



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

Decails	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	16MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	902 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c756at-16-l

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



Example 9-4 shows the sequence to do a 16 x 16 signed multiply. Equation 9-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pairs most significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 9-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

RES3:RES0	
$= ARG1H:ARG1L \bullet ARG2H:ARG2L$	
$= (ARG1H \bullet ARG2H \bullet 2^{16})$	+
$(ARG1H \bullet ARG2L \bullet 2^8)$	+
$(ARG1L \bullet ARG2H \bullet 2^8)$	+
$(ARG1L \bullet ARG2L)$	+
$(-1 \bullet ARG2H < 7 > \bullet ARG1H:ARG1L \bullet 2^{16})$	+
$(-1 \bullet ARG1H < 7 > \bullet ARG2H:ARG2L \bullet 2^{16})$	

EXAMPLE 9-4: 16 x 16 SIGNED MULTIPLY ROUTINE

	MOVFP	ARG1L, WREG		
	MULWF		;	ARG1L * ARG2L ->
				PRODH: PRODL
	MOVPF	PRODH, RES1		
		PRODL, RESO		
;			'	
'	MOVFP	ARG1H, WREG		
	MULWF	ARG2H		ARG1H * ARG2H ->
				PRODH: PRODL
	MOVPF	PRODH, RES3		1110211111022
		PRODL, RES2		
;		111022, 11202	'	
'	MOVFP	ARG1L, WREG		
	MULWF			ARG1L * ARG2H ->
			;	
	MOVFP	PRODL, WREG		I KODII I KODII
	ADDWF			Add cross
	MOVFP	PRODH, WREG		
				products
	CLRF		;	
		WREG, F RES3, F		
	ADDWFC	RESS, F	;	
;	MOVFP	ADCIN MDEC		
		ARG1H, WREG		
	MULWF	ARG2L		ARG1H * ARG2L ->
	MOUTED			PRODH: PRODL
	MOVFP	PRODL, WREG RES1, F		
	MOVFP			products
	ADDWFC	RES2, F	;	
	CLRF	WREG, F	;	
	ADDWFC	RES3, F	;	
;	DEFIC			
		AKGZA, /	;	ARG2H:ARG2L neg?
	GOTO			no, check ARG1
	MOVFP	ARG1L, WREG	;	
	SUBWF	RES2	;	
	MOVFP	ARG1H, WREG	;	
	SUBWFB	RES3		
;	CNI ADC1			
SI	GN_ARG1			ADCILLADCIL DC-2
	BTFSS	CONT_CODE		ARG1H:ARG1L neg?
	GOTO			no, done
	MOVFP	ARG2L, WREG	;	
	SUBWF	RES2	;	
	MOVFP	ARG2H, WREG	;	
	SUBWFB	KES3		
;				
CO	NT_CODE			
	:			
1				

13.1.3.1 PWM Periods

The period of the PWM1 output is determined by Timer1 and its period register (PR1). The period of the PWM2 and PWM3 outputs can be individually software configured to use either Timer1 or Timer2 as the timebase. For PWM2, when TM2PW2 bit (PW2DCL<5>) is clear, the time base is determined by TMR1 and PR1 and when TM2PW2 is set, the time base is determined by Timer2 and PR2. For PWM3, when TM2PW3 bit (PW3DCL<5>) is clear, the time base is determined by TMR1 and PR1, and when TM2PW3 is set, the time base is determined by Timer2 and PR2.

Running two different PWM outputs on two different timers allows different PWM periods. Running all PWMs from Timer1 allows the best use of resources by freeing Timer2 to operate as an 8-bit timer. Timer1 and Timer2 cannot be used as a 16-bit timer if any PWM is being used.

The PWM periods can be calculated as follows:

period of PWM1 = $[(PR1) + 1] \times 4TOSC$

period of PWM2 = $[(PR1) + 1] \times 4TOSC$ or $[(PR2) + 1] \times 4TOSC$

period of PWM3 =
$$[(PR1) + 1] \times 4TOSC$$
 or
 $[(PR2) + 1] \times 4TOSC$

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and the lower 2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>). Table 13-4 shows the maximum PWM frequency (FPWM), given the value in the period register.

The number of bits of resolution that the PWM can achieve depends on the operation frequency of the device as well as the PWM frequency (FPWM).

Maximum PWM resolution (bits) for a given PWM frequency:

$$= \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log\left(2\right)} \quad \text{bits}$$

where: FPWM = 1 / period of PWM

The PWMx duty cycle is as follows:

PWMx Duty Cycle = (DCx) x TOSC

where DCx represents the 10-bit value from PWxDCH:PWxDCL.

If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four Q-clocks (depending on the state of the PWxDCL<7:6> bits). For a duty cycle to be 100%, the PWxDCH value must be greater then the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR1 (or TMR2) overflows and a new PWM period begins, the master latch values are transferred to the slave latches and the PWMx pin is forced high.

Note:	For PW1DCH, PW1DCL, PW2DCH,							
	PW2DCL, PW3DCH and PW3DCL regis-							
	ters, a write operation writes to the "master							
	latches", while a read operation reads the							
	"slave latches". As a result, the user may							
	not read back what was just written to the							
	duty cycle registers (until transferred to							
	slave latch).							

The user should also avoid any "read-modify-write" operations on the duty cycle registers, such as: ADDWF PW1DCH. This may cause duty cycle outputs that are unpredictable.

TABLE 13-4:	PWM FREQUENCY vs.
	RESOLUTION AT 33 MHz

PWM	Frequency (kHz)								
Frequency	32.2	64.5	90.66	128.9	515.6				
PRx Value	0xFF	0x7F	0x5A	0x3F	0x0F				
High Resolution	10-bit	9-bit	8.5-bit	8-bit	6-bit				
Standard Resolution	8-bit	7-bit	6.5-bit	6-bit	4-bit				

13.1.3.2 PWM INTERRUPTS

The PWM modules make use of the TMR1 and/or TMR2 interrupts. A timer interrupt is generated when TMR1 or TMR2 equals its period register and on the following increment is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer rollover. The TMR1 interrupt is latched into the TMR1IF bit and the TMR2 interrupt is latched into the TMR2IF bit. These flags must be cleared in software.

