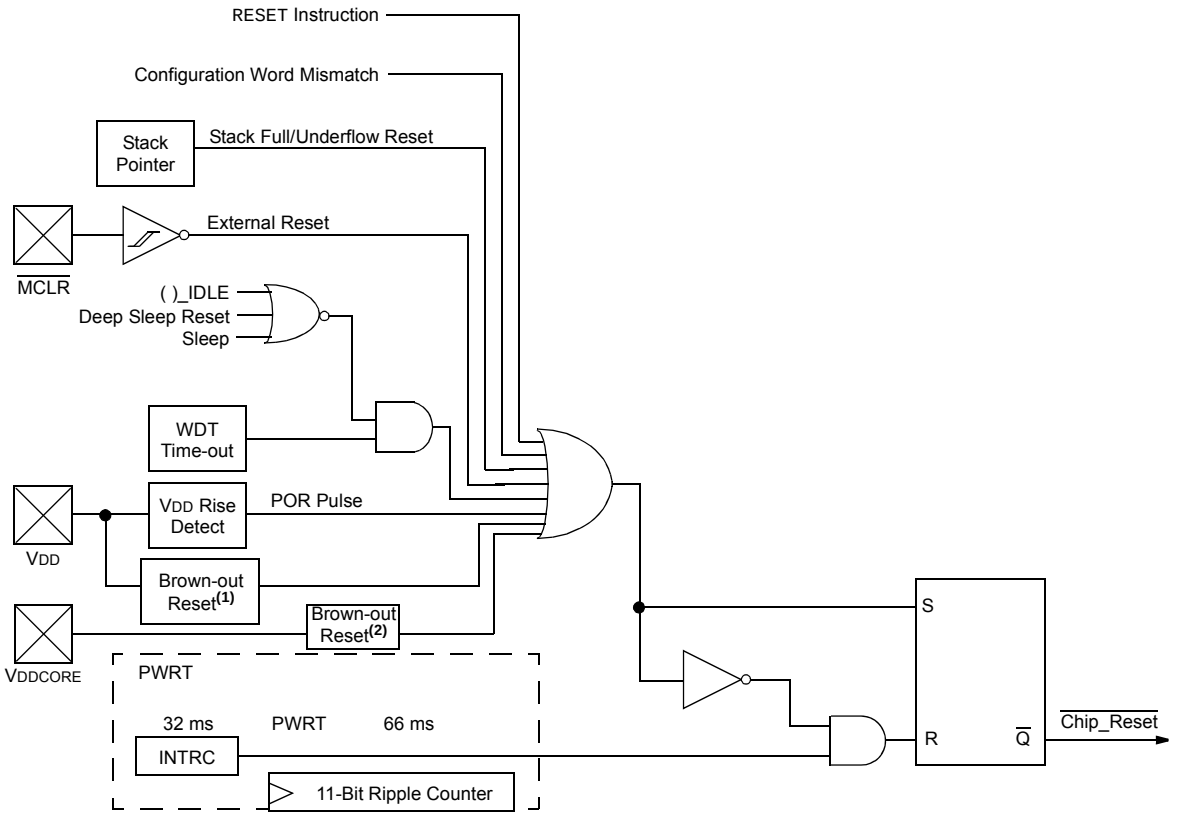
Figure 5-1 provides a simplified block diagram of the on-chip Reset circuit.

5.1 RCON Register

Device Reset events are tracked through the RCON register (Register 5-1). The lower five bits of the register indicate that a specific Reset event has occurred. In most cases, these bits can only be set by the event and must be cleared by the application after the event. The state of these flag bits, taken together, can be read to indicate the type of Reset that just occurred. This is described in more detail in **Section 5.7 "Reset State of Registers"**.

The RCON register also has a control bit for setting interrupt priority (IPEN). Interrupt priority is discussed in **Section 9.0 "Interrupts"**.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: The VDD monitoring BOR circuit can be enabled or disabled on "LF" devices based on the CONFIG3L<DSBOREN> Configuration bit. On "F" devices, the VDD monitoring BOR circuit is only enabled during Deep Sleep mode by CONFIG3L<DSBOREN>.

2: The VDDCORE monitoring BOR circuit is only implemented on "F" devices. It is always used, except while in Deep Sleep mode. The VDDCORE monitoring BOR circuit has a trip point threshold of VBOR (parameter D005).

