

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf8520t-i-pt

Email: info@E-XFL.COM

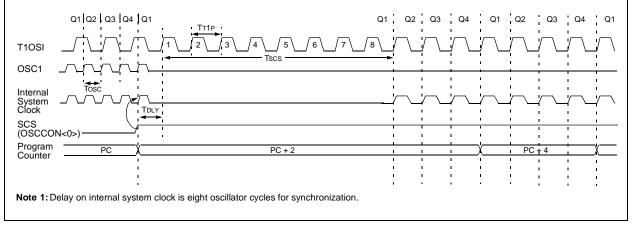
Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

2.6.2 OSCILLATOR TRANSITIONS

PIC18FXX20 devices contain circuitry to prevent "glitches" when switching between oscillator sources. Essentially, the circuitry waits for eight rising edges of the clock source that the processor is switching to. This ensures that the new clock source is stable and that its pulse width will not be less than the shortest pulse width of the two clock sources.

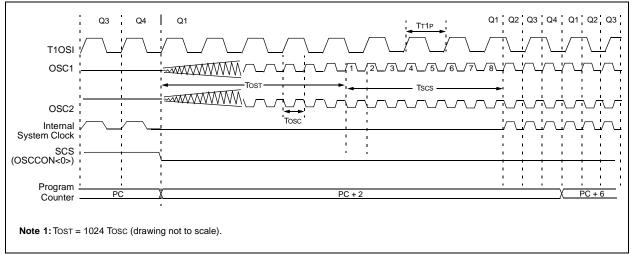
A timing diagram indicating the transition from the main oscillator to the Timer1 oscillator is shown in Figure 2-8. The Timer1 oscillator is assumed to be running all the time. After the SCS bit is set, the processor is frozen at the next occurring Q1 cycle. After eight synchronization cycles are counted from the Timer1 oscillator, operation resumes. No additional delays are required after the synchronization cycles.





The sequence of events that takes place when switching from the Timer1 oscillator to the main oscillator will depend on the mode of the main oscillator. In addition to eight clock cycles of the main oscillator, additional delays may take place. If the main oscillator is configured for an external crystal (HS, XT, LP), then the transition will take place after an oscillator start-up time (TOST) has occurred. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for HS, XT and LP modes, is shown in Figure 2-9.





4.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The data memory map is in turn divided into 16 banks of 256 bytes each. The lower 4 bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper 4 bits of the BSR are not implemented.

The data memory space contains both Special Function Registers (SFR) and General Purpose Registers (GPR). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratch pad operations in the user's application. The SFRs start at the last location of Bank 15 (0FFFh) and extend downwards. Any remaining space beyond the SFRs in the Bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as '0's.

PIC18FX520 devices have 2048 bytes of data RAM, extending from Bank 0 to Bank 7 (000h through 7FFh). PIC18FX620 and PIC18FX720 devices have 3840 bytes of data RAM, extending from Bank 0 to Bank 14 (000h through EFFh). The organization of the data memory space for these devices is shown in Figure 4-6 and Figure 4-7.

The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of a File Select Register (FSRn) and a corresponding Indirect File Operand (INDFn). Each FSR holds a 12-bit address value that can be used to access any location in the data memory map without banking.

The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing, or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction that moves a value from one register to another.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM. **Section 4.10** "Access Bank" provides a detailed description of the Access RAM.

4.9.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly. Indirect addressing operates using a File Select Register and corresponding Indirect File Operand. The operation of indirect addressing is shown in Section 4.12 "Indirect Addressing, INDF and FSR Registers".

Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

Data RAM is available for use as General Purpose Registers by all instructions. The top section of Bank 15 (F60h to FFFh) contains SFRs. All other banks of data memory contain GPR registers, starting with Bank 0.

4.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 4-2 and Table 4-3.

The SFRs can be classified into two sets: those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature. The SFRs are typically distributed among the peripherals whose functions they control.

The unused SFR locations are unimplemented and read as '0's. The addresses for the SFRs are listed in Table 4-2.