13.2 Timer3

Timer3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a another 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (Fosc/4). When TMR3CS is set, the counter increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer/counter to increment. When TMR3ON is clear, the timer will not increment or set flag bit TMR3IF.

Timer3 has two modes of operation, depending on the CA1/PR3 bit (TCON2<3>). These modes are:

- Three capture and one period register mode
- · Four capture register mode

The PIC17C7XX has up to four 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are four capture pins

(RB0/CAP1, RB1/CAP2, RG4/CAP3, and RE3/CAP4), one for each capture register pair. The capture pins are multiplexed with the I/O pins. An event can be:

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

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture modules are truly part of the Timer3 block. Figure 13-5 and Figure 13-6 show the block diagrams for the two modes of operation.

13.2.1 THREE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode, registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 13-5. The timer increments until it equals the period register and then resets to 0000h on the next timer clock. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

FIGURE 13-5: TIMER3 WITH THREE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



14.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 14-2. The data comes in the RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit, RCIF, is set. The actual interrupt can be enabled/ disabled by setting/clearing the RCIE bit. RCIF is a read only bit which is cleared by the hardware. It is cleared when RCREG has been read and is empty. RCREG is a double buffered register (i.e., it is a twodeep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On detection of the STOP bit of the third byte, if the RCREG is still full, then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software which is done by reset-

FIGURE 14-5: RX PIN SAMPLING SCHEME

ting the receive logic (CREN is set). If the OERR bit is set, transfers from the RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The framing error bit FERR (RCSTA<2>) is set if a STOP bit is not detected.

Note: The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG register will allow the RX9D and FERR bits to be loaded with values for the next received data. Therefore, it is essential for the user to read the RCSTA register before reading RCREG, in order not to lose the old FERR and RX9D information.

14.2.3 SAMPLING

The data on the RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX/DT pin. The sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 14-5).

The x16 clock is a free running clock and the three sample points occur at a frequency of every 16 falling edges.



FIGURE 14-6: START BIT DETECT



The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address if the next byte is the completion of 10-bit address and if this will be a read or write data transfer.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON1<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1111 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

15.2.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs. The MSSP module will override the input state with the output data when required (slavetransmitter).

When an address is matched or the data transfer after an address match is received, the hardware automatically will generate the acknowledge (\overline{ACK}) pulse and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the MSSP module not to give this ACK pulse. These are if either (or both):

- a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- b) The overflow bit SSPOV (SSPCON1<6>) was set before the transfer was received.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are set. Table 15-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low time for proper operation. The high and low times of the I^2C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101 of the Electrical Specifications.







16.3 Configuring Analog Port Pins

The ADCON1, and DDR registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding DDR bits set (input). If the DDR bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the DDR bits.

- Note 1: When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.
 - 2: Analog levels on any pin that is defined as a digital input (including the AN15:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

16.4 A/D Conversions

Example 16-2 shows how to perform an A/D conversion. The PORTF and lower four PORTG pins are configured as analog inputs. The analog references (VREF+ and VREF-) are the device AVDD and AVSS. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the RG3/AN0 pin (channel 0).

Note:	
	the same instruction that turns on the A/D.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/ D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started.

In Figure 16-4, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

	MOVLB CLRF MOVLW	5 ADCON1, F 0x01	;	Bank 5 Configure A/D inputs, All analog, TAD = Fosc/8, left just. A/D is on, Channel 0 is selected
	MOVWF	ADCON0	;	
	MOVLB	4	;	Bank 4
	BCF	PIR2, ADIF	;	Clear A/D interrupt flag bit
	BSF	PIE2, ADIE	;	Enable A/D interrupts
	BSF	INTSTA, PEIE	;	Enable peripheral interrupts
	BCF	CPUSTA, GLINTD	;	Enable all interrupts
;				
; Er	sure that	at the required sam	np	ling time for the selected input channel has elapsed.
; Th	nen the o	conversion may be a	sta	arted.
;				
	MOVLB	5	;	Bank 5
	BSF	ADCON0, GO	;	Start A/D Conversion
	:		;	The ADIF bit will be set and the GO/DONE bit
	:		;	is cleared upon completion of the A/D Conversion