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
TMR3H	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu
TMR4	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu
PR4	PIC18F2XJ53	PIC18F4XJ53	1111 1111	1111 1111	uuuu uuuu
T4CON	PIC18F2XJ53	PIC18F4XJ53	-000 0000	-000 0000	-uuu uuuu
SSP2BUF	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
SSP2MSK	PIC18F2XJ53	PIC18F4XJ53	---- ----	0000 0000	uuuu uuuu
SSP2STAT	PIC18F2XJ53	PIC18F4XJ53	1111 1111	1111 1111	uuuu uuuu
SSP2CON1	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XJ53	PIC18F4XJ53	---- --11	---- --11	---- --uu
PMADDRH(5)	PIC18F2XJ53	PIC18F4XJ53	-000 0000	-000 0000	-uuu uuuu
PMDOUT1H(5)	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMADDRL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
TXADDRL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
TXADDRH	PIC18F2XJ53	PIC18F4XJ53	---- 0000	---- 0000	---- uuuu
RXADDRL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
RXADDRH	PIC18F2XJ53	PIC18F4XJ53	---- 0000	---- 0000	---- uuuu
DMABCL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
DMABCH	PIC18F2XJ53	PIC18F4XJ53	---- --00	---- --00	---- --uu
UCON	PIC18F2XJ53	PIC18F4XJ53	-0x0 000-	-0x0 000-	-uuu uuu-
USTAT	PIC18F2XJ53	PIC18F4XJ53	-xxx xxx-	-xxx xxx-	-uuu uuu-
UEIR	PIC18F2XJ53	PIC18F4XJ53	0--0 0000	0--0 0000	u--u uuuu
UIR	PIC18F2XJ53	PIC18F4XJ53	-000 0000	-000 0000	-uuu uuuu
UFRMH	PIC18F2XJ53	PIC18F4XJ53	---- -xxx	---- -xxx	---- -uuu
UFRML	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	xxxxx xxxx	uuuu uuuu
PMCONH	PIC18F2XJ53	PIC18F4XJ53	0-00 0000	0-00 0000	u-uu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

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

4: See Table 5-1 for Reset value for specific condition.

5: Not implemented for PIC18F2XJ53 devices.

6: Not implemented for "LF" devices.

10.7.3.2 Output Mapping

In contrast to inputs, the outputs of the PPS options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-14).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '00000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-14: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Function	Output Function Number(1)	Output Name
NULL	0	NULL(2)
C1OUT	1	Comparator 1 Output
C2OUT	2	Comparator 2 Output
C3OUT	3	Comparator 3 Output
TX2/CK2	6	EUSART2 Asynchronous Transmit/Asynchronous Clock Output
DT2	7	EUSART2 Synchronous Transmit
SDO2	10	SPI2 Data Output
SCK2	11	SPI2 Clock Output
SSDMA	12	SPI DMA Slave Select
ULPOUT	13	Ultra Low-Power Wake-up Event
CCP1/P1A	14	ECCP1 Compare or PWM Output Channel A
P1B	15	ECCP1 Enhanced PWM Output, Channel B
P1C	16	ECCP1 Enhanced PWM Output, Channel C
P1D	17	ECCP1 Enhanced PWM Output, Channel D
CCP2/P2A	18	ECCP2 Compare or PWM Output
P2B	19	ECCP2 Enhanced PWM Output, Channel B
P2C	20	ECCP2 Enhanced PWM Output, Channel C
P2D	21	ECCP2 Enhanced PWM Output, Channel D
CCP3/P3A	22	ECCP3 Compare or PWM Output
P3B	23	ECCP3 Enhanced PWM Output, Channel B
P3C	24	ECCP3 Enhanced PWM Output, Channel C
P3D	25	ECCP3 Enhanced PWM Output, Channel D

Note 1: Value assigned to the RP<4:0> pins corresponds to the peripheral output function number.

2: The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

REGISTER 11-3: PMMODEH: PARALLEL PORT MODE REGISTER HIGH BYTE (BANKED F5Dh)(1)

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
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 **BUSY:** Busy bit (Master mode only)
1 = Port is busy
0 = Port is not busy

bit 6-5 **IRQM<1:0>:** Interrupt Request Mode bits
11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = No interrupt generated, processor stall activated
01 = Interrupt generated at the end of the read/write cycle
00 = No interrupt generated

bit 4-3 **INCM<1:0>:** Increment Mode bits
11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
10 = Decrement ADDR<15,13:0> by 1 every read/write cycle
01 = Increment ADDR<15,13:0> by 1 every read/write cycle
00 = No increment or decrement of address

bit 2 **MODE16:** 8/16-Bit Mode bit
1 = 16-bit mode: Data register is 16 bits; a read or write to the Data register invokes two 8-bit transfers
0 = 8-bit mode: Data register is 8 bits; a read or write to the Data register invokes one 8-bit transfer

bit 1-0 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master Mode 1 (PMCSx, PMRD/PMWR, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
10 = Master Mode 2 (PMCSx, PMRD, PMWR, PMBE, PMA<x:0> and PMD<7:0>)
01 = Enhanced PSP, control signals (PMRD, PMWR, PMCSx, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port, control signals (PMRD, PMWR, PMCSx and PMD<7:0>)

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

NOTES:

13.7 Resetting Timer1 Using the ECCP Special Event Trigger

If ECCP1 or ECCP2 is configured to use Timer1 and to generate a Special Event Trigger in Compare mode (CCPxM<3:0> = 1011), this signal will reset Timer3. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 19.3.4 "Special Event Trigger"** for more information).

