



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

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f722a-e-mv

Email: info@E-XFL.COM

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

1.0 DEVICE OVERVIEW

The PIC16(L)F722A/723A devices are covered by this data sheet. They are available in 28-pin packages. Figure 1-1 shows a block diagram of the PIC16(L)F722A/723A devices. Table 1-1 shows the pinout descriptions.

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F722A/723A has a 13-bit program counter capable of addressing a 2K x 14 program memory space for the PIC16(L)F722A (0000h-07FFh) and a 4K x 14 program memory space for the PIC16(L)F723A (0000h-0FFFh). Accessing a location above the memory boundaries for the PIC16(L)F722A will cause a wrap-around within the first 2K x 14 program memory space. Accessing a location above the memory boundaries for the PIC16(L)F723A will cause a wrap-around within the first 4K x 14 program memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE



FIGURE 2-2:

PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F723A



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Page
Bank 0											
00h ⁽²⁾	INDF	Addressing	this location	uses conter	nts of FSR to	address data	memory (no	t a physical r	egister)	xxxx xxxx	22,30
01h	TMR0	Timer0 Mod	lule Register							XXXX XXXX	91,30
02h ⁽²⁾	PCL	Program Co	ounter (PC) L	east Signific	cant Byte					0000 0000	21,30
03h ⁽²⁾	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	С	0001 1xxx	18,30
04h ⁽²⁾	FSR	Indirect Dat	a Memory A	ddress Point	er					XXXX XXXX	22,30
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	XXXX XXXX	43,30
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	XXXX XXXX	52,30
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	XXXX XXXX	62,30
09h	PORTE	_	_	_	_	RE3	_	_	_	xxxx	69,30
0Ah ^(1, 2)	PCLATH	—	_	_	Write Buffer	for the upper	5 bits of the	Program Cou	unter	0 0000	21,30
0Bh ⁽²⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	36,30
0Ch	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	39,30
0Dh	PIR2	_	_	_	_	_	_	_	CCP2IF	0	40,30
0Eh	TMR1L	Holding Reg	gister for the	Least Signif	icant Byte of	the 16-bit TM	IR1 Register			XXXX XXXX	99,30
0Fh	TMR1H	Holding Reg	gister for the	Most Signifi	cant Byte of t	he 16-bit TM	R1 Register			XXXX XXXX	99,30
10h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	_	TMR10N	0000 00-0	103,30
11h	TMR2	Timer2 Mod	lule Register							0000 0000	106,30
12h	T2CON	_	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	107,30
13h	SSPBUF	Synchronou	is Serial Por	t Receive Bu	uffer/Transmit	Register				XXXX XXXX	147,30
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	164,30
15h	CCPR1L	Capture/Co	mpare/PWM	Register (L	.SB)					XXXX XXXX	116,30
16h	CCPR1H	Capture/Co	mpare/PWM	Register (N	ISB)					xxxx xxxx	116,30
17h	CCP1CON	_	_	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00 0000	115,30
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	134,30
19h	TXREG	USART Tra	nsmit Data F	Register						0000 0000	133,30
1Ah	RCREG	USART Red	ceive Data R	egister						0000 0000	131,30
1Bh	CCPR2L	Capture/Co	mpare/PWM	Register 2	(LSB)					XXXX XXXX	116,30
1Ch	CCPR2H	Capture/Co	mpare/PWM	Register 2	(MSB)					xxxx xxxx	116,30
1Dh	CCP2CON	_	_	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	00 0000	115,30
1Eh	ADRES	A/D Result	Register							xxxx xxxx	86,30
1Fh	ADCON0	_	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	00 0000	85,30

	DIC16/LIE722A/723A SPECIAL EUNCTION PEO	
IADLE Z-I:	PICIO(L)F/22A//23A SPECIAL FUNCTION REC	JOIER JUNINART

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: Accessible only when SSPM<3:0> = 1001.

4: Accessible only when SSPM<3:0> \neq 1001.

5: This bit is always '1' as RE3 is input-only.

2.2.2.3 PCON Register

The Power Control (PCON) register contains flag bits (refer to Table 3-2) to differentiate between a:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- Watchdog Timer Reset (WDT)
- External MCLR Reset

The PCON register also controls the software enable of the BOR.

The PCON register bits are shown in Register 2-3.

REGISTER 2-3: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-q	R/W-q
—	—	—	—	—	—	POR	BOR
bit 7							bit 0

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

hit 7-2	Unimplemented: Read as '0'
bit 1	POR: Power-on Reset Status bit
	 1 = No Power-on Reset occurred 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
bit 0	 BOR: Brown-out Reset Status bit 1 = No Brown-out Reset occurred 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

Note 1: Set BOREN<1:0> = 01 in the Configuration Word register for this bit to control the $\overline{\text{BOR}}$.