FIGURE 16-4: A/D CONVERSION TAD CYCLES



EXAMPLE 16-2: A/D CONVERSION



TABLE 18-2: PIC17CXXX INSTRUCTION SET

Mnemonic,		Description	Cycles		16-bit C	Opcode	Status		
Operands		Description		MSb	LSb		Affected	Notes	
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f,d	ADD WREG to f	1	0000	111d	ffff	ffff	OV,C,DC,Z	
ADDWFC	f,d	ADD WREG and Carry bit to f	1	0001	000d	ffff	ffff	OV,C,DC,Z	
ANDWF	f,d	AND WREG with f	1	0000	101d	ffff	ffff	Z	
CLRF	f,s	Clear f, or Clear f and Clear WREG	1	0010	100s	ffff	ffff	None	3
COMF	f,d	Complement f	1	0001	001d	ffff	ffff	Z	
CPFSEQ	f	Compare f with WREG, skip if f = WREG	1 (2)	0011	0001	ffff	ffff	None	6,8
CPFSGT	f	Compare f with WREG, skip if f > WREG	1 (2)	0011	0010	ffff	ffff	None	2,6,8
CPFSLT	f	Compare f with WREG, skip if f < WREG	1 (2)	0011	0000	ffff	ffff	None	2,6,8
DAW	f,s	Decimal Adjust WREG Register	1	0010	111s	ffff	ffff	С	3
DECF	f,d	Decrement f	1	0000	011d	ffff	ffff	OV,C,DC,Z	
DECFSZ	f,d	Decrement f, skip if 0	1 (2)	0001	011d	ffff	ffff	None	6,8
DCFSNZ	f,d	Decrement f, skip if not 0	1 (2)	0010	011d	ffff	ffff	None	6,8
INCF	f,d	Increment f	1	0001	010d	ffff	ffff	OV,C,DC,Z	
INCFSZ	f,d	Increment f, skip if 0	1 (2)	0001	111d	ffff	ffff	None	6,8
INFSNZ	f,d	Increment f, skip if not 0	1 (2)	0010	010d	ffff	ffff	None	6,8
IORWF	f,d	Inclusive OR WREG with f	1	0000	100d	ffff	ffff	Z	
MOVFP	f,p	Move f to p	1	011p	pppp	ffff	ffff	None	
MOVPF	p,f	Move p to f	1	010p	pppp	ffff	ffff	Z	
MOVWF	f	Move WREG to f	1	0000	0001	ffff	ffff	None	
MULWF	f	Multiply WREG with f	1	0011	0100	ffff	ffff	None	
NEGW	f,s	Negate WREG	1	0010	110s	ffff	ffff	OV,C,DC,Z	1,3
NOP	_	No Operation	1	0000	0000	0000	0000	None	
RLCF	f,d	Rotate left f through Carry	1	0001	101d	ffff	ffff	С	
RLNCF	f,d	Rotate left f (no carry)	1	0010	001d	ffff	ffff	None	
RRCF	f,d	Rotate right f through Carry	1	0001	100d	ffff	ffff	С	
RRNCF	f,d	Rotate right f (no carry)	1	0010	000d	ffff	ffff	None	
SETF	f,s	Set f	1	0010	101s	ffff	ffff	None	3
SUBWF	f,d	Subtract WREG from f	1	0000	010d	ffff	ffff	OV,C,DC,Z	1
SUBWFB	f,d	Subtract WREG from f with Borrow	1	0000	001d	ffff	ffff	OV,C,DC,Z	1
SWAPF	f,d	Swap f	1	0001	110d	ffff	ffff	None	
TABLRD	t,i,f	Table Read	2 (3)	1010	10ti	ffff	ffff	None	7
TABLWT	t,i,f	Table Write	2	1010	11ti	ffff	ffff	None	5
TLRD	t,f	Table Latch Read	1	1010	00tx	ffff	ffff	None	
TLWT	t,f	Table Latch Write	1	1010	01tx	ffff	ffff	None	

Legend: Refer to Table 18-1 for opcode field descriptions.

Note 1: 2's Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL).

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.

6: Two-cycle instruction when condition is true, else single cycle instruction.

7: Two-cycle instruction except for TABLRD to PCL (program counter low byte), in which case it takes 3 cycles.

8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead a NOP is executed.

ADD	OWFC	ADD WR	EG and C	arry bit	to f				
Syn	tax:	[label] A	DDWFC	f,d					
Ope	rands:	$\begin{array}{l} 0 \leq f \leq 25 \\ d \in [0,1] \end{array}$	$0 \le f \le 255$ $d \in [0,1]$						
Ope	ration:	(WREG) ·	+ (f) + C –	→ (dest)					
Stat	us Affected:	OV, C, D0	C, Z						
Enc	oding:	0001	000d	ffff	ffff				
Des	cription:	Add WREC memory lo placed in V placed in d	cation 'f'. If VREG. If 'd	'd' is 0, th ' is 1, the	e result is result is				
Wor	ds:	1							
Сус	les:	1							
QC	ycle Activity:								
	Q1	Q2	Q3		Q4				
	Decode	Read register 'f'	Proces Data	-	rite to tination				
<u>Exa</u>	mple:	ADDWFC	REG (D					
	Before Instru Carry bit REG WREG	= 1 = 0x02							
	After Instruct Carry bit REG								

AND	DLW	W And Literal with WREG								
Synt	ax:	[label] A	[<i>label</i>] ANDLW k							
Ope	rands:	$0 \le k \le 25$	55							
Ope	ration:	(WREG)	AND. (k	$() \rightarrow ()$	WR	EG)				
Statu	us Affected:	Z								
Enco	oding:	1011	0101	kkk	k	kkkk				
Description: The contents of WREG are AND'ed w the 8-bit literal 'k'. The result is placed WREG.										
Wor	ds:	1	1							
Cycl	es:	1								
QC	vcle Activity:									
	Q1	Q2	Q	3		Q4				
	Decode	Read literal 'k'	Proce Dat		-	Vrite to VREG				
<u>Exar</u>	<u>mple</u> :	ANDLW	0x5F							
	Before Instruction									

WREG = 0xA3 After Instruction WREG = 0x03

Carry bit	=	0
REG	=	0x02
WREG	=	0x50

PIC17C7XX

ANDWF	AND WR	EG with f	
Syntax:	[label] A	NDWF f,c	
Operands:	$\begin{array}{l} 0 \leq f \leq 25 \\ d \in [0,1] \end{array}$	5	
Operation:	(WREG) .	AND. (f) \rightarrow (dest)
Status Affected:	Z		
Encoding:	0000	101d ff	ff ffff
Description:	register 'f'.	If 'd' is 0 the re f 'd' is 1 the re	
Words:	1		
Cycles:	1		
Q Cycle Activity:			
Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
Example: Before Instru		REG, 1	

BCF		Bit Clear	f			
Synt	ax:	[label] E	BCF f,	b		
Ope	rands:	$\begin{array}{l} 0 \leq f \leq 25 \\ 0 \leq b \leq 7 \end{array}$	5			
Ope	ration:	$0 \rightarrow (f < b >$	•)			
Statu	us Affected:	None				
Enco	oding:	1000	1bbb	fff	f	ffff
Desc	cription:	Bit 'b' in reg	gister 'f' is	s clear	ed.	
Word	ds:	1				
Cycl	es:	1				
QC	cle Activity:					
	Q1	Q2	Q	3		Q4
	Decode	Read register 'f'	Proce Dat			Write gister 'f'
<u>Exar</u>	<u>mple</u> :	BCF	FLAG_F	REG,	7	