The module must be configured as either a timer or a synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a period register for Timer1.

If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a Special Event Trigger, the write operation will take precedence.

Note:	The Special Event Trigger from the ECCPx module will not set the TMR1IF interrupt flag bit (PIR1<0>).

13.8 Timer1 Gate

The Timer1 can be configured to count freely or the count can be enabled and disabled using the Timer1 gate circuitry. This is also referred to as Timer1 gate count enable.

The Timer1 gate can also be driven by multiple selectable sources.

13.8.1 TIMER1 GATE COUNT ENABLE

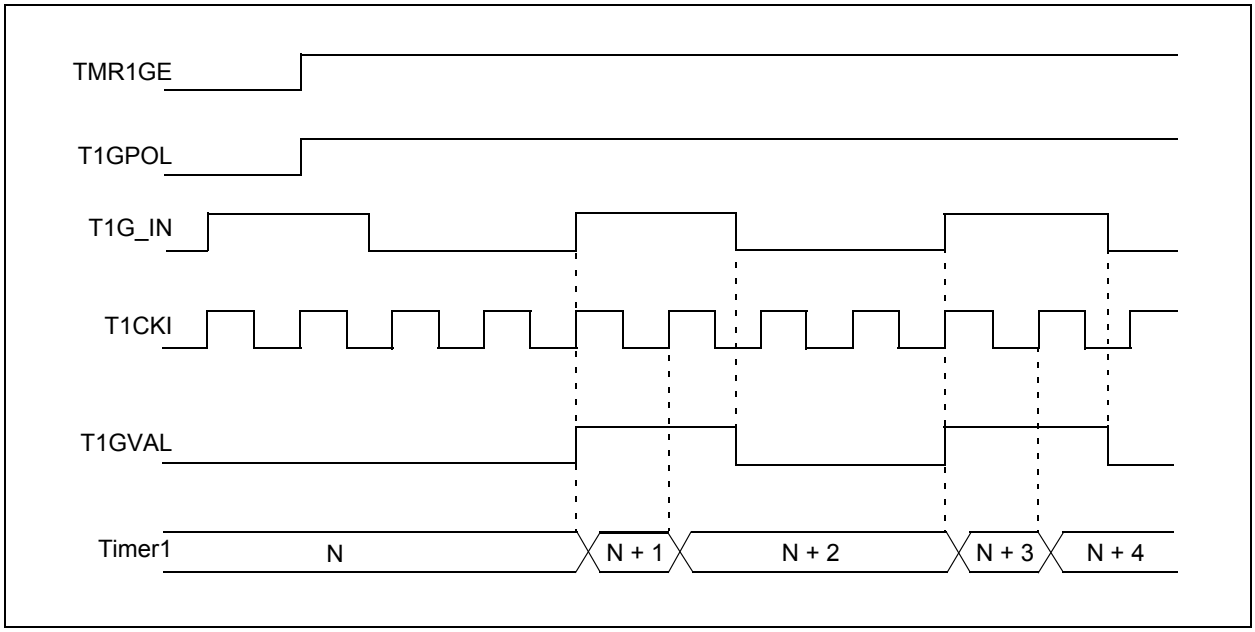
The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 13-4 for timing details.

TABLE 13-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

FIGURE 13-4: TIMER1 GATE COUNT ENABLE MODE



19.2.2 TIMER1/TIMER3 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 and/or Timer3) 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 ECCP module is selected in the CCPTMRS0 register (Register 19-2).

19.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCPxIF, should also be cleared following any such change in operating mode.

