

#### 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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf27j13-i-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

## 7.2.2 TABLE LATCH REGISTER (TABLAT)

The Table Latch (TABLAT) is an 8-bit register mapped into the Special Function Register (SFR) space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

### 7.2.3 TABLE POINTER REGISTER (TBLPTR)

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR comprises three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22<sup>nd</sup> bit allows access to the Device ID, the User ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation.

 Table 7-1 displays these operations. On the TBLPTRT,

 these operations only affect the low-order 21 bits.

### 7.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into TABLAT.

When a TBLWT is executed, the seven Least Significant bits (LSbs) of the Table Pointer register (TBLPTR<6:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 12 Most Significant bits (MSbs) of the TBLPTR (TBLPTR<21:10>) determine which program memory block of 1024 bytes is written to. For more information, see Section 7.5 "Writing to Flash Program Memory".

When an erase of program memory is executed, the 12 MSbs of the Table Pointer register point to the 1024-byte block that will be erased. The LSbs are ignored.

Figure 7-3 illustrates the relevant boundaries of the TBLPTR based on Flash program memory operations.

	TABLE POINTER OPERATIONS WITH THE PO	
IADLE /-I.	TABLE FOINTER OFERATIONS WITH TBLRD	AND TELMT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 7-3:

#### TABLE POINTER BOUNDARIES BASED ON OPERATION



## 9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2, PIE3). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

#### REGISTER 9-9: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ACCESS F9Dh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIE <sup>(1)</sup>	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 7	<b>PMPIE:</b> Parallel Master Port Read/Write Interrupt Enable bit <sup>(1)</sup> 1 = Enables the PMP read/write interrupt
	0 = Disables the PMP read/write interrupt
bit 6	ADIE: A/D Converter Interrupt Enable bit
	1 = Enables the A/D interrupt
	0 = Disables the A/D interrupt
bit 5	RC1IE: EUSART1 Receive Interrupt Enable bit
	1 = Enables the EUSART1 receive interrupt
	0 = Disables the EUSART1 receive interrupt
bit 4	TX1IE: EUSART1 Transmit Interrupt Enable bit
	1 = Enables the EUSART1 transmit interrupt
	0 = Disables the EUSART1 transmit interrupt
bit 3	SSP1IE: Master Synchronous Serial Port 1 Interrupt Enable bit
	1 = Enables the MSSP1 interrupt
	0 = Disables the MSSP1 interrupt
bit 2	CCP1IE: ECCP1 Interrupt Enable bit
	1 = Enables the ECCP1 interrupt
	0 = Disables the ECCP1 interrupt
bit 1	TMR2IE: TMR2 to PR2 Match Interrupt Enable bit
	1 = Enables the TMR2 to PR2 match interrupt
	0 = Disables the TMR2 to PR2 match interrupt
bit 0	TMR1IE: TMR1 Overflow Interrupt Enable bit
	1 = Enables the TMR1 overflow interrupt
	0 = Disables the TMR1 overflow interrupt
Note 1:	These bits are unimplemented on 28-pin devices.

## REGISTER 11-5: PMEH: PARALLEL PORT ENABLE REGISTER HIGH BYTE (BANKED F57h)<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 7							bit 0

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

bit 7-6	PTEN<15:14>: PMCS Port Enable bits
	1 = PMA<15:14> function as either PMA<15:14> or PMCS 0 = PMA<15:14> function as port I/O
bit 5-0	PTEN<13:8>: PMP Address Port Enable bits
	1 = PMA<13:8> function as PMP address lines
	0 = PMA<13:8> function as port I/O

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

## REGISTER 11-6: PMEL: PARALLEL PORT ENABLE REGISTER LOW BYTE (BANKED F56h)<sup>(1)</sup>

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PTEN7 | PTEN6 | PTEN5 | PTEN4 | PTEN3 | PTEN2 | PTEN1 | PTEN0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

