Microchip Technology - PIC18F26J53-I/SS Datasheet

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	64KB (32K 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/pic18f26j53-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			
					PORTB (continued)
RB4/CCP4/KBI0/SCK1/SCL1/ RP7	25(2)	22(2)			
RB4			I/O	TTL/DIG	Digital I/O.
CCP4			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI0			I	TTL	Interrupt-on-change pin.
SCK1			I/O	ST/DIG	Synchronous serial clock input/output.
SCL1			I/O	I^{2}C	I^{2}C clock input/output.
RP7			I/O	ST/DIG	Remappable Peripheral Pin 7 input/output.
RB5/CCP5/KBI1/SDI1/SDA1/ RP8	26(2)	23(2)			
RB5			I/O	TTL/DIG	Digital I/O.
CCP5			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI1			I	TTL	Interrupt-on-change pin.
SDI1			I	ST	SPI data input.
SDA1			I/O	I^{2}C	I^{2}C data input/output.
RP8			I/O	ST/DIG	Remappable Peripheral Pin 8 input/output.
RB6/CCP6/KBI2/PGC/RP9	27(2)	24(2)			
RB6			I/O	TTL/DIG	Digital I/O.
CCP6			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI2			I	TTL	Interrupt-on-change pin.
PGC			I	ST	ICSP™ clock input.
RP9			I/O	ST/DIG	Remappable Peripheral Pin 9 input/output.
RB7/CCP7/KBI3/PGD/RP10	28(2)	25(2)			
RB7			I/O	TTL/DIG	Digital I/O.
CCP7			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI3			I	TTL	Interrupt-on-change pin.
PGD			I/O	ST/DIG	In-Circuit Debugger and ICSP programming data pin.
RP10			I/O	ST/DIG	Remappable Peripheral Pin 10 input/output.

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^{2}C	= Open-Drain, I^{2}C specific

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

2: 5.5V tolerant.

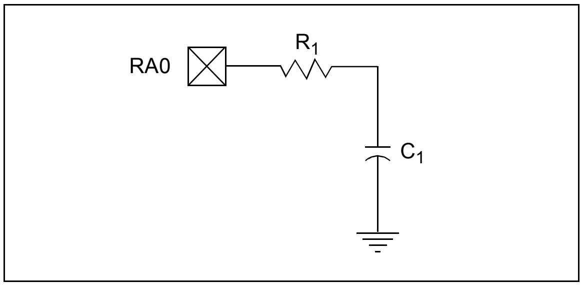
TABLE 3-5: OSCILLATOR CONFIGURATION OPTIONS FOR USB OPERATION

Input Oscillator Frequency	PLL Division (PLLDIV<2:0>)	Clock Mode (FOSC<2:0>)	MCU Clock Division (CPDIV<1:0>)	Microcontroller Clock Frequency
48 MHz	N/A	EC	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
48 MHz	÷12 (000)	ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
40 MHz	÷10 (001)	ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
24 MHz	÷6 (010)	ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
24 MHz	N/A	EC(1)	None (11)	24 MHz
			÷2 (10)	12 MHz
			÷3 (01)	8 MHz
			÷6 (00)	4 MHz
20 MHz	÷5 (011)	ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
16 MHz	÷4 (100)	HSPLL, ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
12 MHz	÷3 (101)	HSPLL, ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
8 MHz	÷2 (110)	HSPLL, ECPLL, INTOSCPLL/ INTOSCPLLO	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz
4 MHz	÷1 (111)	HSPLL, ECPLL	None (11)	48 MHz
			÷2 (10)	24 MHz
			÷3 (01)	16 MHz
			÷6 (00)	8 MHz

Note 1: The 24 MHz EC mode (without PLL) is only compatible with low-speed USB. Full-speed USB requires a 48 MHz system clock.

A series resistor between RA0 and the external capacitor provides overcurrent protection for the RA0/AN0/C1INA/ULPWU/RP0 pin and can allow for software calibration of the time-out (see Figure 4-9).

FIGURE 4-9: SERIAL RESISTOR



A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The ULPWU peripheral can also be configured as a simple Programmable Low-Voltage Detect (LVD) or temperature sensor.

> **Note:** For more information, refer to *AN879, Using the Microchip Ultra Low-Power Wake-up Module* application note (DS00879).

4.8 Peripheral Module Disable

All peripheral modules (except for I/O ports) also have a second control bit that can disable their functionality. These bits, known as the Peripheral Module Disable (PMDISx) bits, are generically named “xxxMD” (using “xxx” as the mnemonic version of the module’s name).

These bits are located in the PMDISx Special Function Registers. In contrast to the module enable bits (generically named “xxxEN” and located in bit position seven of the control registers), the PMDISx bits must be set (= `1`) to disable the modules.