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> \rightarrow PCH). The lower example in Figure 2-5 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> \rightarrow PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note *AN556, Implementing a Table Read* (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (refer to Figures 2-1 and 2-2). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth PUSH overwrites the value that was stored from the first PUSH. The tenth PUSH overwrites the second PUSH (and so on).

- **Note 1:** There are no Status bits to indicate stack overflow or stack underflow conditions.
 - 2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Program Memory Paging

All devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is POPed off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note:	The contents of the PCLATH register are
	unchanged after a RETURN or RETFIE
	instruction is executed. The user must
	rewrite the contents of the PCLATH
	register for any subsequent subroutine
	calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

	ORG 500	h					
	PAGESEL	SUB_P1	;Select page 1				
			;(800h-FFFh)				
	CALL	SUB1_P1	;Call subroutine in				
	:		;page 1 (800h-FFFh)				
	:						
	ORG	900h	;page 1 (800h-FFFh)				
SUB1_P1							
	:		;called subroutine				
			;page 1 (800h-FFFh)				
	:						
	RETURN		;return to				
			;Call subroutine				
			;in page 0				
			;(000h-7FFh)				

4.0 INTERRUPTS

The PIC16(L)F722A/723A device family features an interruptible core, allowing certain events to preempt normal program flow. An Interrupt Service Routine (ISR) is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

The PIC16(L)F722A/723A device family has 12 interrupt sources, differentiated by corresponding interrupt enable and flag bits:

- Timer0 Overflow Interrupt
- External Edge Detect on INT Pin Interrupt
- PORTB Change Interrupt
- Timer1 Gate Interrupt
- A/D Conversion Complete Interrupt
- AUSART Receive Interrupt
- AUSART Transmit Interrupt
- SSP Event Interrupt
- CCP1 Event Interrupt
- Timer2 Match with PR2 Interrupt
- Timer1 Overflow Interrupt
- CCP2 Event Interrupt

A block diagram of the interrupt logic is shown in Figure 4-1.



6.2.1 ANSELA REGISTER

The ANSELA register (Register 6-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

REGISTER 6-4: ANSELA: PORTA ANALOG SELECT REGISTER

'1' = Bit is set

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1		
—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0		
bit 7 bi									
Legend:									
R = Readable bit $W = Writable bit$				U = Unimplemented bit, read as '0'					

'0' = Bit is cleared

bit 7-6 Unimplemented: Read as '0'

-n = Value at POR

bit 5-0 **ANSA<5:0>**: Analog Select between Analog or Digital Function on pins RA<5:0>, respectively 0 = Digital I/O. Pin is assigned to port or digital special function. 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

x = Bit is unknown

PIC16(L)F722A/723A



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/ DONE	ADON	85
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	ADREF1	ADREF0	86
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	44
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	53
ADRES			A/[D Result Re	egister Byte				86
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	115
FVRCON	FVRRDY	FVREN	_	—	—	—	ADFVR1	ADFVR0	90
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	36
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	37
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	43
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	52

TABLE 9-2: SUMMARY OF ASSOCIATED ADC REGISTERS

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', <math>q = value depends on condition. Shaded cells are not used for ADC module.

PIC16(L)F722A/723A

11.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- · Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 11-1 is a block diagram of the Timer0 module.

11.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

11.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the T0CS bit of the OPTION register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note: The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

11.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin or the Capacitive Sensing Oscillator (CPSOSC) signal.

8-Bit Counter mode using the T0CKI pin is selected by setting the T0CS bit in the OPTION register to '1' and resetting the T0XCS bit in the CPSCON0 register to '0'.

8-Bit Counter Mode using the Capacitive Sensing Oscillator (CPSOSC) signal is selected by setting the TOCS bit in the OPTION register to '1' and setting the T0XCS bit in the CPSCON0 register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the T0SE bit in the OPTION register.

FIGURE 11-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



12.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

12.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

Note:	The oscillator requires a start-up and							
	stabilization time before use. Thus,							
	T1OSCEN should be set and a suitable							
	delay observed prior to enabling Timer1.							

12.5 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see Section 12.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode").

Note: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

12.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	53
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	115
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	115
INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	36
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	37
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	52
TMR1H	Holding Reg	ister for the	Most Signifi	cant Byte of	the 16-bit T	MR1 Regis	ter		99
TMR1L	Holding Reg	ister for the	Least Signif	icant Byte o	f the 16-bit 1	MR1 Regis	ster		99
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	52
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR10N	103
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS1	T1GSS0	104

TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	_	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	53
APFCON	—	—	—		_	—	SSSEL	CCP2SEL	42
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	115
CCP2CON	—		DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	115
CCPRxL	Capture/Compare/PWM Register X Low Byte								116
CCPRxH	Capture/Compare/PWM Register X High Byte								116
PR2	Timer2 Period Register							106	
T2CON	_	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	107
TMR2	Timer2 Module Register								106
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	52
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62

TABLE 15-7: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	44
APFCON	—	—	—	—	—	_	SSSEL	CCP2SEL	42
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	36
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	37
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PR2	Timer2 Period Register								
SSPBUF	Synchrono	us Serial Po	rt Receive B	uffer/Transn	nit Register				147
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM2 SSPM1		152
SSPSTAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	153
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	43
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	107

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

START AND STOP CONDITIONS 17.2.2

During times of no data transfer (Idle time), both the clock line (SCL) and the data line (SDA) are pulled high through external pull-up resistors. The Start and Stop conditions determine the start and stop of data transmission. The Start condition is defined as a high-to-low transition of the SDA line while SCL is high. The Stop condition is defined as a low-to-high transition of the SDA line while SCL is high.