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

bit 7-2	PTEN<7:2>: PMP Address Port Enable bits
	1 = PMA<7:2> function as PMP address lines
	0 = PMA<7:2> function as port I/O
bit 1-0	PTEN<1:0>: PMALH/PMALL Strobe Enable bits
	1 = PMA<1:0> function as either PMA<1:0> or PMALH and PMALL
	0 = PMA<1:0> pads functions as port I/O

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

## 11.3 MASTER PORT MODES

In its Master modes, the PMP module provides an 8-bit data bus, up to 16 bits of address, and all the necessary control signals to operate a variety of external parallel devices, such as memory devices, peripherals and slave microcontrollers. To use the PMP as a master, the module must be enabled (PMPEN = 1) and the mode must be set to one of the two possible Master modes (PMMODEH<1:0> = 10 or 11).

Because there are a number of parallel devices with a variety of control methods, the PMP module is designed to be extremely flexible to accommodate a range of configurations. Some of these features include:

- · 8-Bit and 16-Bit Data modes on an 8-bit data bus
- · Configurable address/data multiplexing
- · Up to two chip select lines
- Up to 16 selectable address lines
- · Address auto-increment and auto-decrement
- · Selectable polarity on all control lines
- Configurable Wait states at different stages of the read/write cycle

## 11.3.1 PMP AND I/O PIN CONTROL

Multiple control bits are used to configure the presence or absence of control and address signals in the module. These bits are PTBEEN, PTWREN, PTRDEN and PTEN<15:0>. They give the user the ability to conserve pins for other functions and allow flexibility to control the external address. When any one of these bits is set, the associated function is present on its associated pin; when clear, the associated pin reverts to its defined I/O port function.

Setting a PTENx bit will enable the associated pin as an address pin and drive the corresponding data contained in the PMADDR register. Clearing a PTENx bit will force the pin to revert to its original I/O function.

For the pins configured as chip select (PMCS) with the corresponding PTENx bit set, the PTEN0 and PTEN1 bits will also control the PMALL and PMALH signals. When multiplexing is used, the associated address latch signals should be enabled.

## 11.3.2 READ/WRITE-CONTROL

The PMP module supports two distinct read/write signaling methods. In Master Mode 1, read and write strobes are combined into a single control line, PMRD/PMWR. A second control line, PMENB, determines when a read or write action is to be taken. In Master Mode 2, separate read and write strobes (PMRD and PMWR) are supplied on separate pins.

All control signals (PMRD, PMWR, PMBE, PMENB, PMAL and PMCS) can be individually configured as either positive or negative polarity. Configuration is controlled by separate bits in the PMCONL register. Note that the polarity of control signals that share the same output pin (for example, PMWR and PMENB) are controlled by the same bit; the configuration depends on which Master Port mode is being used.

## 11.3.3 DATA WIDTH

The PMP supports data widths of both 8 bits and 16 bits. The data width is selected by the MODE16 bit (PMMODEH<2>). Because the data path into and out of the module is only 8 bits wide, 16-bit operations are always handled in a multiplexed fashion, with the Least Significant Byte (LSB) of data being presented first. To differentiate data bytes, the byte enable control strobe, PMBE, is used to signal when the Most Significant Byte (MSB) of data is being presented on the data lines.

## 11.3.4 ADDRESS MULTIPLEXING

In either of the Master modes (PMMODEH<1:0> = 1x), the user can configure the address bus to be multiplexed together with the data bus. This is accomplished by using the ADRMUX<1:0> bits (PMCONH<4:3>). There are three address multiplexing modes available. Typical pinout configurations for these modes are displayed in Figure 11-9, Figure 11-10 and Figure 11-11.

In Demultiplexed mode (PMCONH<4:3> = 00), data and address information are completely separated. Data bits are presented on PMD<7:0>, and address bits are presented on PMADDRH<6:0> and PMADDRL<7:0>.

In Partially Multiplexed mode (PMCONH<4:3> = 01), the lower eight bits of the address are multiplexed with the data pins on PMD<7:0>. The upper eight bits of the address are unaffected and are presented on PMADDRH<6:0>. The PMA0 pin is used as an address latch and presents the Address Latch Low (PMALL) enable strobe. The read and write sequences are extended by a complete CPU cycle during which the address is presented on the PMD<7:0> pins.