While the PMD and module enable bits both disable a peripheral’s functionality, the PMD bit completely shuts down the peripheral, effectively powering down all circuits and removing all clock sources. This has the additional effect of making any of the module’s control and buffer registers, mapped in the SFR space, unavailable for operations. Essentially, the peripheral ceases to exist until the PMD bit is cleared.

This differs from using the module enable bit, which allows the peripheral to be reconfigured and buffer registers preloaded, even when the peripheral’s operations are disabled.

The PMDISx bits are most useful in highly power-sensitive applications. In these cases, the bits can be set before the main body of the application to remove peripherals that will not be needed at all.

TABLE 4-2: LOW-POWER MODE REGISTERS

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
PMDIS3	CCP10MD	CCP9MD	CCP8MD	CCP7MD	CCP6MD	CCP5MD	CCP4MD	—	`0000 000-`
PMDIS2	—	TMR8MD	—	TMR6MD	TMR5MD	CMP3MD	CMP2MD	CMP1MD	`-0-0 0000`
PMDIS1	PSPMD(1)	CTMUMD	RTCCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	—	`0000 000-`
PMDIS0	ECCP3MD	ECCP2MD	ECCP1MD	UART2MD	UART1MD	SPI2MD	SPI1MD	ADCMD	`0000 0000`

Note 1: Not implemented on 28-pin devices (PIC18F26J53, PIC18F27J53, PIC18LF26J53 and PIC18LF27J53).

9.0 INTERRUPTS

Devices of the PIC18F47J53 family have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

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

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

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

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

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

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

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

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

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

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

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

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

9.5 RCON Register

The RCON register contains bits used to determine the cause of the last Reset or wake-up from Idle or Sleep mode. RCON also contains the bit that enables interrupt priorities (IPEN).

REGISTER 9-19: RCON: RESET CONTROL REGISTER (ACCESS FD0h)

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **$\overline{\text{CM}}$:** Configuration Mismatch Flag bit
For details on bit operation, see Register 5-1.

bit 4 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
For details on bit operation, see Register 5-1.

bit 3 **$\overline{\text{TO}}$:** Watchdog Timer Time-out Flag bit
For details on bit operation, see Register 5-1.

bit 2 **$\overline{\text{PD}}$:** Power-Down Detection Flag bit
For details on bit operation, see Register 5-1.

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
For details on bit operation, see Register 5-1.

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
For details on bit operation, see Register 5-1.

11.2.5.2 WRITE TO SLAVE PORT

When chip select is active and a write strobe occurs (PMCSx = 1 and PMWR = 1), the data from PMD<7:0> is captured into one of the four input buffer bytes. Which byte is written depends on the 2-bit address placed on ADDRL<1:0>.

Table 11-1 provides the corresponding input registers and their associated address.

When an input buffer is written, the corresponding IBxF bit is set. The IBF flag bit is set when all the buffers are written. If any buffer is already written (IBxF = 1), the next write strobe to that buffer will generate an OBUF event and the byte will be discarded.