Before Instruction FLAG_REG = 0xC7

After Instruction FLAG_REG = 0x47

 $\begin{array}{rrrr} Before Instruction \\ WREG &= & 0x17 \\ REG &= & 0xC2 \\ \end{tabular} \\ After Instruction \\ WREG &= & 0x17 \end{array}$

REG = 0x02

TABLRD	Table Re	ead	
Example1:	TABLRD	1, 1,	REG ;
Before Instruc	ction		
REG		=	0x53
TBLATH		=	0xAA
TBLATL		=	0x55
TBLPTR		=	0xA356
MEMORY	(TBLPTR)	=	0x1234
After Instructi	on (table v	write co	
REG		=	0xAA
TBLATH		=	0x12
TBLATL		=	0x34
TBLPTR		=	0xA357
MEMORY	(TBLPTR)	=	0x5678
Example2:	TABLRD	0, 0,	REG ;
Before Instruc	ction		
REG		=	0x53
TBLATH		=	0xAA
TBLATL		=	0x55
TBLPTR		=	0xA356
MEMORY	(TBLPTR)	=	0x1234
After Instructi	on (table v	write co	mpletion)
REG		=	0x55
TBLATH		=	0x12
TBLATL		=	0x34
TBLPTR		=	0xA356
MEMORY	(TBLPTR)	=	0x1234

TABLWT	Table Write	
Syntax:	[label] TABLWT t,i,f	
Operands:	$0 \le f \le 255$	
	i ∈ [0,1] t ∈ [0,1]	
Operation:	lf t = 0,	
Operation.	$f \rightarrow TBLATL;$	
	If t = 1,	
	$f \rightarrow TBLATH;$ TBLAT $\rightarrow Prog Mem (TBLPTR)$	١.
	If $i = 1$,),
	TBLPTR + 1 \rightarrow TBLPTR	
	If i = 0, TBLPTR is unchanged	
Status Affected	-	
Encoding:	1010 11ti ffff ff	ff
Description:	1. Load value in 'f' into 16-bit tab	
	latch (TBLAT)	16
	If t = 1: load into high byte; If t = 0: load into low byte	
	2. The contents of TBLAT are wr	rit-
	ten to the program memo	ory
	location pointed to by TBLPTF If TBLPTR points to extern	
	program memory location, the	
	the instruction takes two-cycle) .
	the instruction takes two-cycle If TBLPTR points to an intern) .
	the instruction takes two-cycle If TBLPTR points to an intern EPROM location, then the instruction is terminated who	e. nal he
Note: The	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received.	e. hal he en
volta	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming of interview of the programming of interview.	e. hal he en
volta m <u>err</u>	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming of interview of the programming of interview.	e. hal he en
volta m <u>em</u> If M0 the p	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then th instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem-	e. hal he en ning rna
volta mem If M0 the p will	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur	e. hal he en ning rna
volta mem If M0 the p will	the instruction takes two-cycle If TBLPTR points to an intern EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not	e. hal he en ning rna
volta m <u>en</u> If M0 the p will I TCY)	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then th instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur the internal memory location will not sted. 3. The TBLPTR can be automa	e. hal he en ning rna nory t be
volta m <u>en</u> If M0 the p will I TCY)	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then th instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur the internal memory location will not sted. 3. The TBLPTR can be automa- cally incremented	e. hal he en ning rna nory t be
volta m <u>en</u> If M0 the p will I TCY)	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then th instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur the internal memory location will not sted. 3. The TBLPTR can be automa- cally incremented If i = 1; TBLPTR is not incremented	e. hal he en ning rna nory t be
volta m <u>em</u> If MC the p will I Tcy) affec	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then th instruction is terminated who an interrupt is received. <u>MCLR/VPP pin must be at the programming</u> ory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented	e. hal he en ning rna nory t be
volta mem If MC the p will Tcy) affec	the instruction takes two-cycle If TBLPTR points to an intern EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not cted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented	e. hal he en ning rna nory t be
volta mem If MC the p will Tcy) affec	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented 1 2 (many if write is to on-chip	e. hal he en ning rna nory t be
volta mem If MC the p will I Tcy) affec Words: Cycles:	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming rory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur ted. 3. The TBLPTR can be automaticated incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; I	e. hal he en ning rna nory t be
volta mem If MC the p will I Tcy) affec Words: Cycles:	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming rory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur ted. 3. The TBLPTR can be automaticated incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; I	e. hal he en ning rna nory t be
volta mem If MC the p will I Tcy) affec Words: Cycles: Q Cycle Activity	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not teted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented 1 2 (many if write is to on-chip EPROM program memory) /:	e. hal he en ning rna nory r (2 t be
volta mem If MC the p will Tcy) affec Words: Cycles: Q Cycle Activity Q1	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented Mean if write is to on-chip EPROM program memory) /: Q2 Q3 Q4 Read Process Write register 'f' Data register	e. hal he en nory r (2 t be tti-
volta mem If MC the p will Tcy) affec Words: Cycles: Q Cycle Activity Q1	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- nory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented 1 2 (many if write is to on-chip EPROM program memory) /: Q2 Q3 Q4 Read Process Write	e. hal he en nory r (2 t be tti- ed
volta mem If MC the p will Tcy) affec Words: Cycles: Q Cycle Activity Q1	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automaticated If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented MCLR/VPP = VDD	e. hal he en nory r (2 t be tti- ed
volta mem If MC the p will Tcy) affec Words: Cycles: Q Cycle Activity Q1 Decode	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automatic cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incremented If a contained in the second of the second term of the second o	e. hal he en ning rrna nor) r (2 t be nti-
Volta mem If MC the p will TCY) affec Words: Cycles: Q Cycle Activity Q1 Decode No	the instruction takes two-cycle If TBLPTR points to an interr EPROM location, then the instruction is terminated who an interrupt is received. MCLR/VPP pin must be at the programming ge for successful programming of inter- tory. CLR/VPP = VDD programming sequence of internal mem- be interrupted. A short write will occur . The internal memory location will not ted. 3. The TBLPTR can be automation cally incremented If i = 1; TBLPTR is not incremented If i = 0; TBLPTR is incremented If i = 0; TBLPTR is incre	e. hal he en ning rrna nory r (2 t be tti- ed or - n or