In Fully Multiplexed mode (PMCONH<4:3> = 10), the entire 16 bits of the address are multiplexed with the data pins on PMD<7:0>. The PMA0 and PMA1 pins are used to present Address Latch Low (PMALL) enable and Address Latch High (PMALH) enable strobes, respectively. The read and write sequences are extended by two complete CPU cycles. During the first cycle, the lower eight bits of the address are presented on the PMD<7:0> pins with the PMALL strobe active. During the second cycle, the upper eight bits of the address are presented on the PMD<7:0> pins with the PMALH strobe active. In the event the upper address bits are configured as chip select pins, the corresponding address bits are automatically forced to '0'.

## 14.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP modules

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

## 14.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (Fosc/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS<1:0> (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see Section 14.2 "Timer2 Interrupt").

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- · a write to the T2CON register
- any device Reset (Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR))

TMR2 is not cleared when T2CON is written.

## REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER (ACCESS FCAh)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	l 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	<b>T2OUTPS&lt;3:0&gt;:</b> Timer2 Output Postscale Select bits 0000 = 1:1 Postscale 0001 = 1:2 Postscale
	•
	1111 = 1:16 Postscale
bit 2	<b>TMR2ON:</b> Timer2 On bit 1 = Timer2 is on 0 = Timer2 is off
bit 1-0	<b>T2CKPS&lt;1:0&gt;:</b> Timer2 Clock Prescale Select bits 00 = Prescaler is 1 01 = Prescaler is 4 10 = Prescaler is 16

NOTES:

# PIC18F47J13 FAMILY

## REGISTER 17-9: DAY: DAY VALUE REGISTER<sup>(1)</sup>

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 7-6	Unimplemented: Read as '0'
bit 5-4	<b>DAYTEN&lt;1:0&gt;:</b> Binary Coded Decimal value of Day's Tens Digit bits Contains a value from 0 to 3.
bit 3-0	<b>DAYONE&lt;3:0&gt;:</b> Binary Coded Decimal Value of Day's Ones Digit bits Contains a value from 0 to 9.

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

## **REGISTER 17-10: WKDY: WEEKDAY VALUE REGISTER<sup>(1)</sup>**

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

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

bit 7-3 Unimplemented: Read as '0'

bit 2-0 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits Contains a value from 0 to 6.

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

# PIC18F47J13 FAMILY

## **REGISTER 17-11: HOURS: HOURS VALUE REGISTER<sup>(1)</sup>**

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

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

bit 7-6	Unimplemented: Read as '0'
bit 5-4	<b>HRTEN&lt;1:0&gt;:</b> Binary Coded Decimal Value of Hour's Tens Digit bits Contains a value from 0 to 2.
bit 3-0	<b>HRONE&lt;3:0&gt;:</b> Binary Coded Decimal Value of Hour's Ones Digit bits Contains a value from 0 to 9.

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

### **REGISTER 17-12: MINUTES: MINUTES VALUE REGISTER**

U-0	R/W-x						
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
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 6-4	<b>MINTEN&lt;2:0&gt;:</b> Binary Coded Decimal Value of Minute's Tens Digit bits Contains a value from 0 to 5.
bit 3-0	<b>MINONE&lt;3:0&gt;:</b> Binary Coded Decimal Value of Minute's Ones Digit bits Contains a value from 0 to 9.

## REGISTER 17-13: SECONDS: SECONDS VALUE REGISTER

U-0	R/W-x						
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

Legend:				
R = Reada	ble 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	Unimple	mented: Read as '0'		
bit 6-4	SECTEN	I<2:0>: Binary Coded Decin	nal Value of Second's Tens Dig	git bits
	Contains	a value from 0 to 5.		
L:1 0 0	05001			C. M. L. M.

bit 3-0 SECONE<3:0>: Binary Coded Decimal Value of Second's Ones Digit bits Contains a value from 0 to 9.