FIGURE 11-8: PARALLEL SLAVE PORT WRITE WAVEFORMS

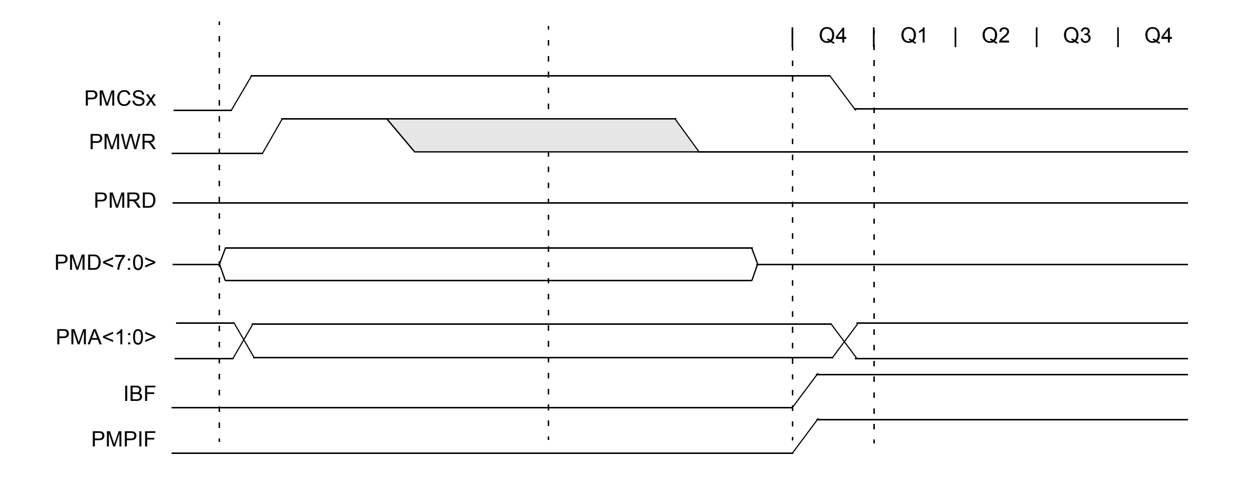


FIGURE 11-21: READ TIMING, 16-BIT DATA, DEMULTIPLEXED ADDRESS

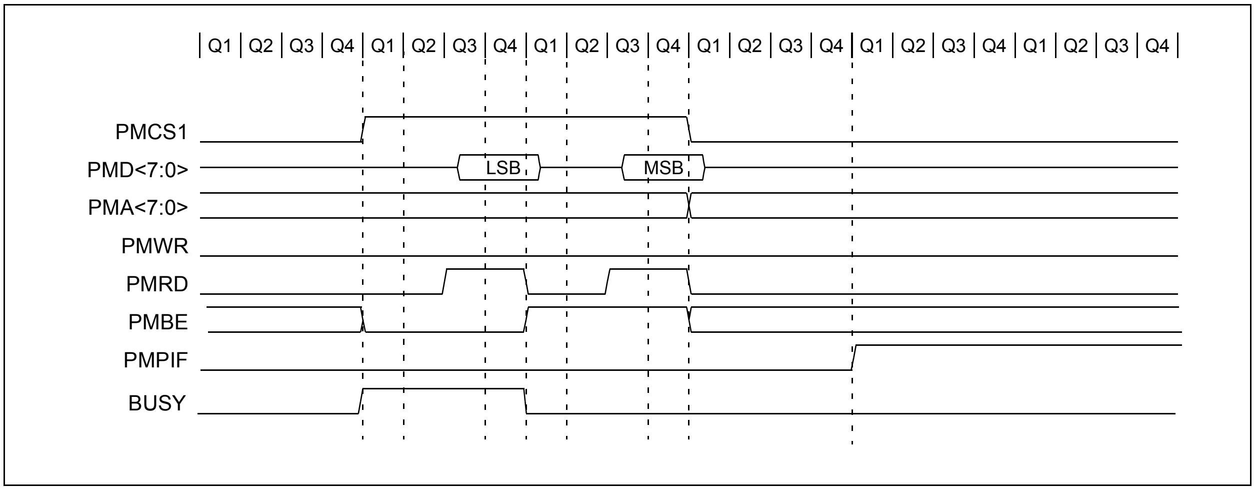


FIGURE 11-22: WRITE TIMING, 16-BIT DATA, DEMULTIPLEXED ADDRESS

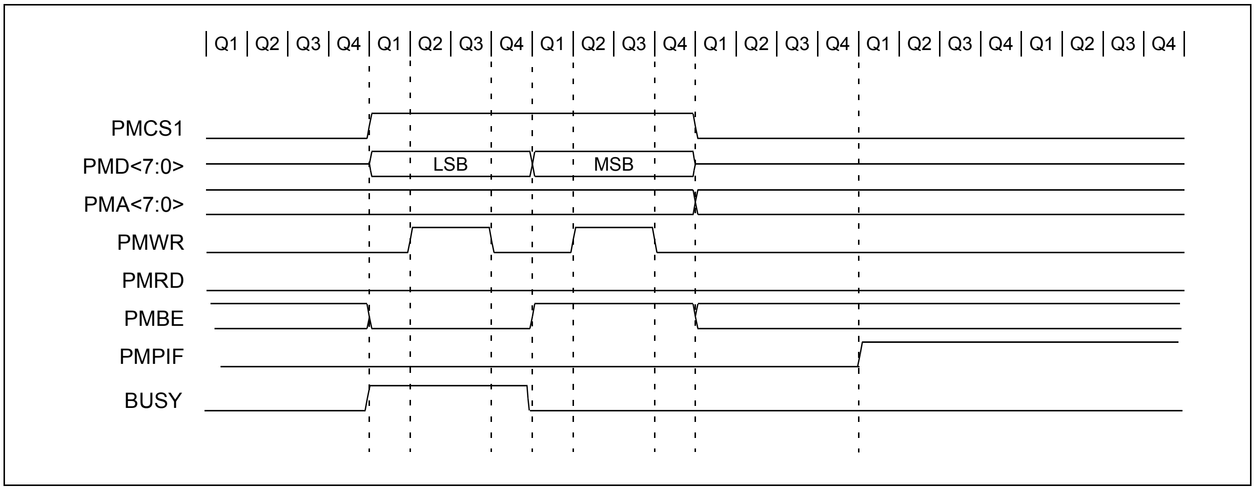
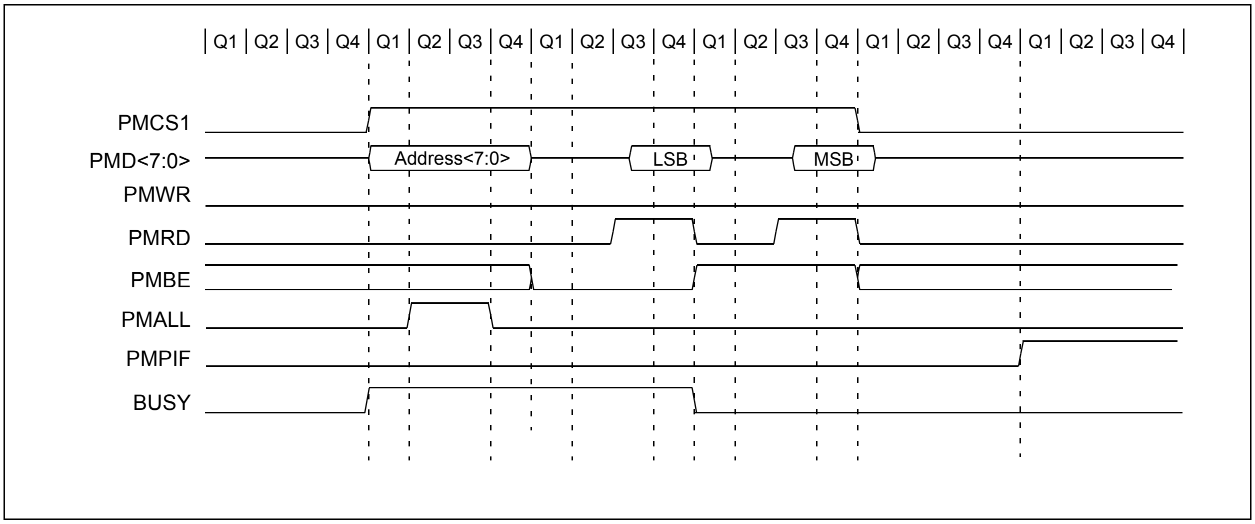