FIGURE 20-5: PARAMETER MEASUREMENT INFORMATION





TABLE 20-13: I²C BUS DATA REQUIREMENTS

Param No.	Sym	Character	istic	Min	Max	Units	Conditions
100	Thigh	Clock high time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
101	Tlow	Clock low time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
102	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns	Cb is specified to be from
			400 kHz mode	20 + 0.1Cb	300	ns	10 to 400 pF
			1 MHz mode ⁽¹⁾	-	300	ns	
103	Tf	SDA and SCL fall time	100 kHz mode	_	300	ns	Cb is specified to be from
			400 kHz mode	20 + 0.1Cb	300	ns	10 to 400 pF
			1 MHz mode ⁽¹⁾	-	10	ns	
90	Tsu:sta	START condition setup	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	Only relevant for Repeated
		time	400 kHz mode	2(Tosc)(BRG + 1)	—	ms	Start condition
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
91	Thd:sta	START condition hold	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	After this period, the first
		time	400 kHz mode	2(Tosc)(BRG + 1)	—	ms	clock pulse is generated
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
106	Thd:dat	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
			1 MHz mode ⁽¹⁾	0	—	ns	
107	Tsu:dat	Data input setup time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
			1 MHz mode ⁽¹⁾	100	—	ns	
92	Tsu:sto	STOP condition	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
		setup time	400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
109	Taa	Output valid from clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode ⁽¹⁾	_	400	ns	

Note 1: Maximum pin capacitance = 10 pF for all I^2C pins.

2: A fast mode (400 KHz) I²C bus device can be used in a standard mode I²C bus system, but the parameter # 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line. Parameter #102 + #107 = 1000 + 250 = 1250 ns (for 100 kHz mode) before the SCL line is released.

3: C_b is specified to be from 10-400pF. The minimum specifications are characterized with $C_b=10pF$. The rise time spec (t_r) is characterized with $R_p=R_p$ min. The minimum fall time specification (t_f) is characterized with $C_b=10pF$, and $R_p=R_p$ max. These are only valid for fast mode operation (VDD=4.5-5.5V) and where the SPM bit (SSPSTAT<7>) =1.)

4: Max specifications for these parameters are valid for falling edge only. Specs are characterized with R_p=R_p min and C_b=400pF for standard mode, 200pF for fast mode, and 10pF for 1MHz mode.



FIGURE 21-15: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD vs. Vdd (-40°C TO +125°C)





Bus Collision During a RESTART Condition	
Bus Collision During a START Condition	
Bus Collision During a STOP Condition	
Bus Collision Interrupt Enable, BCLIE	
Bus Collision Interrupt Flag bit, BCLIF	.38
C	
C11,	51
CA1/PR3 1	102
CA1ED0 1	101
CA1ED1 1	101
CA1IE	35
CA1IF	37
CA10VF1	102
CA2ED0	101
CA2ED1 1	101
CA2H	49
CA2IE	111
CA2IF	111
CA2L	49
CA2OVF	102
CA3H	
CA3IE	36
CA3IF	.38
CA3L	
CA4H	
CA4IE	36
CA4IF	
Calculating Baud Rate Error	
CALL	
Capacitor Selection	- • ·
Ceramic Resonators	18
Crystal Oscillator	
Capture	110
Capture Sequence to Read Example	113
Capture Sequence to Read Example1 Capture1	13
Capture1	
Capture1 Mode1	101
Capture1 Mode Overflow102,	101 103
Capture1 Mode Overflow	101 103
Capture1 Mode Overflow	101 103 .37
Capture1 Mode Overflow	101 103 .37
Capture1 Mode Overflow	101 103 . 37 101 103
Capture1 Mode	101 103 .37 101 103 .37
Capture1 Mode	101 103 .37 101 103 .37 .36
Capture1 Mode	101 103 .37 101 103 .37 .36 .38
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .36 .38
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .11
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .36 .38 .36 .38 .11
Capture1 Mode	101 103 .37 103 .36 .38 .36 .38 .38 .11 .17 .54
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .11 .17 .54 134
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .11 .17 .54 134 135
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .36 .38 .11 .17 .54 134 135 193
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .11 .17 .54 135 193 135
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .38 .11 .17 .54 135 135 .21
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .11 .17 .54 135 .21 .21
Capture1 Mode	101 103 .37 .01 103 .37 .36 .38 .37 .36 .38 .11 .17 .54 135 .21 .21 .207
Capture1 Mode	101 103 .37 .01 103 .37 .36 .38 .37 .36 .38 .11 .17 .54 135 .21 .21 .207
Capture1 Mode	101 103 .37 .01 103 .37 .36 .38 .36 .38 .11 .17 .54 135 .21 .21 207
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .38 .117 .54 135 .21 .207 208 .55
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .37 .54 135 .21 .207 208 .55 138
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .36 .38 .36 .38 .36 .38 .37 .54 135 .21 .207 208 .55 .38 .42
Capture1 Mode	101 103 .37 101 103 .37 .36 .38 .38 .38 .38 .38 .38 .38 .37 .36 .38 .37 .36 .38 .37 .36 .38 .37 .36 .38 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .37 .37 .36 .37 .37 .36 .37 .37 .36 .37 .37 .36 .37 .37 .36 .38 .37 .37 .36 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .36 .38 .37 .37 .37 .37 .36 .38 .37 .37 .37 .37 .37 .37 .37 .37 .37 .37
Capture1 Mode	101 103 .37 .01 103 .37 .36 .38 .38 .38 .38 .38 .38 .38 .38 .38 .38
Capture1 Mode	101 103 .37 .01 103 .37 .36 .38 .38 .38 .38 .38 .38 .38 .38 .38 .38