REGISTER 5-1:	EECON1 F	REGISTER	(ADDRE	SS FA6h)						
	R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0		
	EEPGD	CFGS	_	FREE	WRERR	WREN	WR	RD		
	bit 7							bit 0		
bit 7	EEPGD: F	EEPGD: Flash Program or Data EEPROM Memory Select bit								
	1 = Acces	1 = Access Flash program memory								
bit 6		 0 = Access data EEPROM memory CFGS: Flash Program/Data EEPROM or Configuration Select bit 								
		 1 = Access configuration registers 0 = Access Flash program or data EEPROM memory 								
bit 5	Unimplem	ented: Read	d as '0'							
bit 4	FREE: Flas	sh Row Eras	se Enable b	oit						
	(cleare	 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation) 0 = Perform write only 								
bit 3	WRERR: F	lash Progra	m/Data EE	PROM Error	[·] Flag bit					
	 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation) 0 = The write operation completed 									
	Note:	When a Wi tracing of th			GD and CFGS	bits are not	t cleared. T	his allows		
bit 2	WREN: Fla	ash Program	/Data EEP	ROM Write I	Enable bit					
		•	•	rogram/data program/data						
bit 1	WR: Write	Control bit								
	 1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.) 0 = Write cycle to the EEPROM is complete 									
bit 0		•		bompiete						
bit 0	1 = Initiate can or	 RD: Read Control bit 1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bic can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.) 0 = Does not initiate an EEPROM read 								
	Legend:]		
	R = Reada	ble bit	W = V	Vritable bit	U = Unimp	lemented b	it. read as '	0'		
					-					

'1' = Bit is set

'0' = Bit is cleared

- n = Value at POR

x = Bit is unknown

EXAMPLE 5-3: WRITING TO FLASH PROGRAM MEMORY

EXAMPLE 5-3:	WRI	TING TO FLASH PROC	
	MOVLW	D'64	; number of bytes in erase block
	MOVWF	COUNTER	
	MOVLW	BUFFER_ADDR_HIGH	; point to buffer
	MOVWF MOVLW	FSR0H BUFFER ADDR LOW	
	MOVWF	FSROL	
	MOVLW	CODE_ADDR_UPPER	; Load TBLPTR with the base
	MOVWF	TBLPTRU	; address of the memory block
	MOVLW	CODE_ADDR_HIGH	
	MOVWF MOVLW	TBLPTRH CODE ADDR LOW	
	MOVHW	CODE_ADDR_LOW TBLPTRL	
READ_BLOCK			
	TBLRD*+	÷	; read into TABLAT, and inc
	MOVF	TABLAT, W	; get data
	MOVWF	POSTINCO COUNTER	; store data ; done?
	BRA	READ BLOCK	; repeat
MODIFY_WORD			
_	MOVLW	DATA_ADDR_HIGH	; point to buffer
	MOVWF	FSROH	
	MOVLW MOVWF	DATA_ADDR_LOW FSR0L	
	MOVWF MOVLW	NEW DATA LOW	; update buffer word
	MOVWF	POSTINC0	, apaaoo barror nora
	MOVLW	NEW_DATA_HIGH	
	MOVWF	INDF0	
ERASE_BLOCK	MOLITIN		
	MOVLW MOVWF	CODE_ADDR_UPPER TBLPTRU	; load TBLPTR with the base ; address of the memory block
	MOVLW	CODE ADDR HIGH	, address of the memory brock
	MOVWF	TBLPTRH	
	MOVLW	CODE_ADDR_LOW	
	MOVWF	TBLPTRL	
	BSF BCF	EECON1, EEPGD EECON1, CFGS	; point to Flash program memory ; access Flash program memory
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Row Erase operation
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	55h	
Required	MOVWF MOVLW	EECON2 AAh	; write 55H
Sequence	MOVWF	EECON2	; write AAH
-	BSF	EECON1, WR	; start erase (CPU stall)
	NOP		
	BSF TDIDD+	INTCON, GIE	; re-enable interrupts ; dummy read decrement
WRITE BUFFER B	TBLRD*- BACK	-	; dummy read decrement
	MOVLW	8	; number of write buffer groups of 8 bytes
	MOVWF	COUNTER_HI	
	MOVLW	BUFFER_ADDR_HIGH	; point to buffer
	MOVWF	FSROH	
	MOVLW MOVWF	BUFFER_ADDR_LOW FSR0L	
PROGRAM LOOP	110 V W1	IDROL	
	MOVLW	8	; number of bytes in holding register
	MOVWF	COUNTER	
WRITE_WORD_TO_	-		
	MOVFF	POSTINCO, WREG	; get low byte of buffer data ; present data to table latch
	TBLWT+*	•	; present data to table latch ; write data, perform a short write
			; to internal TBLWT holding register.
	DECFSZ	COUNTER	; loop until buffers are full
	BRA	WRITE_WORD_TO_HREGS	
r			