19.2.4 ECCP PRESCALER

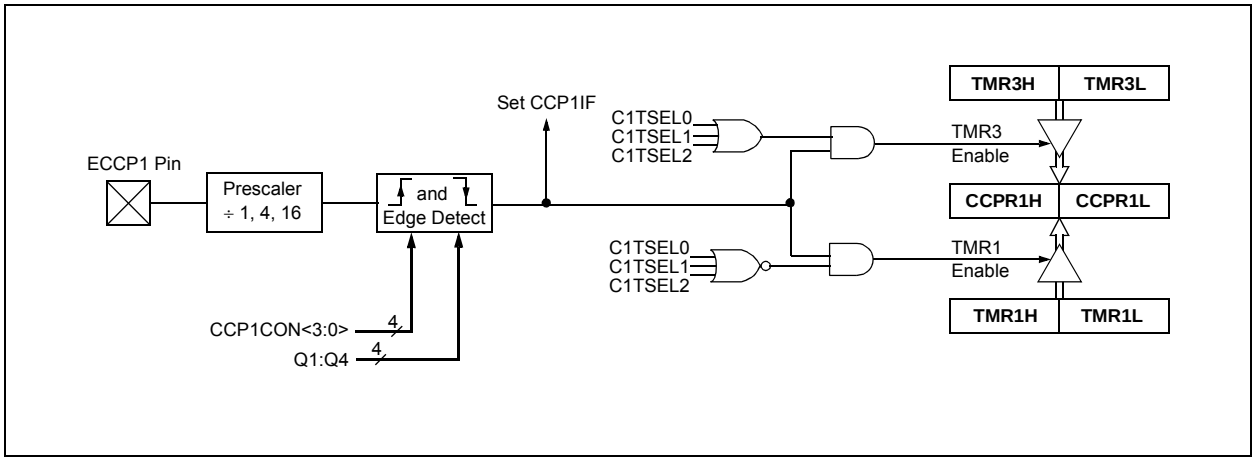
There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCPxM<3:0>). Whenever the ECCP module is turned off, or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 19-1 provides the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 19-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   ECCP1CON    ; Turn ECCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                   ; new prescaler mode
                   ; value and ECCP ON
MOVWF  CCP1CON     ; Load ECCP1CON with
                   ; this value
```

FIGURE 19-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



20.4.4.3 DMABCH and DMABCL

The DMABCH and DMABCL register pair forms a 10-bit Byte Count register, which is used by the SPI DMA module to send/receive up to 1,024 bytes for each DMA transaction. When the DMA module is actively running (DMAEN = 1), the DMA Byte Count register decrements after each byte is transmitted/received. The DMA transaction will halt and the DMAEN bit will be automatically cleared by hardware after the last byte has completed. After a DMA transaction is complete, the DMABC register will read 0x000.

Prior to initiating a DMA transaction by setting the DMAEN bit, user firmware should load the appropriate value into the DMABCH/DMABCL registers. The DMABC is a "base zero" counter, so the actual number of bytes which will be transmitted follows in Equation 20-1.

For example, if user firmware wants to transmit 7 bytes in one transaction, DMABC should be loaded with 006h. Similarly, if user firmware wishes to transmit 1,024 bytes, DMABC should be loaded with 3FFh.

EQUATION 20-1: BYTES TRANSMITTED FOR A GIVEN DMABC

$$Bytes_{XMIT} \equiv (DMABC + 1)$$

20.4.4.4 TXADDRH and TXADDRL

The TXADDRH and TXADDRL registers pair together to form a 12-bit Transmit Source Address Pointer register. In modes that use TXADDR (Full-Duplex and Half-Duplex Transmit), the TXADDR will be incremented after each byte is transmitted. Transmitted data bytes will be taken from the memory location pointed to by the TXADDR register. The contents of the memory locations pointed to by TXADDR will not be modified by the DMA module during a transmission.

The SPI DMA module can read from, and transmit data from, all general purpose memory on the device, including memory used for USB endpoint buffers. The SPI DMA module cannot be used to read from the Special Function Registers (SFRs) contained in Banks 14 and 15.

20.4.4.5 RXADDRH and RXADDRL

The RXADDRH and RXADDRL registers pair together to form a 12-bit Receive Destination Address Pointer. In modes that use RXADDR (Full-Duplex and Half-Duplex Receive), the RXADDR register will be incremented after each byte is received. Received data bytes will be stored at the memory location pointed to by the RXADDR register.

The SPI DMA module can write received data to all general purpose memory on the device, including memory used for USB endpoint buffers. The SPI DMA module cannot be used to modify the Special Function Registers contained in Banks 14 and 15.

20.4.5 INTERRUPTS

The SPI DMA module alters the behavior of the SSP2IF interrupt flag. In normal non-DMA modes, the SSP2IF is set once after every single byte is transmitted/received through the MSSP2 module. When MSSP2 is used with the SPI DMA module, the SSP2IF interrupt flag will be set according to the user-selected INTLVL<3:0> value specified in the DMACON2 register. The SSP2IF interrupt condition will also be generated once the SPI DMA transaction has fully completed and the DMAEN bit has been cleared by hardware.

