



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f690-e-ml

Email: info@E-XFL.COM

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

2.2 Data Memory Organization

The data memory (see Figures 2-6 through 2-8) is partitioned into four banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. The General Purpose Registers, implemented as static RAM, are located in the last 96 locations of each Bank. Register locations F0h-FFh in Bank 1, 170h-17Fh in Bank 2 and 1F0h-1FFh in Bank 3 point to addresses 70h-7Fh in Bank 0. The actual number of General Purpose Resisters (GPR) in each Bank depends on the device. Details are shown in Figures 2-4 through 2-8. All other RAM is unimplemented and returns '0' when read. RP<1:0> of the STATUS register are the bank select bits:

<u>RP1</u> <u>RP0</u>

0	0	\rightarrow	Bank 0 is selected
0	1	\rightarrow	Bank 1 is selected
1	0	\rightarrow	Bank 2 is selected

1 1 \rightarrow Bank 3 is selected

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 in the PIC16F687 and 256 x 8 in the PIC16F685/PIC16F689/ PIC16F690. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Tables 2-1 through 2-4). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Registers related to the operation of peripheral features are described in the section of that peripheral feature.

2.2.2.6 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-6.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-6: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
_	ADIF ⁽⁵⁾	RCIF ⁽³⁾	TXIF ⁽³⁾	SSPIF ⁽⁴⁾	CCP1IF ⁽²⁾	TMR2IF ⁽¹⁾	TMR1IF
bit 7							bit 0

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

bit 7		Unimplemented: Read as '0'
bit 6		ADIF: A/D Converter Interrupt Flag bit ⁽⁵⁾
		 1 = A/D conversion complete (must be cleared in software) 0 = A/D conversion has not completed or has not been started
bit 5		RCIF: EUSART Receive Interrupt Flag bit ⁽³⁾
		1 = The EUSART receive buffer is full (cleared by reading RCREG)0 = The EUSART receive buffer is not full
bit 4		TXIF: EUSART Transmit Interrupt Flag bit ⁽³⁾
		 1 = The EUSART transmit buffer is empty (cleared by writing to TXREG) 0 = The EUSART transmit buffer is full
bit 3		SSPIF: Synchronous Serial Port (SSP) Interrupt Flag bit ⁽⁴⁾
		1 = The Transmission/Reception is complete (must be cleared in software)0 = Waiting to Transmit/Receive
bit 2		CCP1IF: CCP1 Interrupt Flag bit ⁽²⁾
		Capture mode:
		 1 = A TMR1 register capture occurred (must be cleared in software) 0 = No TMR1 register capture occurred
		<u>Compare mode:</u>
		1 = A T M R T register compare match occurred (must be cleared in software) $0 = No T M R 1$ register compare match occurred
		<u>PWM mode:</u>
		Unused in this mode
bit 1		TMR2IF: Timer2 to PR2 Interrupt Flag bit ⁽¹⁾
		 1 = A Timer2 to PR2 match occurred (must be cleared in software) 0 = No Timer2 to PR2 match occurred
bit 0		TMR1IF: Timer1 Overflow Interrupt Flag bit
		1 = The TMR1 register overflowed (must be cleared in software)0 = The TMR1 register did not overflow
Note	1:	PIC16F685/PIC16F690 only.
	2:	PIC16F685/PIC16F689/PIC16F690 only.
	3:	PIC16F687/PIC16F689/PIC16F690 only.
	4:	PIC16F677/PIC16F687/PIC16F689/PIC16F690 only.
	5:	PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690 only.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7			•		•	•	bit 0
Legend:							
R = Readable bit W = Writable bit		bit	U = Unimplemented bit, read as '0'				
-n = Value at POR '1' = Bit is set			0' = Bit is cleared x = Bit is up			nown	

REGISTER 4-3: ANSEL: ANALOG SELECT REGISTER

bit 7-0 ANS<7:0>: Analog Select bits Analog select between analog or digital function on pins AN<7:0>, respectively. 1 = Analog input. Pin is assigned as analog input⁽¹⁾. 0 = Digital I/O. Pin is assigned to port or special function.

REGISTER 4-4: ANSELH: ANALOG SELECT HIGH REGISTER⁽²⁾

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANS11	ANS10	ANS9	ANS8
bit 7							bit 0

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

bit 3-0 ANS<11:8>: Analog Select bits

Analog select between analog or digital function on pins AN<7:0>, respectively.

1 =Analog input. Pin is assigned as analog input⁽¹⁾.

0 = Digital I/O. Pin is assigned to port or special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

2: PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690 only.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

6.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or $$\overline{\text{T1G}}$ pin$
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function (PIC16F685/PIC16F690 only)
- Special Event Trigger (with ECCP) (PIC16F685/PIC16F690 only)
- Comparator output synchronization to Timer1
 clock

Figure 6-1 is a block diagram of the Timer1 module.

6.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

6.2 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is FOSC/4. When TMR1CS = 1, the clock source is supplied externally.