FIGURE 10-6: BLOCK DIAGRAM OF RB2:RB0 PINS

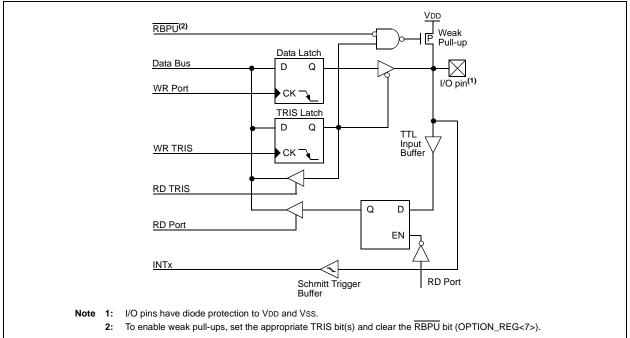
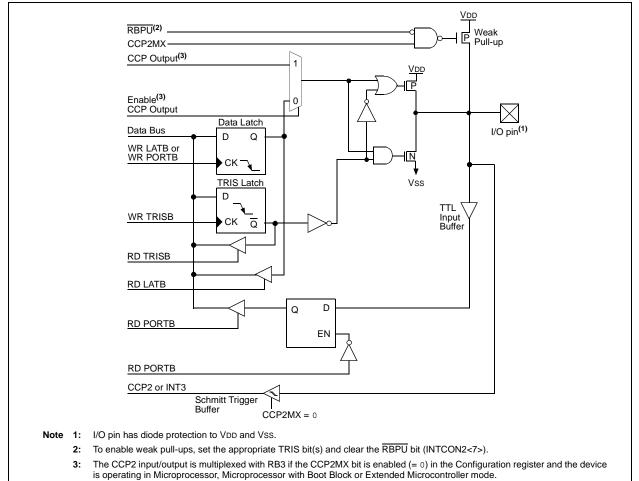


FIGURE 10-7: BLOCK DIAGRAM OF RB3 PIN



10.10 Parallel Slave Port

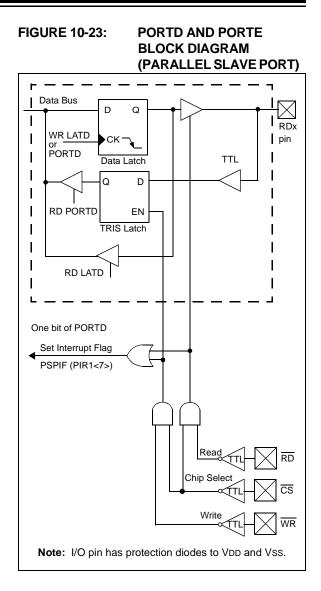
PORTD also operates as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (PSPCON<4>) is set. It is asynchronously readable and writable by the external world through the RD control input pin, RE0/RD/AD8 and the WR control input pin, RE1/WR/AD9.

Note:	For PIC18F8X20 devices, th	devices, the Paralle			
	Slave Port is available	only	in		
	Microcontroller mode.				

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/RD/AD8 to be the RD input, RE1/WR/AD9 to be the WR input and RE2/ CS/AD10 to be the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port configuration bits, PCFG2:PCFG0 (ADCON1<2:0>), must be set which will configure pins RE2:RE0 as digital I/O.

A write to the PSP occurs when both the \overline{CS} and \overline{WR} lines are first detected low. A read from the PSP occurs when both the \overline{CS} and \overline{RD} lines are first detected low.

The PORTE I/O pins become control inputs for the microprocessor port when bit PSPMODE (PSPCON<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs) and the ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.



REGISTER 17-5:	SSPCON	2: MSSP CO	NTROL R	EGISTER 2	(I ² C MOD	E)					
	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0			
	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN			
	bit 7							bit 0			
bit 7	GCEN: G	eneral Call En	able bit (Sla	ve mode onl	y)						
		 1 = Enable interrupt when a general call address (0000h) is received in the SSPSR 0 = General call address disabled 									
bit 6	ACKSTAT	F: Acknowledg	e Status bit	(Master Tran	smit mode c	only)					
		1 = Acknowledge was not received from slave0 = Acknowledge was received from slave									
bit 5	ACKDT: A	Acknowledge [Data bit (Ma	ster Receive	mode only)						
		1 = Not Acknowledge 0 = Acknowledge									
	Note:	Value that w the end of a		itted when th	e user initiat	tes an Ackı	nowledge s	equence at			
bit 4	ACKEN:	Acknowledge	Sequence E	nable bit (Ma	aster Receiv	e mode on	ly)				
	Autor	te Acknowledg matically clear owledge seque	ed by hardw		SCL pins a	nd transmi	t ACKDT da	ata bit.			
bit 3	RCEN: R	eceive Enable	bit (Master	mode only)							
	1 = Enabl 0 = Recei	es Receive mo ve Idle	ode for I ² C								
bit 2	-	p Condition En									
		e Stop conditio condition Idle	on on SDA a	ind SCL pins	. Automatica	ally cleared	by hardwa	re.			
bit 1		epeated Start (-							
		e Repeated St ated Start con		on SDA and	SCL pins. A	utomaticall	y cleared by	hardware.			
bit 0	SEN: Start Condition Enabled/Stretch Enabled bit										
In Master mode: 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware 0 = Start condition Idle In Slave mode: 1 = Clock stretching is enabled for both Slave Transmit and Slave Receive (stretch e 0 = Clock stretching is disabled								re.			
								n enabled)			
Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I ² C module is not in the Idle this bit may not be set (no spooling) and the SSPBUF may not be written (or to the SSPBUF are disabled).											
	Logondi										
	Legend:	ahla hit	$\lambda \lambda I = \lambda \lambda I$	ritable hit	II – Unimr	lamented	hit read as	' O'			