The SSP2IF flag becomes set once the DMA byte count value indicates that the specified INTLVL has been reached. For example, if DMACON2<3:0> = 0101 (16 bytes remaining), the SSP2IF interrupt flag will become set once DMABC reaches 00Fh. If user firmware then clears the SSP2IF interrupt flag, the flag will not be set again by the hardware until after all bytes have been fully transmitted and the DMA transaction is complete.

Note:	User firmware may modify the INTLVL bits while a DMA transaction is in progress (DMAEN = 1). If an INTLVL value is selected which is higher than the actual remaining number of bytes (indicated by DMABC + 1), the SSP2IF interrupt flag will immediately become set.

For example, if DMABC = 00Fh (implying 16 bytes are remaining) and user firmware writes '1111' to INTLVL<3:0> (interrupt when 576 bytes remaining), the SSP2IF interrupt flag will immediately become set. If user firmware clears this interrupt flag, a new interrupt condition will not be generated until either: user firmware again writes INTLVL with an interrupt level higher than the actual remaining level, or the DMA transaction completes and the DMAEN bit is cleared.

Note:	If the INTLVL bits are modified while a DMA transaction is in progress, care should be taken to avoid inadvertently changing the DLYCYC<3:0> value.

23.2.5 USB ADDRESS REGISTER (UADDR)

The USB Address register contains the unique USB address that the peripheral will decode when active. UADDR is reset to 00h when a USB Reset is received, indicated by URSTIF, or when a Reset is received from the microcontroller. The USB address must be written by the microcontroller during the USB setup phase (enumeration) as part of the Microchip USB firmware support.

23.2.6 USB FRAME NUMBER REGISTERS (UFRMH:UFRML)

The Frame Number registers contain the 11-bit frame number. The low-order byte is contained in UFRML, while the three high-order bits are contained in UFRMH. The register pair is updated with the current frame number whenever a SOF token is received. For the microcontroller, these registers are read-only. The Frame Number registers are primarily used for isochronous transfers. The contents of the UFRMH and UFRML registers are only valid when the 48 MHz SIE clock is active (i.e., contents are inaccurate when SUSPND (UCON<1>) bit = 1).

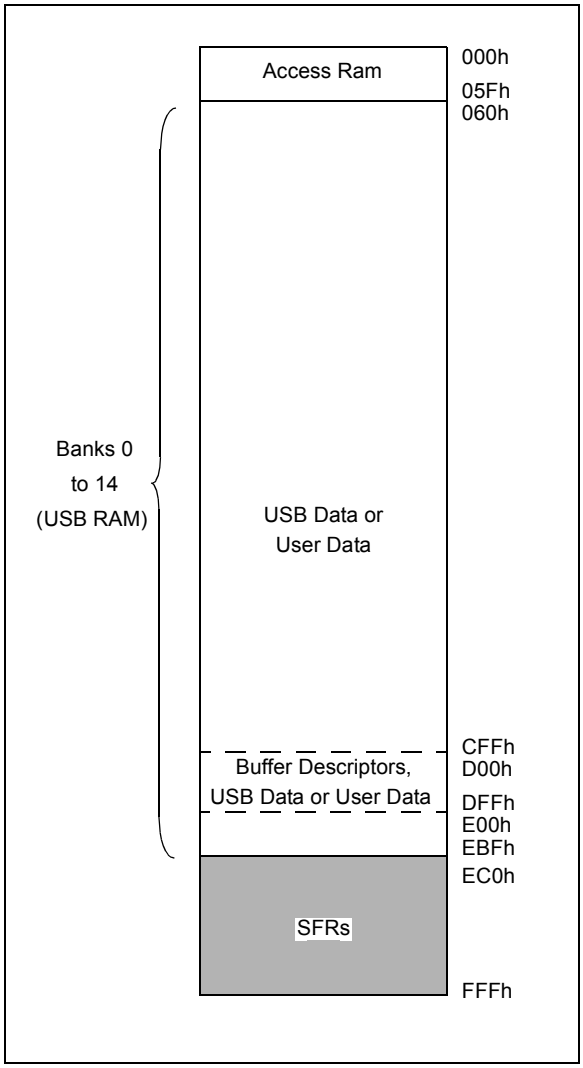
23.3 USB RAM

USB data moves between the microcontroller core and the SIE through a memory space known as the USB RAM. This is a special dual access memory that is mapped into the normal data memory space in Banks, 0 through 14 (00h to EBFh), for a total of 3.8 Kbytes (Figure 23-4).

Bank 13 (D00h through DFFh) is used specifically for endpoint buffer control, while Banks 0 through 12 and Bank 14 are available for USB data. Depending on the type of buffering being used, all but 8 bytes of Bank 13 may also be available for use as USB buffer space.

Although USB RAM is available to the microcontroller as data memory, the sections that are being accessed by the SIE should not be accessed by the microcontroller. A semaphore mechanism is used to determine the access to a particular buffer at any given time. This is discussed in **Section 23.4.1.1 "Buffer Ownership"**.

FIGURE 23-4: IMPLEMENTATION OF USB RAM IN DATA MEMORY SPACE



The BDnSTAT byte of the BDT should always be the last byte updated when preparing to arm an endpoint. The SIE will clear the UOWN bit when a transaction has completed.