## REGISTER 19-6: PSTRxCON: PULSE STEERING CONTROL (1, ACCESS FBFh; 2, FB9h; 3, BANKED F1Ah)<sup>(1)</sup>

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
CMPL1	CMPL0	_	STRSYNC	STRD	STRC	STRB	STRA
bit 7	•				•	•	bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplei	mented bit, rea	d as '0'	
-n = Value at I	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown
bit 7-6	CMPL<1:0>: 1 = Modulate 0 = Complem ing mode	Complementa ed output pin to nentary output	ry Mode Outpu ggles between assignment is c	t Assignment PxA and PxB disabled; the S	Steering Sync b for each period TRD:STRA bits	bits d are used to de	termine Steer-
bit 5	Unimplemen	ted: Read as '	0'				
bit 4	STRSYNC: S	teering Sync b	it				
	1 = Output st 0 = Output st	eering update	occurs on the occurs at the b	next PWM per beginning of th	iod e instruction cy	cle boundary	
bit 3	<b>STRD:</b> Steerin 1 = PxD pin h 0 = PxD pin i	ng Enable D b nas the PWM v s assigned to a	it waveform with   a port pin	polarity contro	from CCPxM<	:1:0>	
bit 2	STRC: Steerin	ng Enable C b	it				
	1 = PxC pin h 0 = PxC pin i	has the PWM s assigned to	waveform with a port pin	polarity contro	from CCPxM<	:1:0>	
bit 1	STRB: Steerin	ng Enable B bi	it				
	1 = PxB pin h 0 = PxB pin is	has the PWM v s assigned to a	vaveform with   a port pin	polarity control	from CCPxM<	:1:0>	
bit 0	STRA: Steerin	ng Enable A bi	it ,				
	1 = PxA pin h 0 = PxA pin is	s assigned to a	vaveform with j a port pin	polarity contro	trom CCPxM<	1:0>	
Note 1: The Px	e PWM Steering M<1:0> = 00.	g mode is avai	able only wher	n the CCPxCO	N register bits,	CCPxM<3:2>	= 11 and

## 19.4.8 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2 will not increment and the state of the module will not change. If the ECCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from HFINTOSC and the postscaler may not be immediately stable.

In PRI\_IDLE mode, the primary clock will continue to clock the ECCPx module without change.

## 19.4.8.1 Operation with Fail-Safe Clock Monitor (FSCM)

If the Fail-Safe Clock Monitor (FSCM) is enabled, a clock failure will force the device into the power-managed RC RUN mode and the OSCFIF bit of the PIR2 register

will be set. The ECCPx will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

## 19.4.9 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states.

This forces the ECCP module to reset to a state compatible with previous, non-enhanced CCP modules used on other PIC18 and PIC16 devices.

TABLE 19-4:	REGISTERS ASSOCIATED WITH ECCP1/2/3 MODULE AND
	TIMER1/2/3/4/6/8

File 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
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR4	CCP10IF	CCP9IF	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
PIE2	OSCFIE	CM2IE	CM1IE		BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE4	CCP10IE	CCP9IE	CCP8IE	CCP7IE	CCP6IE	CCP5IE	CCP4IE	CCP3IE
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
IPR2	OSCFIP	CM2IP	CM1IP		BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR4	CCP10IP	CCP9IP	CCP8IP	CCP7IP	CCP6IP	CCP5IP	CCP4IP	CCP3IP
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
TRISC	TRISC7	TRISC6	_	—	—	TRISC2	TRISC1	TRISC0
TRISE	RDPU	REPU	—	—	—	TRISE2	TRISE1	TRISE0
TMR1H	Timer1 Register	High Byte						
TMR1L	Timer1 Register	Low Byte						
TMR2	Timer2 Register							
TMR3H	Timer3 Register	High Byte						
TMR3L	Timer3 Register	Low Byte						
TMR4	Timer4 Register	-						
TMR6	Timer6 Register	-						
TMR8	Timer8 Register	•						
PR2	Timer2 Period R	Register						
PR4	Timer4 Period R	Register						
PR6	Timer6 Period F	Register						
PR8	Timer8 Period F	Register						
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	RD16	TMR10N
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	T3SYNC	RD16	TMR3ON
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
T6CON	_	T6OUTPS3	T6OUTPS2	T6OUTPS1	T6OUTPS0	TMR6ON	T6CKPS1	T6CKPS0

© 2010-2017 Microchip Technology Inc.