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

17.4.3.2 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low (ACK).

When the address byte overflow condition exists, then the no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

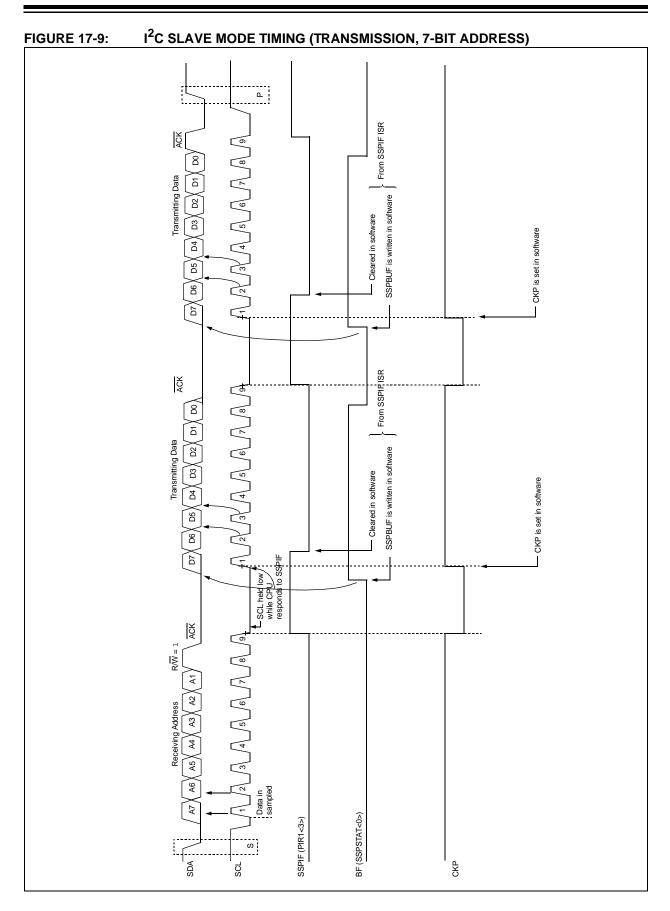
If SEN is enabled (SSPCON1<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit CKP (SSPCON<4>). See **Section 17.4.4** "**Clock Stretching**" for more detail.

17.4.3.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low, regardless of SEN (see Section 17.4.4 "Clock Stretching", for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/ SCK/SCL should be enabled by setting bit CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 17-9).

The ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not ACK), then the data transfer is complete. In this case, when the ACK is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low (ACK), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.



17.4.4 CLOCK STRETCHING

Both 7- and 10-bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

17.4.4.1 Clock Stretching for 7-bit Slave Receive Mode (SEN = 1)

In 7-bit Slave Receive mode, <u>on the falling edge of the</u> ninth clock at the end of the ACK sequence, if the BF bit is set, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 17-13).

- Note 1: If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.
 - 2: The CKP bit can be set in software, regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence, in order to prevent an overflow condition.

17.4.4.2 Clock Stretching for 10-bit Slave Receive Mode (SEN = 1)

In 10-bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/W bit cleared to '0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence, as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

17.4.4.3 Clock Stretching for 7-bit Slave Transmit Mode

7-bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock, if the BF bit is clear. This occurs, regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 17-9).

- Note 1: If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.
 - 2: The CKP bit can be set in software, regardless of the state of the BF bit.

17.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled as in 7-bit Slave Transmit mode (see Figure 17-11).

18.4 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the TXx pin (RC6/TX1/CK1 or RG1/TX2/CK2), instead of being supplied internally in Master mode. TRISC<6> must be set for this mode. This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit CSRC (TXSTAx<7>).