FIGURE 11-23: READ TIMING, 16-BIT MULTIPLEXED DATA, PARTIALLY MULTIPLEXED ADDRESS



13.2 Timer1 Operation

The Timer1 module is an 8-bit or 16-bit incrementing counter, which is accessed through the TMR1H:TMR1L register pair.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively.

When Timer1 is enabled, the RC1/CCP8/T1OSI/$\overline{\text{UOE}}$/RP12 and RC0/T1OSO/T1CKI/RP11 pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

13.3 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Register 13-1 displays the clock source selections.

13.3.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

13.3.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input, T1CKI, or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR Reset
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0)

when T1CKI is high, then Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

TABLE 13-1: TIMER1 CLOCK SOURCE SELECTION

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	1	x	Clock Source (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	0	0	External Clock on T1CKI Pin
1	0	1	Oscillator Circuit on T1OSI/T1OSO Pin

TABLE 15-5: REGISTERS ASSOCIATED WITH TIMER3/5 AS A TIMER/COUNTER

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
PIR5	—	—	CM3IF	TMR8IF	TMR6IF	TMR5IF	TMR5GIF	TMR1GIF
PIE5	—	—	CM3IE	TMR8IE	TMR6IE	TMR5IE	TMR5GIE	TMR1GIE
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
TMR3H	Timer3 Register High Byte							
TMR3L	Timer3 Register Low Byte							
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{T3DONE}}$	T3GVAL	T3GSS1	T3GSS0
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	$\overline{\text{T3SYNC}}$	RD16	TMR3ON
TMR5H	Timer5 Register High Byte							
TMR5L	Timer5 Register Low Byte							
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{T5DONE}}$	T5GVAL	T5GSS1	T5GSS0
T5CON	TMR5CS1	TMR5CS0	T5CKPS1	T5CKPS0	T5OSCEN	$\overline{\text{T5SYNC}}$	RD16	TMR5ON
OSCCON2	—	SOSCRUN	—	SOSCDRV	SOSCGO	PRISD	—	—
CCPTMRS0	C3TSEL1	C3TSEL0	C2TSEL2	C2TSEL1	C2TSEL0	C1TSEL2	C1TSEL1	C1TSEL0
CCPTMRS1	C7TSEL1	C7TSEL0	—	C6TSEL0	—	C5TSEL0	C4TSEL1	C4TSEL0
CCPTMRS1	—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0
CCPTMRS2	—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer3 module.

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”**.