## 20.3.4 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPxCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCON1 registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and SSx pins as serial port pins. For the pins to behave as the serial port function, the appropriate TRISx bits, PCFGx bits and Peripheral Pin Select registers (if using MSSP2) should be correctly initialized prior to setting the SSPEN bit.

A typical SPI serial port initialization process follows:

- Initialize the ODCON3 register (optional open-drain output control)
- Initialize the remappable pin functions (if using MSSP2, see Section 10.7 "Peripheral Pin Select (PPS)")
- Initialize the SCKx/LAT value to the desired Idle SCKx level (if master device)
- Initialize the SCKx/PCFGx bit (if in Slave mode and multiplexed with the ANx function)
- Initialize the SCKx/TRISx bit as an output (Master mode) or input (Slave mode)
- Initialize the SDIx/PCFGx bit (if SDIx is multiplexed with the ANx function)
- Initialize the SDIx/TRISx bit
- Initialize the SSx/PCFG bit (if in Slave mode and multiplexed with the ANx function)
- Initialize the SSx/TRISx bit (Slave modes)
- · Initialize the SDOx/TRISx bit
- Initialize the SSPxSTAT register
- Initialize the SSPxCON1 register
- Set the SSPEN bit to enable the module

Any MSSP1 serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value. If individual MSSP2 serial port functions will not be used, they may be left unmapped.

Note: When MSSP2 is used in SPI Master mode, the SCK2 function must be configured as both an output and an input in the PPS module. SCK2 must be initialized as an output pin (by writing 0x0A to one of the RPORx registers). Additionally, SCK2IN must also be mapped to the same pin by initializing the RPINR22 register. Failure to initialize SCK2/SCK2IN as both output and input will prevent the module from receiving data on the SDI2 pin, as the module uses the SCK2IN signal to latch the received data.

## 20.3.5 TYPICAL CONNECTION

Figure 20-2 illustrates a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends valid data Slave sends dummy data
- Master sends valid data Slave sends valid data
- · Master sends dummy data Slave sends valid data



## FIGURE 20-2: SPI MASTER/SLAVE CONNECTION

## 20.5.6.1 I<sup>2</sup>C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted, 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. S and P conditions are output to indicate the beginning and end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the  $R/\overline{W}$  bit. In this case, the  $R/\overline{W}$  bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address, followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. S and P conditions indicate the beginning and end of transmission.

The BRG used for the SPI mode operation is used to set the SCLx clock frequency for either 100 kHz, 400 kHz or 1 MHz  $I^2$ C operation. See Section 20.5.7 "Baud Rate" for more details.

A typical transmit sequence would go as follows:

- 1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPxCON2<0>).
- SSPxIF is set. The MSSP module will wait for the required start time before any other operation takes place.
- 3. The user loads the SSPxBUF with the slave address to transmit.
- 4. The address is shifted out of the SDAx pin until all 8 bits are transmitted.
- The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPx-CON2 register (SSPxCON2<6>).
- The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
- 7. The user loads the SSPxBUF with 8 bits of data.
- 8. Data is shifted out of the SDAx pin until all 8 bits are transmitted.
- The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPx-CON2 register (SSPxCON2<6>).
- 10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
- 11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPxCON2<2>).
- 12. The interrupt is generated once the Stop condition is complete.

## 20.5.7 BAUD RATE

In I<sup>2</sup>C Master mode, the BRG reload value is placed in the lower seven bits of the SSPxADD register (Figure 20-19). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TcY) on the Q2 and Q4 clocks. In I<sup>2</sup>C Master mode, the BRG is reloaded automatically.

Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCLx pin will remain in its last state.

Table 20-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD. The SSPADD BRG value of 0x00 is not supported.







## 21.2 EUSART Asynchronous Mode

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

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

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

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

#### 21.2.1 EUSART ASYNCHRONOUS TRANSMITTER

Figure 21-3 displays the EUSART transmitter block diagram.