18.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical, except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- a) The first word will immediately transfer to the TSR register and transmit.
- b) The second word will remain in TXREG register.
- c) Flag bit TXxIF will not be set.
- d) When the first word has been shifted out of TSR, the TXREGx register will transfer the second word to the TSR and flag bit TXxIF will now be set.
- e) If enable bit TXxIE is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- 2. Clear bits CREN and SREN.
- 3. If interrupts are desired, set enable bit TXxIE.
- 4. If 9-bit transmission is desired, set bit TX9.
- 5. Enable the transmission by setting enable bit TXEN.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- 7. Start transmission by loading data to the TXREGx register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INTOIF	RBIF	0000 0000	0000 0000
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0111 1111	0111 1111
PIR3	_	_	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	00 0000	00 0000
PIE3	_		RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	00 0000	00 0000
IPR3	_	_	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	11 1111	11 1111
RCSTAx ⁽¹⁾	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
TXREGx ⁽¹⁾ USART Transmit Register									0000 0000	0000 0000
TXSTAx ⁽¹⁾	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRGx ⁽¹⁾ Baud Rate Generator Register									0000 0000	0000 0000

TABLE 18-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

Note 1: Register names generically refer to both of the identically named registers for the two USART modules, where 'x' indicates the particular module. Bit names and Reset values are identical between modules.

BTG	Bit Toggle	e f		BOV	Branch if	Overflow				
Syntax:	[label] B	TG f,b[,a]		Syntax:	[<i>label</i>] B	[<i>label</i>] BOV n				
Operands:	$0 \le f \le 255$	5		Operands:	-128 ≤ n ≤	-128 ≤ n ≤ 127				
	0 ≤ b < 7 a ∈ [0,1]			Operation:		if Overflow bit is '1' (PC) + 2 + 2n \rightarrow PC				
Operation:	$(f < b >) \rightarrow f$			Status Affected:	None					
Status Affected:	None			Encoding:	1110	0100 nn	nn nnnn			
Encoding:	0111	bbba f	fff ffff	Description:	If the Ove	rflow bit is '1	' then the			
Description:	inverted. If will be sele value. If 'a	f 'a' is '0', the ected, overri	location 'f' is Access Bank ding the BSR le bank will be R value		The 2's co added to t have incre instruction PC+2+2n.	he PC. Sind emented to f , the new ac	umber '2n' is the PC will etch the next dress will be ction is then			
Words:	1			Words:	1		•			
Cycles:	1									
Q Cycle Activity:				Cycles:	1(2)					
Q1	Q2	Q3	Q4	Q Cycle Activity If Jump:						
Decode	Read register 'f'	Process Data	Write register 'f'	Q1	Q2	Q3	Q4			
Example:	BTG F	PORTC, 4,	0	Decode	Read literal 'n'	Process Data	Write to PC			
Before Instru				No	No	No	No			
PORTC		101 [0x75]		operation	operation	operation	operation			
After Instruct	ion:			If No Jump: Q1	Q2	Q3	Q4			
PORTC	= 0110 0	101 [0x65]		Decode	Read literal	Process	No			
				Decode	'n'	Data	operation			
				Example:	HERE	BOV Jump)			
				Before Instru	uction					

PC

After Instruction

If Overflow = PC = If Overflow = PC =

=

1;

address (HERE)

address (Jump) 0; address (HERE+2)

© 2003-2013 Microchip	Technology Inc.
-----------------------	-----------------

COMF	Complement f
Syntax:	[<i>label</i>] COMF f [,d [,a]
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	$(\overline{f}) \rightarrow dest$
Status Affected:	N, Z
Encoding:	0001 11da ffff ffff
Description:	The contents of register 'f' are com- plemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).
Words:	1
Cycles:	1
Q Cycle Activity:	
Q1	Q2 Q3 Q4
Decode	ReadProcessWrite toregister 'f'Datadestination
Example:	COMF REG, 0, 0
Before Instru REG	uction = 0x13
After Instruct REG W	tion = 0x13 = 0xEC