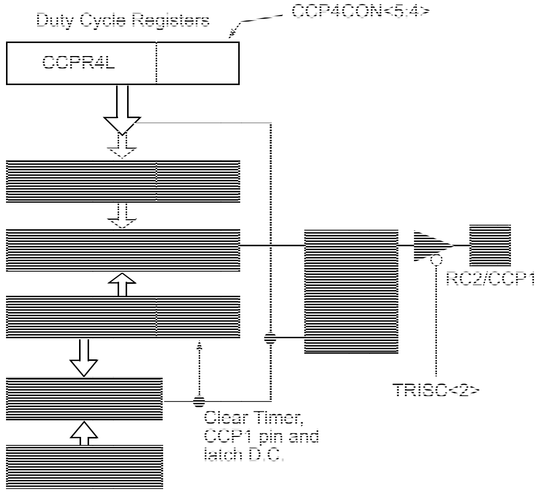
18.4 PWM Mode

In Pulse-Width Modulation (PWM) mode, the CCP4 pin produces up to a 10-bit resolution PWM output. Since the CCP4 pin is multiplexed with a PORTB data latch, the appropriate TRIS bit must be cleared to make the CCP4 pin an output.

Figure 18-3 shows a simplified block diagram of the CCP1 module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 18.4.3 "Setup for PWM Operation"**.

FIGURE 18-3: SIMPLIFIED PWM BLOCK DIAGRAM

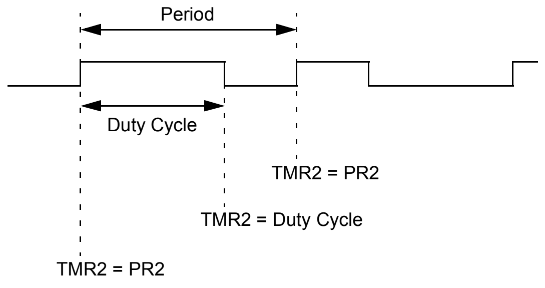


Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

2: CCP4 and its appropriate timers are used as an example. For details on all of the CCP modules and their timer assignments, see Table 18-2 and Table 18-3.

A PWM output (Figure 18-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 18-4: PWM OUTPUT



18.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 18-1:

$$\text{PWM Period} = [(\text{PR2}) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP4 pin is set
  (An exception: If the PWM duty cycle = 0%, the CCP4 pin will not be set)
- The PWM duty cycle is latched from CCPR4L into CCPR4H

> **Note:** The Timer2 postscalers (see **Section 14.0 "Timer2 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

24.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode when enabled. Each operational comparator will consume additional current. To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

24.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 24-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

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
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR5	—	—	CM3IF	TMR8IF	TMR6IF	TMR5IF	TMR5GIF	TMR1GIF
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE5	—	—	CM3IE	TMR8IE	TMR6IE	TMR5IE	TMR5GIE	TMR1GIE
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR5	—	—	CM3IP	TMR8IP	TMR6IP	TMR5IP	TMR5GIP	TMR1GIP
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
CMSTAT	—	—	—	—	—	COUT3	COUT2	COUT1
ANCON0	PCFG7(Legend:)	PCFG6(Legend:)	PCFG5(Legend:)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
ANCON1	VBGEN	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0

Legend: — = unimplemented, read as ‘0’. Shaded cells are not related to comparator operation.

27.6 Creating a Delay with the CTMU Module

A unique feature on board the CTMU module is its ability to generate system clock, independent output pulses based on an external capacitor value. This is accomplished using the internal comparator voltage reference module, Comparator 2 input pin and an external capacitor. The pulse is output onto the CTPLS pin. To enable this mode, set the TGEN bit.

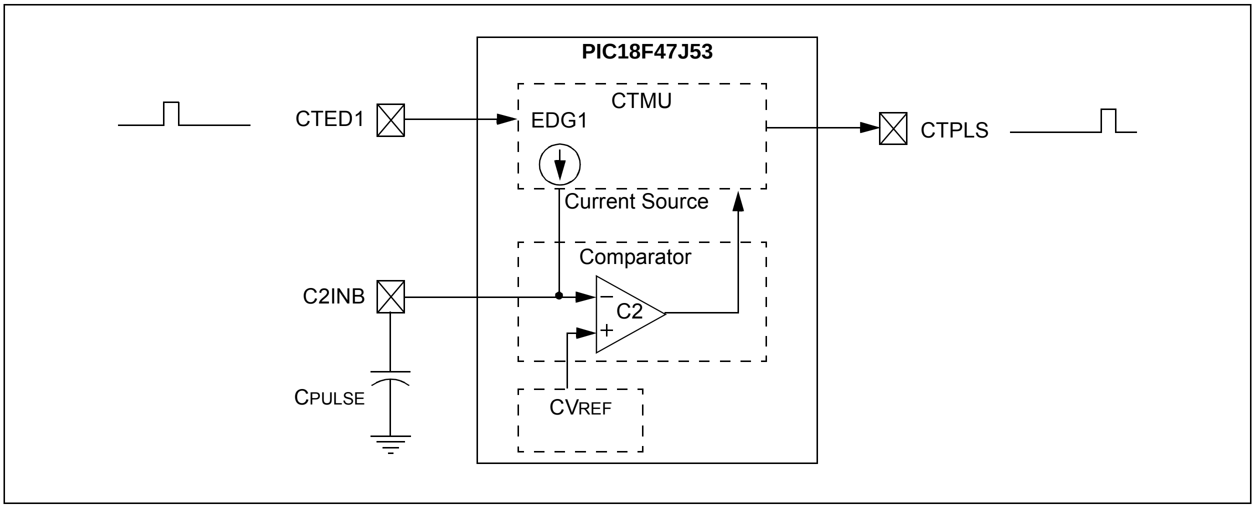
See Figure 27-4 for an example circuit. CPULSE is chosen by the user to determine the output pulse width on CTPLS. The pulse width is calculated by $T = (CPULSE/I)*V$, where I is known from the current source measurement step (**Section 27.3.1 "Current Source Calibration"**) and V is the internal reference voltage (CVREF).