Configuration	
Bits	
Locations	192
Oscillator1	7, 192
Word	191
CPFSEQ	209
CPFSGT	209
CPFSLT	
CPUSTA	
Crystal Operation, Overtone Crystals	
Crystal or Ceramic Resonator Operation	
Crystal Oscillator	
, ,	17
D	
D/Ā	134
Data Memory	
GPR	43, 46
Indirect Addressing	54
Organization	
SFR	
Data Memory Banking	
Data/Address bit, D/A	
DAW	
DC	, -
DDRB	,
DDRC	,
DDRD	
DDRE	,
DDRF	49
DDRG	49
DECF	211
DECFSNZ	212
DECFSZ	
Delay From External Clock Edge	
Digit Borrow	
Digit Carry (DC)	
Duty Cycle	107
E	
Electrical Characteristics	
PIC17C752/756	
Absolute Maximum Ratings	239
Capture Timing	253
CLKOUT and I/O Timing	250
DC Characteristics	242
External Clock Timing	
Memory Interface Read Timing	
Memory Interface Write Timing	
Parameter Measurement Information	
Reset, Watchdog Timer, Oscillator Start-up	240
e 1	054
Timer and Power-up Timer Timing	
Timer0 Clock Timing	
Timer1, Timer2 and Timer3 Clock Timing	
Timing Parameter Symbology	
USART Module Synchronous Receive Timin	g. 261
USART Module Synchronous Transmission	
Timing	260
EPROM Memory Access Time Order Suffix	45
Errata	
Extended Microcontroller	
Extended Microcontroller Mode	
External Memory Interface	
External Program Memory Waveforms	
External royant wentury waveloning	40

PIC17C7XX

MOVPF
MOVWF
MPLAB Integrated Development Environment Software 233
MULLW
Multi-Master Communication 170
Multi-Master Mode 152
Multiply Examples
16 x 16 Routine68
16 x 16 Signed Routine69
8 x 8 Routine
8 x 8 Signed Routine
MULWF
Ν
NEGW
NOP
0
Opcode Field Descriptions
Opcodes
Oscillator
Configuration17, 192
Crystal
External Clock19
External Crystal Circuit19
External Parallel Resonant Crystal Circuit
External Series Resonant Crystal Circuit
RC20
RC Frequencies
Oscillator Start-up Time (Figure)
Oscillator Start-up Timer (OST)24
OST24
OV
Overflow (OV)
Ρ
•
P
P
P
P
P
P
P
P
P
P
P
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCLATH 56 PD 52, 194
P
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Interrupt Request (PIR1) 37
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 56 PCL 56 PD 52 PEIE 34 Peripheral Bank 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICmicro Demo Board 235
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 56 PCL 56 PD 52 PEIE 34 Peripheral Bank 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICmicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 235
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICInicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICSTART" Plus Entry Level Development System 235
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICInicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICSTART" Plus Entry Level Development System 235
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-1 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART" Plus Entry Level Development System 235 PIE 126, 130, 132 PIE1 28, 48
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART* Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Banks 46 PICDEM-1 Low-Cost PICmicro Demo Board 235 PICDEM-1 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICSTART** Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49 PIR 126, 130, 132
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART* Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Banks 46 PICDEM-1 Low-Cost PICmicro Demo Board 235 PICDEM-1 Low-Cost PIC16CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICSTART** Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49 PIR 126, 130, 132
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PCL 56, 198 PCLATH 56 PD 52, 194 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Banks 57 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICMicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXXX Demo Board 235 PIE 28, 48 PIE2 28, 36, 49 PIR 28, 48 PIR2 28, 48
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PD 56 PD 52, 194 PEIE 34, 111 PEIF 34, 111 PEIF 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Interrupt Request (PIR1) 37 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICI6CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART [®] Plus Entry Level Development System 235 PIE 126, 130, 132 PIE1 28, 48 PIE2 28, 36, 49 PIR 28, 48 PIR2 28, 48 PIR2 28, 48 PIR2 28, 49 PM0 191, 195
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART** Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49 PIR 126, 130, 132 PIR1 28, 48 PIR2 28, 48 PIR2 28, 48 PIR2 28, 49 PM0 191, 195
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PD 56 PD 52, 194 PEIE 34, 111 PEIF 34, 111 PEIF 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Interrupt Request (PIR1) 37 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICI6CXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART [®] Plus Entry Level Development System 235 PIE 126, 130, 132 PIE1 28, 48 PIE2 28, 36, 49 PIR 28, 48 PIR2 28, 48 PIR2 28, 48 PIR2 28, 49 PM0 191, 195
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Interrupt Request (PIR1) 37 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICIficOXX Demo Board 235 PICDEM-3 Low-Cost PIC16CXXX Demo Board 236 PICSTART [®] Plus Entry Level Development System 235 PIE 126, 130, 132 PIE1 28, 48 PIE2 28, 36, 49 PIR 28, 48 PIR2 28, 48 PIR2 28, 48 PIR2 28, 49 PM0 191, 195
P 134 Packaging Information 281 PC (Program Counter) 56 PCFG0 bit 180 PCFG1 bit 180 PCFG2 bit 180 PCFG2 bit 180 PCH 56 PD 52 PEIE 34, 111 PEIF 34 Peripheral Bank 57 Peripheral Banks 57 Peripheral Interrupt Enable 35 Peripheral Register Banks 46 PICDEM-1 Low-Cost PICnicro Demo Board 235 PICDEM-2 Low-Cost PIC16CXX Demo Board 236 PICDEM-3 Low-Cost PIC16CXX Demo Board 236 PICSTART** Plus Entry Level Development System 235 PIE1 28, 48 PIE2 28, 36, 49 PIR 126, 130, 132 PIR1 28, 48 PIR2 28, 48 PIR2 28, 48 PIR2 28, 49 PM0 191, 195