CPF	SEQ	Compare	f with W,	skip if	f = W					
Synt	tax:	[label]	CPFSEQ	f [,a]						
Ope	rands:	0 ≤ f ≤ 25 a ∈ [0,1]	$\begin{array}{l} 0 \leq f \leq 255 \\ a \in [0,1] \end{array}$							
Ope	ration:									
State	us Affected:	None	None							
Enco	oding:	0110	001a f	fff	ffff					
Description: Compares the contents of data memory location 'f' to the com of W by performing an unsign subtraction. If 'f' = W, then the fetched instruction is discarded and a is executed instead, making t two-cycle instruction. If 'a' is 'd Access Bank will be selected overriding the BSR value. If 'a then the bank will be selected										
14/00	de .	•	SR value (c	ierauit).					
Wor		1								
Cycl		by	cycles if ski / a 2-word i	-						
QC	Cycle Activity: Q1	Q2	Q3		Q4					
	Decode	Read	Process		No					
	Debbac	register 'f'	Data	ор	eration					
lf sl	kip:									
	Q1	Q2	Q3		Q4					
	No	No	No		No					
14 - 1	operation	operation	operation		eration					
IT SH	kip and follow	-		n:	_					
	Q1 No	Q2 No	Q3 No		Q4 No					
	operation	operation	operation	ор	eration					
	No	No	No		No					
	operation	operation	operation	ор	eration					
<u>Exa</u>	<u>mple</u> :	HERE NEQUAL EQUAL	CPFSEQ R : :	EG, O						
	Before Instru									
	PC Addre W		ERE							
	REG	= ? = ?								
	After Instruct	ion								
	If REG	= W								
	PC		ddress (EQU	JAL)						
	If REG PC	≠ W = Ao	; ddress (NEQ)TTAT.)						
	10	- 70	~~; • • • • (14 L) \$	(تند دی ع						

MO\	/LW	Move lite	eral to W			
Synt	ax:	[label]	MOVLW	/ k		
Ope	rands:	$0 \le k \le 2$	55			
Ope	ration:	$k \to W$				
Statu	us Affected:	None				
Enco	oding:	0000	1110	kkk	k	kkkk
Des	cription:	The eigh W.	t-bit litera	l 'k' is	s loa	ded into
Wor	ds:	1				
Cycl	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3	6		Q4
	Decode	Read literal 'k'	Proce Data		Wr	ite to W
Exar	mple:	MOVLW	0x5A			

MOV	/WF	Move W t	Move W to f				
Synt	ax:	[label]	MOVWF	= f	[,a]		
Ope	rands:	0 ≤ f ≤ 25 a ∈ [0,1]	5				
Ope	ration:	$(W)\tof$					
Statu	is Affected:	None					
Enco	oding:	0110	111a	fff	f ffff		
Word	ds:	256-byte Access B	bank. If ' ank will I the BSI bank will	a' is ' be se R valu be se	lected, ue. If 'a' = 1, elected as		
Cycl	es:	1					
QC	ycle Activity:						
	Q1	Q2	2 Q3		Q4		
	Decode	Read register 'f'	Process Data		Write register 'f'		
Example: MOVWF REG, 0 Before Instruction							

MOVLW 0x5A Example:

After Instruction

W = 0x5A

Before Instruction

W	=	0x4F
REG	=	0xFF

After Instruction

 $\begin{array}{rcl} W & = & 0x4F \\ REG & = & 0x4F \end{array}$

Oper Statu	ax: rands: ration: us Affected:	(PC) + 2 -	≤ 1023 → TOS,	n						
Oper Statu	ration:	(PC) + 2 -	→ TOS,							
Statu				$-1024 \le n \le 1023$						
0.0.0	is Affected:		$\begin{array}{l} (PC) + 2 \rightarrow TOS, \\ (PC) + 2 + 2n \rightarrow PC \end{array}$							
Enco		None	None							
	oding:	1101	1nnn	nnnı	n nnnn					
		return add onto the s compleme Since the to fetch the new addree instruction	Subroutine call with a jump up to 1K from the current location. First, return address (PC+2) is pushed onto the stack. Then, add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2+2n. This instruction is a two-cycle instruction.							
Word	ds:	1	1							
Cycle	es:	2								
QC	ycle Activity:									
	Q1	Q2	Q3		Q4					
	Decode	Read literal 'n' Push PC to	Proce Data		Write to PC					
		stack								
	No operation	No operation	No operat		No operation					

Example:	HERE	RCALL Jump
		oump

Before Instruction PC = Address (HERE)

After Instruction

PC = Address (Jump) TOS = Address (HERE+2)

RES	ET	Reset					
Synt	ax:	[label]	RESET				
Ope	rands:	None					
Ope	ration:		Reset all registers and flags that are affected by a MCLR Reset.				
State	us Affected:	All					
Enco	oding:	0000	0000	1111			
Des	cription:		This instruction provides a way to execute a MCLR Reset in software.				
Wor	ds:	1	1				
Cycl	es:	1					
QC	cycle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Start	No		No		
		Reset	operatio	on op	peration		