The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREGx register (if available). Once the TXREGx register transfers the data to the TSR register (occurs in one TcY), the TXREGx register is empty and the TXxIF flag bit is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxIE. TXxIF will be set regardless of the state of TXxIE; it cannot be cleared in software. TXxIF is also not cleared immediately upon loading TXREGx, but becomes valid in the second instruction cycle following the load instruction. Polling TXxIF immediately following a load of TXREGx will return invalid results.

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

Note 1:	The TSR register is not mapped in data
	memory, so it is not available to the user.
2:	Flag bit, TXxIF, is set when enable bit, TXEN, is set.

To set up an Asynchronous Transmission:

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

## FIGURE 21-3: EUSART TRANSMIT BLOCK DIAGRAM



### TABLE 28-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM Access bit:
	a = 0: RAM location in Access RAM (BSR register is ignored)
	a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: Carry, Digit Carry, Zero, Overflow, Negative.
d	Destination Select bit:
	d = 0: store result in title register f
dest	Destination: either the WREG register or the specified register file location
f	8-bit Register file address (00h to FEh) or 2-bit FSR designator (0h to 3h)
f	12-bit Register file address (000h to FFFh). This is the source address
f,	12-bit Register file address (000h to FEFh). This is the destination address
GIE	Global Interrupt Enable bit
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	
mm	The mode of the TBLPTR register for the table read and table write instructions.
	Only used with table read and table write instructions:
*	No change to register (such as TBLPTR with table reads and writes)
*+	Post-Increment register (such as TBLPTR with table reads and writes)
*_	Post-Decrement register (such as TBLPTR with table reads and writes)
+*	Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for
	Call/Branch and Return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
PD	Power-Down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
S	Fast Call/Return Mode Select bit:
	s = 0: do not update into/from shadow registers
משת זמש	$S = \pm$ . Certain registers loaded into non shadow registers (r as mode)
	8-bit Table Latch
TO	Ton-of-Stack
100	
WDT	Watchdog Timer
WREG	Working register (accumulator)
x	Don't care ('0' or '1') The assembler will generate code with $x = 0$ . It is the recommended form of use for
	compatibility with all Microchip software tools.
Zs	7-bit offset value for Indirect Addressing of register files (source).
Zd	7-bit offset value for Indirect Addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an Indexed Address.
(text)	The contents of text.
[expr] <n></n>	Specifies bit n of the register indicated by the pointer expr.
$\rightarrow$	Assigned to.
< >	Register bit field.
E	In the set of.
italics	User-defined term (font is Courier New).

## PIC18F47J13 FAMILY

РОР		Pop Top of Return Stack				
Synt	ax:	POP				
Oper	rands:	None				
Oper	ration:	$(TOS) \rightarrow b$	it bucke	t		
Statu	is Affected:	None				
Enco	oding:	0000	0000	000	0 0	0110
Desc	cription:	The TOS v stack and is then becon was pushe This instruct the user to stack to inc	alue is s discar nes the d onto t ction is properl corporat	pulled o ded. Th previou he retu provide y mana e a soft	off the ne T( us va rn sta d to uge th tware	e return OS value Iue that ack. enable ne return e stack.
Word	ds:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	G	23		Q4
	Decode	No operation	POP val	TOS ue	ор	No eration
Exar	<u>nple:</u>	POP GOTO	NEW			
	Before Instruc TOS Stack (1	tion level down)	= =	0031A 01433	2h 2h	
	After Instructio TOS PC	n	= =	014332 NEW	2h	

PUS	н	Push Top o	of Retu	rn Stac	:k			
Synta	ax:	PUSH	PUSH					
Oper	ands:	None						
Oper	ation:	$(PC + 2) \rightarrow$	TOS					
Statu	s Affected:	None						
Encoding:		0000	0000	000	00	0101		
Desc	ription:	The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.				e top of TOS stack. enting a OS and stack.		
Word	ls:	1						
Cycle	es:	1						
QC	ycle Activity:							
	Q1	Q2	Q	3		Q4		
	Decode	PUSH PC + 2 onto return stack	No opera	o ation	ор	No eration		
Exan	nple:	PUSH						
	Before Instruc TOS PC	tion	= =	345Ah 0124h				
	After Instructio PC TOS Stack (1	on level down)	= = =	0126h 0126h 345Ah				