PORTB PORTB Interrupt on Change			
PORTC			
PORTD			
PORTE	20,	-10, ∕10	82
PORTF			
PORTG			
Power-down Mode			
Power-on Reset (POR)			24
Power-up Timer (PWRT)			
PR1			
PR2		28,	49
PR3/CA1H			
PR3/CA1L			28
PR3H/CA1H			
PR3L/CA1L			
Prescaler Assignments			
PRO MATE" II Universal Programmer			
PRODH			
PRODL			
Program Counter (PC)	•••••		56
Program Memory			
External Access Waveforms			
External Connection Diagram			45
Мар			43
Modes			
Extended Microcontroller			43
Microcontroller			
Microprocessor			
			.0
Protected Microcontroller			43
Protected Microcontroller			
Operation			43
Operation Organization			43 43
Operation Organization Protected Microcontroller		 	43 43 43
Operation Organization Protected Microcontroller PS0		 53,	43 43 43 97
Operation Organization Protected Microcontroller PS0 PS1		 53, 53,	43 43 43 97 97
Operation Organization Protected Microcontroller PS0 PS1 PS2		 53, 53, 53,	43 43 43 97 97 97
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3		 53, 53, 53, 53,	43 43 97 97 97 97
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH		 53, 53, 53, 53, 39,	43 43 97 97 97 97 54
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH		 53, 53, 53, 53, 39, 28,	43 43 97 97 97 97 54 49
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH		 53, 53, 53, 53, 39, 28,	43 43 97 97 97 97 54 49
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH		 53, 53, 53, 53, 39, 28, 28,	43 43 97 97 97 97 54 49 49
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH		 53, 53, 53, 53, 39, 28, 28, 28,	43 43 97 97 97 97 54 49 49
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL		 53, 53, 53, 53, 39, 28, 28, 28, 28,	43 43 97 97 97 97 54 49 49 49
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCH PW3DCH		 53, 53, 53, 53, 39, 28, 28, 28, 28, 30,	43 43 97 97 97 54 49 49 49 50
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCL PW3DCL		 53, 53, 53, 53, 28, 28, 28, 28, 30, 30,	43 43 97 97 97 97 54 49 49 49 50 50
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCL PW3DCL PW3DCL PW4	10	 53, 53, 53, 53, 39, 28, 28, 28, 28, 30, 30, 30, 1, 1	43 43 97 97 97 97 54 49 49 49 50 50 07
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle	. 10	 53, 53, 53, 53, 28, 28, 28, 28, 30, 30, 1, 1	43 43 97 97 97 54 49 49 49 50 50 07 08
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source	. 10	 53, 53, 53, 53, 53, 39, 28, 28, 28, 30, 30, 1, 1 1 1	43 43 97 97 97 54 49 49 49 50 50 07 08 09
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution	. 10	 53, 53, 53, 53, 39, 28, 28, 28, 30, 30, 1, 1 1 1 1	43 43 97 97 97 97 54 49 49 49 50 07 08 09 08
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts	. 10	 53, 53, 53, 53, 53, 28, 28, 28, 28, 30, 30, 1, 1 1 1 1	43 43 97 97 97 54 49 49 49 50 50 07 08 09 08 08
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock		 53, 53, 53, 53, 53, 28, 28, 28, 28, 30, 30, 1, 1 1 1 1 1 1	43 43 97 97 97 54 49 49 49 50 07 08 09 08 08 09
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCL PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output	. 10 Inpu	 53, 53, 53, 53, 53, 53, 39, 28, 28, 28, 30, 30, 1, 1 1 1 1 1 1 1	43 43 97 97 97 54 49 49 49 50 07 08 09 08 09 07
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods	. 10 Inpu	 53, 53, 53, 53, 53, 53, 39, 28, 28, 28, 30, 30, 1, 1 1 1 1 1 1 1	43 43 97 97 97 97 54 49 49 49 50 07 08 09 08 09 07 08
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCH PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1	. 10 . 10	 53, 53, 53, 53, 53, 53, 53, 53, 53, 53,	43 43 97 97 97 97 97 97 97 97 97 97 97 97 97
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCL PW1DCL PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1 PWM1	. 10 . 10 . 10	 53, 53, 53, 53, 53, 53, 53, 53, 53, 53,	43 43 97 97 97 54 49 49 50 50 07 08 09 08 08 09 07 08 03 07
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCH PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1	. 10 . 10 . 10	 53, 53, 53, 53, 53, 53, 53, 53, 53, 53,	43 43 97 97 97 54 49 49 50 50 07 08 09 08 08 09 07 08 03 07
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCL PW1DCL PW2DCL PW2DCL PW3DCL PW3DCL PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1 PWM1	. 10 . 10 . 10. . 10. . 10.	 53, 53, 53, 53, 53, 53, 53, 53, 53, 53,	43 43 97 97 97 54 49 49 50 50 07 08 09 08 09 07 08 03 07 03
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCH PW1DCL PW2DCH PW2DCL PW3DCH PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1 PWM2	. 10 . 10 . 10 . 10 . 10 . 10 . 10	 53, 53, 53, 53, 53, 53, 53, 53, 53, 53,	$\begin{array}{c} 43\\ 43\\ 97\\ 97\\ 97\\ 54\\ 49\\ 49\\ 50\\ 50\\ 07\\ 08\\ 09\\ 08\\ 09\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ \end{array}$
Operation Organization Protected Microcontroller PS0 PS1 PS2 PS3 PUSH PW1DCL PW1DCL PW2DCH PW2DCL PW3DCH PW3DCL PW3DCL PWM Duty Cycle External Clock Source Frequency vs. Resolution Interrupts Max Resolution/Frequency for External Clock Output Periods PWM1 PWM2 PWM2 PWM2ON	. 10 . 10 . 10 . 10 . 10 . 10	 53, 53, 53, 53, 53, 53, 28, 28, 28, 28, 28, 30, 1, 1 1 1 2, 1 2, 1 2, 1 2, 1 2, 1 2, 1 2, 1 2,	$\begin{array}{r} 43\\ 43\\ 97\\ 97\\ 57\\ 49\\ 49\\ 49\\ 49\\ 50\\ 07\\ 08\\ 09\\ 08\\ 09\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 07\\ 03\\ 03\\ 07\\ 03\\ 07\\ 03\\ 03\\ 07\\ 03\\ 03\\ 07\\ 03\\ 03\\ 07\\ 03\\ 03\\ 03\\ 03\\ 03\\ 03\\ 03\\ 03\\ 03\\ 03$