An example use of this feature is for interfacing with variable capacitive-based sensors, such as a humidity sensor. As the humidity varies, the pulse width output on CTPLS will vary. The CTPLS output pin can be connected to an input capture pin and the varying pulse width is measured to determine the humidity in the application.

Follow these steps to use this feature:

1. Set the CPOL bit (CMxCON<5>).
2. Initialize the comparator voltage reference.
3. Initialize the CTMU and enable time delay generation by setting the TGEN bit.
4. Set EDG1STAT.
5. When CPULSE charges to the value of the voltage reference trip point, an output pulse is generated on CTPLS.

FIGURE 27-4: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



27.7 Operation During Sleep/Idle Modes

27.7.1 SLEEP MODE AND DEEP SLEEP MODES

When the device enters any Sleep mode, the CTMU module current source is always disabled. If the CTMU is performing an operation that depends on the current source when Sleep mode is invoked, the operation may not terminate correctly. Capacitance and time measurements may return erroneous values.

27.7.2 IDLE MODE

The behavior of the CTMU in Idle mode is determined by the CTMUSIDL bit (CTMUCONH<5>). If CTMUSIDL is cleared, the module will continue to operate in Idle mode. If CTMUSIDL is set, the module's current source is disabled when the device enters Idle mode. In this case, if the module is performing an operation when Idle mode is invoked, the results will be similar to those with Sleep mode.

27.8 Effects of a Reset on CTMU

Upon Reset, all registers of the CTMU are cleared. This leaves the CTMU module disabled; its current source is turned off and all configuration options return to their default settings. The module needs to be re-initialized following any Reset.

If the CTMU is in the process of taking a measurement at the time of Reset, the measurement will be lost. A partial charge may exist on the circuit that was being measured, and should be properly discharged before the CTMU makes subsequent attempts to make a measurement. The circuit is discharged by setting and then clearing the IDISSEN bit (CTMUCONH<1>) while the A/D Converter is connected to the appropriate channel.

28.0 SPECIAL FEATURES OF THE CPU

PIC18F47J53 family devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)
- Two-Speed Start-up
- Code Protection
- In-Circuit Serial Programming (ICSP)

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 3.0 "Oscillator Configurations"**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet. In addition to their Power-up and Oscillator Start-up Timers provided for Resets, the PIC18F47J53 family of devices has a configurable Watchdog Timer (WDT), which is controlled in software.

The inclusion of an internal RC oscillator also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

28.1 Configuration Bits

The Configuration bits can be programmed to select various device configurations. The configuration data is stored in the last four words of Flash program memory; Figure 6-1 depicts this. The configuration data gets loaded into the volatile Configuration registers, CONFIG1L through CONFIG4H, which are readable and mapped to program memory, starting at location 300000h.

Table 28-2 provides a complete list. A detailed explanation of the various bit functions is provided in Register 28-1 through Register 28-6.

28.1.1 CONSIDERATIONS FOR CONFIGURING THE PIC18F47J53 FAMILY DEVICES

Unlike some previous PIC18 microcontrollers, devices of the PIC18F47J53 family do not use persistent memory registers to store configuration information. The Configuration registers, CONFIG1L through CONFIG4H, are implemented as volatile memory.

Immediately after power-up, or after a device Reset, the microcontroller hardware automatically loads the CONFIG1L through CONFIG4L registers with configuration data stored in nonvolatile Flash program memory. The last four words of Flash program memory, known as the Flash Configuration Words (FCW), are used to store the configuration data.

Table 28-1 provides the Flash program memory, which will be loaded into the corresponding Configuration register.

When creating applications for these devices, users should always specifically allocate the location of the FCW for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The four Most Significant bits (MSb) of the FCW, corresponding to CONFIG1H, CONFIG2H, CONFIG3H and CONFIG4H, should always be programmed to '1111'. This makes these FCWs appear to be NOP instructions in the remote event that their locations are ever executed by accident.

The four MSbs of the CONFIG1H, CONFIG2H, CONFIG3H and CONFIG4H registers are not implemented, so writing '1's to their corresponding FCW has no effect on device operation.