Add Literal to FSR2 and Return

ADDULNK

#### 28.2.2 EXTENDED INSTRUCTION SET

	FSR	Add Literal to FSR						
Synta	ax:	ADDFSR	ADDFSR f, k					
Oper	ands:	$0 \le k \le 63$	0.1					
Operation: $FSR(f) + k \rightarrow FSR(f)$								
Statu	s Affected:	None	None					
Enco	ding:	1110	1000	ffk	k kkkk			
Desc	ription:	The 6-bit I contents of	The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.					
Word	s:	1						
Cycle	es:	1	1					
QC	cle Activity:							
_	Q1	Q2	Q3		Q4			
	Decode	Read	Proces	SS	Write to			
		literal 'k'	Data	I	FSR			

ADDFSR 2, 0x23

03FFh

0422h

Example:

Before Instruction

FSR2

After Instruction FSR2

=

=

Syntax:	ADDULN	ADDULNK k			
Operands:	$0 \le k \le 63$	$0 \le k \le 63$			
Operation:	$FSR2 + k \rightarrow FSR2$ ,				
	$(TOS) \rightarrow F$	$(TOS) \rightarrow PC$			
Status Affected:	None	None			
Encoding:	1110	1000 11}	k kkkk		
Description:	The 6-bit I contents o executed I TOS.	The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS.			
	The instruction takes two cycles to execute; a NOP is performed during the second cycle.				
	This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.				
Words:	1	1			
Cycles:	2	2			
Q Cycle Activity:					
Q1	Q2	Q3	Q4		
Decode	Read literal 'k'	Process Data	Write to FSR		
No	No	No	No		
Operation	Operation	Operation	Operation		
Example:	ADDULNK (	0x23			
Before Instruc	tion				

impic.	Л	
Before Instru	ction	
FSR2	=	03FFh
PC	=	0100h
After Instructi	on	
FSR2	=	0422h
PC	=	(TOS)

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

## 30.5 AC (Timing) Characteristics

### 30.5.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS		3. Tcc:st	(I <sup>2</sup> C specifications only)
2. TppS		4. Ts	(I <sup>2</sup> C specifications only)
Т			
F	Frequency	Т	Time
Lowercase le	tters (pp) and their meanings:		
рр			
сс	CCP1	osc	OSC1
ck	CLKO	rd	RD
CS	CS	rw	RD or WR
di	SDI	SC	SCK
do	SDO	SS	SS
dt	Data in	tO	TOCKI
io	I/O port	t1	T13CKI
mc	MCLR	wr	WR
Uppercase le	tters and their meanings:		
S			
F	Fall	Р	Period
н	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance
I <sup>2</sup> C only			
AA	output access	High	High
BUF	Bus free	Low	Low
TCC:ST (I <sup>2</sup> C s	pecifications only)		
CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	Stop condition
STA	Start condition		

### 30.5.2 TIMING CONDITIONS

The temperature and voltages specified in Table 30-8 apply to all timing specifications unless otherwise noted. Figure 30-4 specifies the load conditions for the timing specifications.

#### TABLE 30-8: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

	Standard Operating Conditions (unless otherwise stated)			
AC CHARACTERISTICS	Operating temperature	$-40^\circ C \le TA \le +85^\circ C$	for industrial	
	Operating voltage VDD range as described in <b>Section 30.1</b> and <b>Section 30.3</b> .			

#### FIGURE 30-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS







## TABLE 30-27: MSSPx I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Мах	Units	Conditions
90	TSU:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)		ns	Only relevant for
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_		Repeated Start condition
91	THD:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	—		
92	Tsu:sto	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_		
93	THD:STO	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ns	
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	_		

## FIGURE 30-20: MSSPx I<sup>2</sup>C BUS DATA TIMING