No hardware mechanism exists to block access when the UOWN bit is set. Thus, unexpected behavior can occur if the microcontroller attempts to modify memory when the SIE owns it. Similarly, reading such memory may produce inaccurate data until the USB peripheral returns ownership to the microcontroller.

23.4.1.2 BDnSTAT Register (CPU Mode)

When UOWN = 0, the microcontroller core owns the BD. At this point, the other seven bits of the register take on control functions.

The Data Toggle Sync Enable bit, DTSEN (BDnSTAT<3>), controls data toggle parity checking. Setting DTSEN enables data toggle synchronization by the SIE. When enabled, it checks the data packet's parity against the value of DTS (BDnSTAT<6>). If a packet arrives with an incorrect synchronization, the data will essentially be ignored. It will not be written to the USB RAM and the USB transfer complete interrupt flag will not be set. The SIE will send an ACK token back to the host to Acknowledge receipt, however. The effects of the DTSEN bit on the SIE are summarized in Table 23-1.

The Buffer Stall bit, BSTALL (BDnSTAT<2>), provides support for control transfers, usually one-time stalls on Endpoint 0. It also provides support for the SET_FEATURE/CLEAR_FEATURE commands specified in Chapter 9 of the USB Specification; typically, continuous STALLs to any endpoint other than the default control endpoint.

The BSTALL bit enables buffer stalls. Setting BSTALL causes the SIE to return a STALL token to the host if a received token would use the BD in that location. The EPSTALL bit in the corresponding UEPn Control register is set and a STALL interrupt is generated when a STALL is issued to the host. The UOWN bit remains set and the BDs are not changed unless a SETUP token is received. In this case, the STALL condition is cleared and the ownership of the BD is returned to the microcontroller core.

The BCx<9:8> bits (BDnSTAT<1:0>) store the two most significant digits of the SIE byte count. The lower 8 digits are stored in the corresponding BDnCNT register. See **Section 23.4.2 "BD Byte Count"** for more information.

TABLE 23-1: EFFECT OF DTSEN BIT ON ODD/EVEN (DATA0/DATA1) PACKET RECEPTION

OUT Packet from Host	BDnSTAT Settings		Device Response after Receiving Packet			
	DTSEN	DTS	Handshake	UOWN	TRNIF	BDnSTAT and USTAT Status
DATA0	1	0	ACK	0	1	Updated
DATA1	1	0	ACK	1	0	Not Updated
DATA0	1	1	ACK	1	0	Not Updated
DATA1	1	1	ACK	0	1	Updated
Either	0	x	ACK	0	1	Updated
Either, with error	x	x	NAK	1	0	Not Updated

Legend: x = don't care

23.5.2 USB INTERRUPT ENABLE REGISTER (UIE)

The USB Interrupt Enable (UIE) register (Register 23-8) contains the enable bits for the USB status interrupt sources. Setting any of these bits will enable the respective interrupt source in the UIR register.

The values in this register only affect the propagation of an interrupt condition to the microcontroller's interrupt logic. The flag bits are still set by their interrupt conditions, allowing them to be polled and serviced without actually generating an interrupt.

REGISTER 23-8: UIE: USB INTERRUPT ENABLE REGISTER (BANKED F36h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
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 **SOFIE:** Start-Of-Frame Token Interrupt Enable bit
1 = Start-Of-Frame token interrupt is enabled
0 = Start-Of-Frame token interrupt is disabled

bit 5 **STALLIE:** STALL Handshake Interrupt Enable bit
1 = STALL interrupt is enabled