PIC17C7XX

SEEVAL Evaluation and Programming System236
Serial Clock, SCK
Serial Clock, SCL
Serial Data Address, SDA
Serial Data In, SDI
Serial Data Out, SDO 137
SETF
SFR198
SFR (Special Function Registers)43
SFR As Source/Destination
Signed Math 11
Slave Select Synchronization
Slave Select, SS
SLEEP
SLEEP Mode, All Peripherals Disabled
SLEEP Mode, BOR Enabled
SMP
Software Simulator (MPLAB SIM)
SPBRG
SPBRG1
SPBRG2
SPE
Special Features of the CPU
Summary
SPI
Master Mode 139
Serial Clock
Serial Data In
Serial Data Out
Serial Peripheral Interface (SPI)
Slave Select
SPI clock
SPI Mode
SPI Clock Edge Select, CKE
SPI Data Input Sample Phase Select, SMP 134
SPI Master/Slave Connection
SPI Module
Master/Slave Connection
Slave Mode140
Slave Select Synchronization140
Slave Synch Timing 140
<u>SS</u> 137
SSP133
Block Diagram (SPI Mode)137
SPI Mode 137
SSPADD 144, 145
SSPBUF
SSPCON1
SSPCON2
SSPSR
SSPSTAT
SSP I ² C Operation
SSP Module SPI Master Mode
SPI Master Mode
SPI Master/Slave Connection
SSPCON1 Register
SSP Overliow Detect bit, SSPOV
SSFADD
SSFB0F
SSPCON1
SSPEN

SSPIE	
SSPIF	. 38, 145
SSPM3:SSPM0	135
SSPOV 135,	144, 162
SSPSTAT	
ST Input	278
Stack	
Operation	
Pointer	
Stack	
START bit (S)	
START Condition Enabled bit, SAE	
STKAV	
STOP bit (P)	
STOP Condition Enable bit	136
SUBLW	
SUBWF	
SUBWFB	
SWAPF	
Synchronous Master Mode	
Synchronous Master Reception	
Synchronous Master Transmission	107
Synchronous Serial Port	
Synchronous Serial Port Enable bit, SSPEN	
Synchronous Serial Port Interrupt	
Synchronous Serial Port Interrupt Enable, SSPIE	
Synchronous Serial Port Mode Select bits,	
SSPM3:SSPM0	
Synchronous Slave Mode	131
Т	
TOCKI	39
T0CKI Pin	40
T0CKIE	34
	••••••
TOCKIF	
TOCKIF	
	34 53, 97
TOCS	34 53, 97 34
TOCS TOIE	34 53, 97 34 34
T0CS T0IE T0IF	34 53, 97 34 34 53, 97
T0CS	
TOCS	34 53, 97 34 34 53, 97 53 101
TOCS	
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer	
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read	34 53, 97 34 53, 97 53 101 55 55
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example	34 53, 97 34 34 53, 97 53 101 55 55 64
TOCS TOIE TOIF TOSE TOSTA Table Latch Table Pointer Table Read Example Table Reads Section	34 53, 97 34 34 53, 97 53 101 55 55 64 64
TOCS	34 53, 97 34 34 53, 97 53 101 55 55 64 64
TOCS	34
TOCS	34
TOCS TOIE TOIF TOSE TOSTA Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing	34 53, 97 34 53, 97 53 101 55 55 64 64 64 62 62
TOCS	34
TOCS	34
TOCS	34 53, 97 34 53, 97 53 101 55 55 64 64 64 64 62 62 62 227, 228 228, 229
TOCS	34 53, 97 34 53, 97 53 101 55 55 64 64 64 64 62 62 62 227, 228 228, 229 185
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLATL	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLATL TBLPTRH	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLATL TBLPTRH TBLPTRL TCLK12	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLPTRH TCLK12 TCLK3	34
TOCS TOIE TOIF TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLPTRH TBLPTRH TCLK12 TCON1	34
TOCS TOIE TOIF TOSE TOSTA Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TAL TBLATH TBLATH TBLPTRH TCLK12 TCON1 TCON2	34
TOCS TOIE TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLRD TABLATL TBLATH TBLPTRH TCLK12 TCON1 TCON2 TCON2,TCON3	34
TOCS TOIE TOIF TOSE TOSTA Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLWT TAD TBLATH TBLATH TBLPTRH TBLPTRL TCLK12 TCON1 TCON2 TCON3	34
TOCS TOIE TOSE TOSTA T16 Table Latch Table Pointer Table Read Example Table Reads Section TLRD Table Write Code Timing To External Memory TABLRD TABLRD TABLATL TBLATH TBLPTRH TCLK12 TCON1 TCON2 TCON2,TCON3	34

 \odot 1998-2013 Microchip Technology Inc.