Example: RESET

After Instruction	
Registers =	Reset Value
Flags* =	Reset Value

RLN	CF	Rotate Lo	Rotate Left f (no carry)				
Synt	ax:	[label]	RLNCF	f [,d [,a	a]		
Ope	rands:	0 ≤ f ≤ 25 d ∈ [0,1] a ∈ [0,1]	5				
Ope	ration:	$(f) \rightarrow$ $(f<7>) \rightarrow$		1>,			
Statu	us Affected:	N, Z					
Enco	oding:	0100	01da	ffff	ffff		
	cription:	rotated or the result the result 'f' (defaul Bank will the BSR bank will	The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).				
		-	regis	ster f	┫		
Wor	ds:	1					
Cycl	es:	1					
QC	cycle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Read register 'f'	Process Data		rite to ination		
<u>Exar</u>	<u>nple</u> :	RLNCF	REG,	1, 0			
	Before Instru REG	iction = 1010 1	011				
	After Instruct REG	tion = 0101 0	111				

RRCF Rotate Right f through Carry							
Syntax:	[label]	RRCF f[,d	l [,a]				
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	5					
Operation:	$(f < n >) \rightarrow dest < n-1>,$ $(f < 0 >) \rightarrow C,$ $(C) \rightarrow dest < 7>$						
Status Affected:	C, N, Z						
Encoding:	0011 00da ffff ffff						
	is placed i is placed b (default). I Bank will b the BSR v bank will b BSR value	n W. If 'd' is back in regis f 'a' is '0', th be selected, alue. If 'a' is be selected a	e Access overriding '1', then the as per the				
Words:	1						
Cycles:	1	1					
Q Cycle Activity:							
Q1	Q2	Q3	Q4				
Decode	Read register 'f'	Process Data	Write to destination				
Example:	RRCF	REG, 0,	0				
Refore Instru	uction						

Before Instruction					
REG	=	1110	0110		
С	=	0			
After Instruc	tion				
REG	=	1110	0110		
W	=	0111	0011		
С	=	0			

SLE	EP	Enter SL	Enter SLEEP mode					
Synt	ax:	[label]	[label] SLEEP					
Ope	rands:	None						
Ope	ration:							
Statu	us Affected:	TO, PD						
Enco	oding:	0000	0000	0000	0011			
Des	cription:	cleared. (TO) is se its postso The proce	The Power-down status bit (\overline{PD}) is cleared. The Time-out status bit (\overline{TO}) is set. Watchdog Timer and its postscaler are cleared. The processor is put into Sleep mode with the oscillator stopped.					
Wor	ds:	1	1					
Cycl	es:	1	1					
QC	ycle Activity:							
	Q1	Q2	Q3		Q4			
	Decode	No operation	Proce Data		Go to Sleep			
<u>Exar</u>	<u>mple</u> :	SLEEP						
Before Instruction $\overline{TO} = ?$ $\overline{PD} = ?$								
$PD = ?$ After Instruction $\frac{TO}{PD} = 1 \dagger$ $PD = 0$								

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

SUBFWB	Subtract f from W with borrow					
Syntax:	[label]	SUBFWB	f [,d	[,a]	
Operands:	c) ≤ f ≤ 25 I ∈ [0,1] a ∈ [0,1]	5			
Operation:	(W) – (f) -	$-(\overline{C}) \rightarrow de$	st		
Status Affected:	٢	N, OV, C,	DC, Z			
Encoding:	Γ	0101	01da f	fff	ffff	
Description:	() s s li s	Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).				
Words:	1					
Cycles:	1					
Q Cycle Activity:						
Q1		Q2	Q3		Q4	
Decode		Read gister 'f'	Process Data		Write to estination	
Example 1: SUBFWB REG, 1, 0						
Before Instru REG W C	= = =	n 3 2 1				
After Instruct REG	ion =	FF				
W C	=	2 0				
Z	=	0	sult is negati			
Example 2:	=	UBFWB	REG, 0,			
Before Instru			KEG, 0,	0		
REG W C	= = =	2 5 1				
After Instruct REG	ion	2				
W C Z N	= = =	2 3 1 0 0 : re:	sult is positiv	10		
Example 3:	-	UBFWB	REG, 1,			
Before Instru				-		
REG W C	= = =	1 2 0				
After Instruct REG W	ion = =	0 2				
C Z N	= = =	1 1 ; re: 0	sult is zero			

26.1 DC Characteristics: Supply Voltage PIC18F6520/8520/6620/8620/6720/8720 (Industrial, Extended) PIC18LF6520/8520/6620/8620/6720/8720 (Industrial)