To prevent inadvertent configuration changes during code execution, the Configuration registers, CONFIG1L through CONFIG4L, are loaded only once per power-up or Reset cycle. User's firmware can still change the configuration by using self-reprogramming to modify the contents of the FCW.

Modifying the FCW will not change the active contents being used in the CONFIG1L through CONFIG4H registers until after the device is reset.

REGISTER 28-8: CONFIG4H: CONFIGURATION REGISTER 4 HIGH (BYTE ADDRESS 300007h)

U-1	U-1	U-1	U-1	R/WO-1	U-0	R/WO-1	R/WO-1
—	—	—	—	LS48MHZ	—	WPEND	WPDIS
bit 7							bit 0

Legend:			
R = Readable bit	WO = Write-Once bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-4 **Unimplemented:** Program the corresponding Flash Configuration bit to ‘1’

bit 3 **LS48MHZ:** Low-Speed USB Clock Selection

1 = 48-MHz system clock is expected; divide-by-8 generates low-speed USB clock
0 = 24-MHz system clock is expected; divide-by-4 generates low-speed USB clock

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **WPEND:** Write-Protect Disable bit

1 = Flash pages, WPFP<6:0> to (Configuration Words page), are write/erase protected
0 = Flash pages 0 to WPFP<6:0> are write/erase-protected

bit 0 **WPDIS:** Write-Protect Disable bit

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

REGISTER 28-9: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F47J53 FAMILY DEVICES (BYTE ADDRESS 3FFFFEh)

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

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

bit 7-5 **DEV<2:0>:** Device ID bits

These bits are used with DEV<10:3> bits in Device ID Register 2 to identify the part number. See Register 28-10.

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

DECFSZ — Decrement f, Skip if 0

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

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

Operation: (f) – 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

0010	11da	ffff	ffff

Description: The contents of register ‘f’ are decremented. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed back in register ‘f’ (default).

If the result is ‘0’, the next instruction which is already fetched is discarded and a NOP is executed instead, making it a 2-cycle instruction.

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

If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1(2)

Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

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

If skip:

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

If skip and followed by 2-word instruction:

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

Example:

```
HERE      DECFSZ   CNT, 1, 1
          GOTO     LOOP
CONTINUE
```

Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT – 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT ≠ 0;
PC = Address (HERE + 2)

DCFSNZ — Decrement f, Skip if Not 0

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

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

0100	11da	ffff	ffff

Description: The contents of register ‘f’ are decremented. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed back in register ‘f’ (default).

If the result is not ‘0’, the next instruction which is already fetched is discarded and a NOP is executed instead, making it a 2-cycle instruction.

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

If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1(2)

Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

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

If skip:

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

If skip and followed by 2-word instruction:

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

Example:

```
HERE    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```

Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP – 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

29.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note:	Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing (**Section 6.6.1 "Indexed Addressing with Literal Offset"**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank (a = 0) or in a GPR bank designated by the BSR (a = 1). When the extended instruction set is enabled and a = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward-compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 29.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands"**).

Although the Indexed Literal Offset mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

29.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument 'f' in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value 'k'. As already noted, this occurs only when 'f' is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[ ]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within the brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled), when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM Assembler.

The destination argument 'd' functions as before.

In the latest versions of the MPASM Assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, /y, or the PE directive in the source listing.

29.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18F47J53 family, it is very important to consider the type of code. A large, re-entrant application that is written in C and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

31.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings[†]

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any digital only I/O pin or $\overline{\text{MCLR}}$ with respect to V_{SS} (except V_{DD})	-0.3V to 6.0V
Voltage on any combined digital and analog pin with respect to V_{SS} (except V_{DD})	-0.3V to (V_{DD} + 0.3V)
Voltage on V_{DDCORE} with respect to V_{SS}	-0.3V to 2.75V
Voltage on V_{DD} with respect to V_{SS}	-0.3V to 4.0V
Voltage on V_{USB} with respect to V_{SS}	(V_{DD} – 0.3V) to 4.0V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of V_{SS} pin	300 mA
Maximum current into V_{DD} pin	250 mA
Maximum output current sunk by any PORTB, PORTC and RA6 I/O pin	25 mA
Maximum output current sunk by any PORTA (except RA6), PORTD and PORTE I/O pin	8 mA
Maximum output current sourced by any PORTB, PORTC and RA6 I/O pin	25 mA
Maximum output current sourced by any PORTA (except RA6), PORTD and PORTE I/O pin	8 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

Note 1: Power dissipation is calculated as follows:

$$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

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

PIC18LF47J53 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J53 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param. No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (IDD)[2]						
	PIC18LFXXJ53	5.5	14.2	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 31 kHz (**RC_RUN** mode, Internal RC Oscillator, INTSRC = `0`)
		5.8	14.2	μA	+25°C		
		7.9	19.0	μA	+85°C		
	PIC18LFXXJ53	8.4	16.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		8.5	16.5	μA	+25°C		
		11.3	25.0	μA	+85°C		
	PIC18FXXJ53	23.7	60.0	μA	-40°C	VDD = 2.15V VDDCORE = 10 μF Capacitor	
		27.8	60.0	μA	+25°C		
		34.0	70.0	μA	+85°C		
	PIC18FXXJ53	26.1	70.0	μA	-40°C	VDD = 3.3V VDDCORE = 10 μF Capacitor	
		29.6	70.0	μA	+25°C		
		36.2	96.0	μA	+85°C		
	PIC18LFXXJ53	0.87	1.5	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0	FOSC = 4 MHz, **RC_RUN** mode, Internal RC Oscillator
		0.91	1.5	mA	+25°C		
		0.95	1.6	mA	+85°C		
	PIC18LFXXJ53	1.23	2.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		1.24	2.0	mA	+25°C		
		1.25	2.0	mA	+85°C		
	PIC18FXXJ53	0.99	2.4	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF capacitor	
		1.02	2.4	mA	+25°C		
		1.06	2.6	mA	+85°C		
	PIC18FXXJ53	1.31	2.6	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF capacitor	
		1.25	2.6	mA	+25°C		
		1.26	2.7	mA	+85°C		

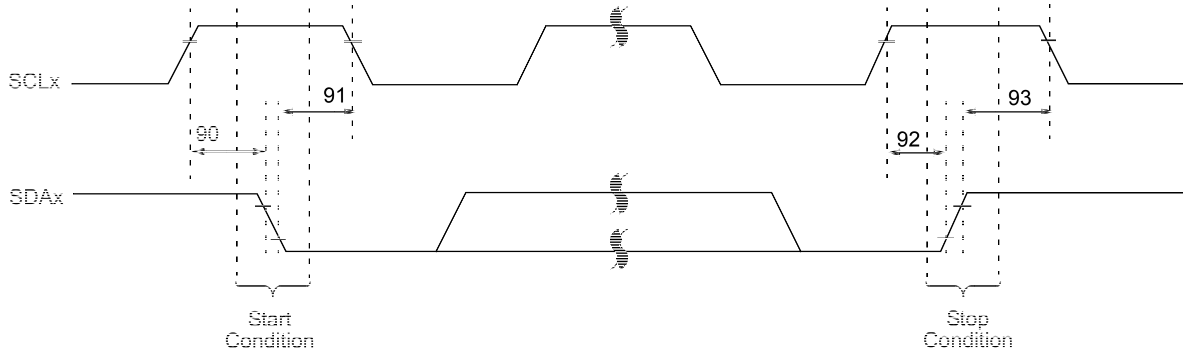
Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

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:
$\overline{\text{OSC1}}$ = 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 31-16: I²C BUS START/STOP BITS TIMING

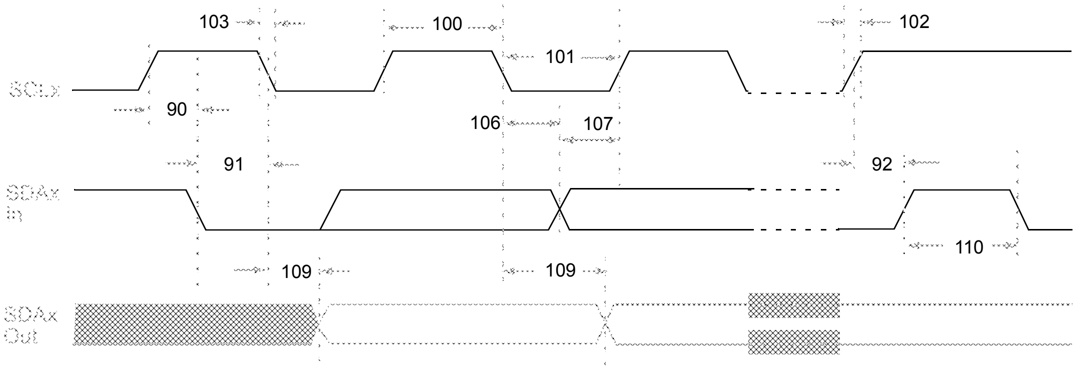


Note: Refer to Figure 31-4 for load conditions.

TABLE 31-24: I²C BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
90	TSU:STA	Start Condition	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
		Setup Time	400 kHz mode	600	—		
91	THD:STA	Start Condition	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	600	—		
92	TSU:STO	Stop Condition	100 kHz mode	4700	—	ns	
		Setup Time	400 kHz mode	600	—		
93	THD:STO	Stop Condition	100 kHz mode	4000	—	ns	
		Hold Time	400 kHz mode	600	—		

FIGURE 31-17: I²C BUS DATA TIMING



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