0 = STALL interrupt is disabled

bit 4 **IDLEIE:** Idle Detect Interrupt Enable bit
1 = Idle detect interrupt is enabled
0 = Idle detect interrupt is disabled

bit 3 **TRNIE:** Transaction Complete Interrupt Enable bit
1 = Transaction interrupt is enabled
0 = Transaction interrupt is disabled

bit 2 **ACTVIE:** Bus Activity Detect Interrupt Enable bit
1 = Bus activity detect interrupt is enabled
0 = Bus activity detect interrupt is disabled

bit 1 **UERRIE:** USB Error Interrupt Enable bit
1 = USB error interrupt is enabled
0 = USB error interrupt is disabled

bit 0 **URSTIE:** USB Reset Interrupt Enable bit
1 = USB Reset interrupt is enabled
0 = USB Reset interrupt is disabled

23.7 Oscillator

The USB module has specific clock requirements. For full-speed operation, the clock source must be 48 MHz. Even so, the microcontroller core and other peripherals are not required to run at that clock speed. Available clocking options are described in detail in **Section 3.3 "Oscillator Settings for USB"**.

23.8 USB Firmware and Drivers

Microchip provides a number of application-specific resources, such as USB firmware and driver support. Refer to www.microchip.com for the latest firmware and driver support.

TABLE 23-4: REGISTERS ASSOCIATED WITH USB MODULE OPERATION(1)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the USB module.

Note 1: This table includes only those hardware mapped SFRs located in Bank 15 of the data memory space. The Buffer Descriptor registers, which are mapped into Bank 4 and are not true SFRs, are listed separately in Table 23-3.

SLEEP **Enter Sleep Mode**

Syntax: SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT postscaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

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

Encoding:

0000	0000	0000	0011

Description: The Power-Down status bit ($\overline{PD}$) is cleared. The Time-out status bit ($\overline{TO}$) is set. The Watchdog Timer and its postscaler are cleared.

The processor is put into Sleep mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to Sleep

Example: SLEEP

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?
After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared.

SUBFWB **Subtract f from W with Borrow**

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

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (W) – (f) – ($\overline{C}$) → dest

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

Encoding:

0101	01da	ffff	ffff

Description: Subtract register ‘f’ and Carry flag (borrow) from W (2’s complement method). If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored 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 ≤ 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 1: SUBFWB REG, 1, 0

Before Instruction
REG = 3
W = 2
C = 1
After Instruction
REG = FF
W = 2
C = 0
Z = 0
N = 1 ; result is negative

Example 2: SUBFWB REG, 0, 0

Before Instruction
REG = 2
W = 5
C = 1
After Instruction
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; result is positive

Example 3: SUBFWB REG, 1, 0

Before Instruction
REG = 1
W = 2
C = 0
After Instruction
REG = 0
W = 2
C = 1
Z = 1 ; result is zero
N = 0

MOVSS **Move Indexed to Indexed**

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: MOVSS [05h], [06h]

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 33h

PUSHL **Store Literal at FSR2, Decrement FSR2**

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: k → (FSR2),
FSR2 – 1 → FSR2

Status Affected: None