Figure 17-9 shows the Start and Stop conditions. A master device generates these conditions for starting and terminating data transfer. Due to the definition of the Start and Stop conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

ACKNOWLEDGE 17.2.3

After the valid reception of an address or data byte, the hardware automatically will generate the Acknowledge (ACK) pulse and load the SSPBUF register with the received value currently in the SSPSR register. There are certain conditions that will cause the SSP module not to generate this ACK pulse. They include any or all of the following:

- The Buffer Full bit, BF of the SSPSTAT register, was set before the transfer was received.
- The SSP Overflow bit. SSPOV of the SSPCON register, was set before the transfer was received.
- The SSP module is being operated in Firmware Master mode.

In such a case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF of the PIR1 register is set. Table 17-2 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

٦ SDA SCL S Р Change of Change of Data Allowed Data Allowed Start Stop Condition Condition

	TABLE 17-2:	DATA TRANSFER RECEIVED BYTE ACTIONS
--	-------------	-------------------------------------

Status Bits as Data Transfer is Received		$SSPSR \to SSPBUF$	Generate ACK	Set bit SSPIF (SSP Interrupt occurs		
BF	SSPOV		Fuise	if enabled)		
0	0	Yes	Yes	Yes		
1	0	No	No	Yes		
1	1	No	No	Yes		
0	1	No	No	Yes		

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

FIGURE 17-9: START AND STOP CONDITIONS

21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F722A/723A instruction set is highly orthogonal and is comprised of three basic categories:

- Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASMTM assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μ s. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTB instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RBIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 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.
d	Destination select; $d = 0$: store result in W, d = 1: store result in file register f. Default is $d = 1$.
PC	Program Counter
TO	Time-out bit
С	Carry bit
DC	Digit carry bit
Z	Zero bit
PD	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS



© 2010-2016 Microchip Technology Inc.

Mnemonic, Operands		Description			14-Bit	Opcode	•	Status	Notoo
				MSb			LSb	Affected	Notes
		BYTE-ORIENTED FILE REGIS	RATIO	NS					
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1, 2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1, 2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00	1011	dfff	ffff		1, 2, 3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00	1111	dfff	ffff		1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	С	1, 2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	С	1, 2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1, 2
		BIT-ORIENTED FILE REGIST		RATION	IS				
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1, 2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
		LITERAL AND CONTROL	OPERAT	IONS					1
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO, PD	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	TO, PD	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

TABLE 21-2: PIC16(L)F722A/723A INSTRUCTION SET

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTA, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.





TABLE 23-9: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$										
Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions				
US120	TCKH2DTV	SYNC XMIT (Master and Slave)	3.0-5.5V	—	80	ns				
		Clock high to data-out valid	1.8-5.5V	—	100	ns				
US121	TCKRF	Clock out rise time and fall time	3.0-5.5V	—	45	ns				
		(Master mode)	1.8-5.5V	—	50	ns				
US122	TDTRF	Data-out rise time and fall time	3.0-5.5V	_	45	ns				
			1.8-5.5V	—	50	ns				

FIGURE 23-15: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



TABLE 23-10: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Standar Operatir	Standard Operating Conditions (unless otherwise stated)Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$									
Param. No.	Symbol	Min.	Max.	Units	Conditions					
US125	TDTV2CKL	<u>SYNC RCV (Master and Slave)</u> Data-hold before CK ↓ (DT hold time)	10		ns					
US126	TCKL2DTL	Data-hold after CK \downarrow (DT hold time)	15		ns					

Param. No.	Symbol	Characteristic		Min.	Тур†	Max.	Units	Conditions
CS01	ISRC	Current Source	High	—	-5.8	-6	μΑ	
			Medium	—	-1.1	-3.2	μΑ	-40, -85°C
			Low	—	-0.2	-0.9	μΑ	
CS02	ISNK	Current Sink	High	—	6.6	6	μΑ	
			Medium	—	1.3	3.2	μΑ	-40, -85°C
			Low	—	0.24	0.9	μΑ	
CS03	VCHYST	Cap Hysteresis	High	—	525		mV	
			Medium	—	375	_	mV	VCTH-VCTL
			Low	_	280		mV	

TABLE 23-14: CAP SENSE OSCILLATOR SPECIFICATIONS

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



FIGURE 23-22: CAP SENSE OSCILLATOR