PIC18LF6520/8520/6620/8620/6720/8720 (Industrial)		Standard Operating Conditions (unless otherwise stated)Operating temperature $-40^{\circ}C \le T_A \le +85^{\circ}C$ for industrial					
	6520/8520 ustrial, Ext	/6620/8620/6720/8720 ended)					
Param No.	Symbol	Characteristic	Min Typ Max Units Conditions				
D001 VDD Supply Voltage							
		PIC18LFXX20	2.0	_	5.5	V	HS, XT, RC and LP Oscillator mode
		PIC18FXX20	4.2	_	5.5	V	
D001A	AVdd	Analog Supply Voltage	Vdd - 0.3	_	VDD + 0.3	V	
D002	Vdr	RAM Data Retention Voltage ⁽¹⁾	1.5	—	-	V	
D003	VPOR	VDD Start Voltage to ensure internal Power-on Reset signal	_	—	0.7	V	See section on Power-on Reset for details
D004	Svdd	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	_	V/ms	See section on Power-on Reset for details
D005	VBOR	Brown-out Reset Voltage					
		BORV1:BORV0 = 11	N/A		N/A	V	Reserved
		BORV1:BORV0 = 10	2.64	_	2.92	V	
		BORV1:BORV0 = 01	4.11	_	4.55	V	
		BORV1:BORV0 = 00	4.41		4.87	V	

Legend: Shading of rows is to assist in readability of the table.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

RH1/A1719
RH2/A1819
RH3/A19
RH4/AN12
RH5/AN1319 RH6/AN14
RH0/AN14
RJ0/ALE
RJ1/OE
RJ2/WRL
RJ3/WRH
RJ4/BA020
RJ5/ <u>CE</u> 20
RJ6/ <u>LB</u> 20
RJ7/UB
VDD
Vss
PIR Registers
PLL Clock Timing Specifications
Pointer, FSR
POP
POR. See Power-on Reset.
PORTA
Associated Registers105
Functions105
LATA Register 103
PORTA Register
TRISA Register
PORTB Associated Registers108
Functions
LATB Register
PORTB Register
RB0/INT Pin, External
TRISB Register
PORTC
Associated Registers110
Functions110
LATC Register109
PORTC Register 109
RC3/SCK/SCL Pin171
TRISC Register
PORTD
Associated Registers
Functions113 LATD Register111
Parallel Slave Port (PSP) Function
PORTD Register
TRISD Register
PORTE
Analog Port Pins128
Associated Registers116
Functions116
LATE Register114
PORTE Register114
PSP Mode Select (PSPMODE Bit)111, 128
RE0/RD/AN5 Pin
RE1/WR/AN6 Pin128
RE1/WR/AN6 Pin128 RE2/CS/AN7 Pin128
RE1/WR/AN6 Pin
RE1/WR/AN6 Pin 128 RE2/CS/AN7 Pin 128 TRISE Register 114 PORTF 114
RE1/WR/AN6 Pin 128 RE2/CS/AN7 Pin 128 TRISE Register 114 PORTF Associated Registers 119
RE1/WR/AN6 Pin 128 RE2/CS/AN7 Pin 128 TRISE Register 114 PORTF 114

PORTF Register 117
TRISF Register 117
PORTG
Associated Registers 121
Functions 121
LATG Register 120
PORTG Register 120
TRISG Register 120, 197
PORTH
Associated Registers 124
Functions124
LATH Register 122
PORTH Register 122
TRISH Register 122
PORTJ
Associated Registers 127
Functions 127
LATJ Register 125
PORTJ Register 125
TRISJ Register 125
Postscaler, WDT
Assignment (PSA Bit)
Rate Select (T0PS2:T0PS0 Bits) 133
Switching Between Timer0 and WDT 133
Power-down Mode. See Sleep.
Power-on Reset (POR)
Oscillator Start-up Timer (OST)
Power-up Timer (PWRT)
Time-out Sequence
Prescaler, Capture
Prescaler, Timer0
Assignment (PSA Bit)
Rate Select (TOPS2:TOPS0 Bits)
Switching Between Timer0 and WDT
Prescaler, Timer2
Product Identification System
Program Counter PCL, PCLATH and PCLATU Registers
Program Memory
Instructions
Interrupt Vector
Map and Stack for PIC18FXX20
Map and Stack for FIC101 XX20 40 Maps for PIC18F8X20 Program Memory Modes 41
PIC18F8X20 Modes
Reset Vector
Program Memory Write Timing Requirements
Program Verification and Code Protection
Associated Registers
Configuration Register Protection
Data EEPROM Code Protection
Memory Code Protection
Programming, Device Instructions
PSP.See Parallel Slave Port.
Pulse Width Modulation. See PWM (CCP Module).
PUSH
PWM (CCP Module)
Associated Registers 155
5
CCPR1H:CCPR1L Registers 154
CCPR1H:CCPR1L Registers 154 Duty Cycle
Duty Cycle
Duty Cycle
Duty Cycle

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949=

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2003-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620769423

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEEL0Q® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and mulfacture of development systems is ISO 9001:2000 certified.