Encoding:

1111	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: PUSHL 08h

Before Instruction
FSR2H:FSR2L = 01ECh
Memory (01ECh) = 00h

After Instruction
FSR2H:FSR2L = 01EBh
Memory (01ECh) = 08h

SUBFSR **Subtract Literal from FSR**

Syntax: SUBFSR f, k

Operands: $0 \le k \le 63$

$f \in [0, 1, 2]$

Operation: FSRf – k → FSRf

Status Affected: None

Encoding:

1110	1001	ffkk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: SUBFSR 2, 23h

Before Instruction
FSR2 = 03FFh

After Instruction
FSR2 = 03DCh

SUBULNK **Subtract Literal from FSR2 and Return**

Syntax: SUBULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 – k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1001	11kk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR2. A RETURN is then executed by loading the PC with the TOS.

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

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

Words: 1

Cycles: 2

Q Cycle Activity:

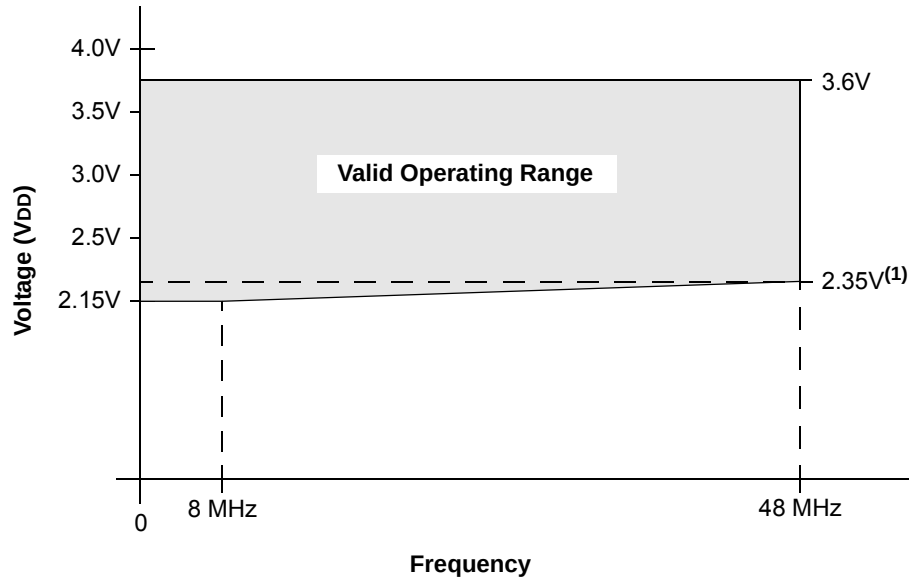
Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: SUBULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h

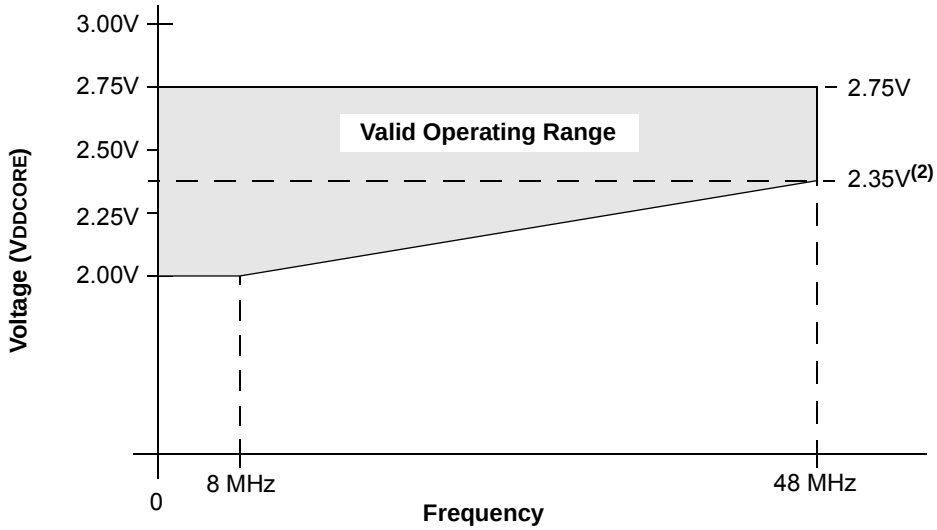
After Instruction
FSR2 = 03DCh
PC = (TOS)

FIGURE 31-1: PIC18F47J53 FAMILY VDD FREQUENCY GRAPH (INDUSTRIAL)



Note 1: When the USB module is enabled, VUSB should be provided 3.0V-3.6V while VDD must be ≥2.35V. When the USB module is not enabled, the wider limits shaded in grey apply. VUSB should be maintained ≥ VDD, but may optionally be high-impedance when the USB module is not in use.

FIGURE 31-2: PIC18LF47J53 FAMILY VDDCORE FREQUENCY GRAPH (INDUSTRIAL)(1)

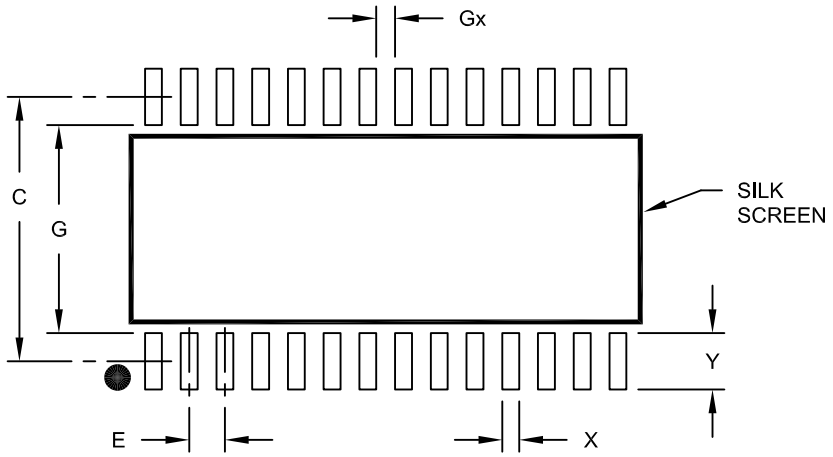


Note 1: VDD and VDDCORE must be maintained so that VDDCORE ≤ VDD.

2: When the USB module is enabled, VUSB should be provided 3.0V-3.6V while VDDCORE must be ≥2.35V. When the USB module is not enabled, the wider limits shaded in grey apply. VUSB should be maintained ≥ VDD, but may optionally be high-impedance when the USB module is not in use.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

APPENDIX B: MIGRATION FROM PIC18F46J50 TO PIC18F47J53

Code for the devices in the PIC18F46J50 family can be migrated to the PIC18F47J53 without many changes. The differences between the two device families are listed in Table B-1.

TABLE B-1: NOTABLE DIFFERENCES BETWEEN PIC18F47J53 AND PIC18F46J50 FAMILIES

Characteristic	PIC18F47J53 Family	PIC18F46J50 Family
Max Program Memory	128 Kbytes	64 Kbytes
Oscillator options	PLL can be enabled at start-up with Config bit option	Requires firmware to set the PLLEN bit at run time
SOSC Oscillator Options	Low-power oscillator option for SOSC, with run-time switch	Low-power oscillator option for SOSC, only via Configuration bit setting
T1CKI Clock Input	T1CKI can be used as a clock input without enabling the Timer1 oscillator	No
INTOSC	Up to 8 MHz	Up to 8 MHz
Timers	8	5
ECCP	3	2
CCP	7	0
SPI Fosc/8 Master Clock Option	Yes	No
ADC	13 Channel, 10/12-bit conversion modes with Special Event Trigger option.	13 Channel, 10-bit only
Peripheral Module Disable Bits	Yes, allowing further power reduction	No
Band Gap Voltage Reference Output	Yes, enabled on pin RA1 by setting the VBGOE bit (WDTCON<4>)	No
REPU/RDPU Pull-Up Enable Bits	Moved to TRISE register (avoids read, modify, write issues)	Pull-up bits configured in PORTE register
Comparators	Three, each with four input-pin selections	Two, each with two input-pin selections
Increased Output Drive Strength	RA0 through RA5, RDx and REx	No