Clock Source	T1OSCEN	FOSC Mode	TMR1CS
Fosc/4	х	xxx	0
T1CKI pin	0	xxx	1
T1LPOSC	1	LP or INTOSCIO	1

8.10.5 FIXED VOLTAGE REFERENCE

The Fixed Voltage Reference is independent of VDD, with a nominal output voltage of 0.6V. This reference can be enabled by setting the VP6EN bit of the VRCON register to '1'. This reference is always enabled when the HFINTOSC oscillator is active.

8.10.6 FIXED VOLTAGE REFERENCE STABILIZATION PERIOD

When the Fixed Voltage Reference module is enabled, it will require some time for the reference and its amplifier circuits to stabilize. The user program must include a small delay routine to allow the module to settle. See the electrical specifications section for the minimum delay requirement.

8.10.7 VOLTAGE REFERENCE SELECTION

Multiplexers on the output of the Voltage Reference module enable selection of either the CVREF or Fixed Voltage Reference for use by the comparators.

Setting the C1VREN bit of the VRCON register enables current to flow in the CVREF voltage divider and selects the CVREF voltage for use by C1. Clearing the C1VREN bit selects the fixed voltage for use by C1.

Setting the C2VREN bit of the VRCON register enables current to flow in the CVREF voltage divider and selects the CVREF voltage for use by C2. Clearing the C2VREN bit selects the fixed voltage for use by C2.

When both the C1VREN and C2VREN bits are cleared, current flow in the CVREF voltage divider is disabled minimizing the power drain of the voltage reference peripheral.



FIGURE 8-8: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

9.2 ADC Operation

9.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/ DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/DONE bit should not be set in the
	same instruction that turns on the ADC.
	Refer to Section 9.2.6 "A/D Conversion
	Procedure".

9.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

9.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 TAD delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

Note:	A device Reset forces all registers to their
	Reset state. Thus, the ADC module is
	turned off and any pending conversion is terminated.

9.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

9.2.5 SPECIAL EVENT TRIGGER

An ECCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/DONE bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Section 11.0 "Enhanced Capture/Compare/ PWM Module" for more information.

9.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

- 1. Configure Port:
 - Disable pin output driver (See TRIS register)
 - Configure pin as analog
- 2. Configure the ADC module:
 - Select ADC conversion clock
 - Configure voltage reference
 - Select ADC input channel
 - Select result format
 - Turn on ADC module
- 3. Configure ADC interrupt (optional):
 - Clear ADC interrupt flag
 - Enable ADC interrupt
 - Enable peripheral interrupt
 - Enable global interrupt⁽¹⁾
- 4. Wait the required acquisition time⁽²⁾.
- 5. Start conversion by setting the GO/DONE bit.
- 6. Wait for ADC conversion to complete by one of the following:
 - Polling the GO/DONE bit
 - Waiting for the ADC interrupt (interrupts enabled)
- 7. Read ADC Result
- 8. Clear the ADC interrupt flag (required if interrupt is enabled).
 - **Note 1:** The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: See Section 9.3 "A/D Acquisition Requirements".

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0		
—	ADCS2	ADCS1	ADCS0		—	—	—		
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	d as '0'			
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkn	nown		
bit 7	Unimplemen	ted: Read as '	0'						
bit 6-4	ADCS<2:0>:	A/D Conversio	on Clock Selec	t bits					
	000 = Fosc/2	2							
	001 = Fosc/8	3							
	010 = Fosc/3	32							
	x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)								
	100 = Fosc/4								
	101 = Fosc/16								
	110 = Fosc/64								
bit 3-0	Unimplemented: Read as '0'								

REGISTER 9-2: ADCON1: A/D CONTROL REGISTER 1

9.3 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 9-4. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD), see Figure 9-4. **The maximum recommended impedance for analog sources is 10 k** Ω . As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 9-1: ACQUISITION TIME EXAMPLE

Assumptions: Temperature =
$$50^{\circ}C$$
 and external impedance of $10k\Omega 5.0V VDD$
 $TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient$
 $= TAMP + TC + TCOFF$
 $= 2\mu s + TC + [(Temperature - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$

The value for TC can be approximated with the following equations:

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD}$$
$$V_{APPLIED}\left(1 - e^{\frac{-TC}{RC}}\right) = V_{CHOLD}$$

;[1] VCHOLD charged to within 1/2 lsb

;[2] VCHOLD charge response to VAPPLIED

$$V_{APPLIED}\left(1-e^{\frac{-Ic}{RC}}\right) = V_{APPLIED}\left(1-\frac{l}{(2^{n+1})-l}\right) \quad (combining [1] and [2])$$

Note: Where n = number of bits of the ADC.

Solving for TC:

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/2047)$$

= $-10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)$
= $1.37\mu s$

Therefore:

$$TACQ = 2MS + 1.37MS + [(50°C - 25°C)(0.05MS/°C)]$$

= 4.67MS

10.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit of the EECON1 register, and then set control bit RD. The data is available at the very next cycle, in the EEDAT register; therefore, it can be read in the next instruction. EEDAT will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 10-1: DATA EEPROM READ

BANKSEL	EEADR ;	
MOVF	DATA_EE_ADD	DR, W;
MOVWF	EEADR	;Data Memory
		;Address to read
BANKSEL	EECON1	;
BCF	EECON1, EEP	GD;Point to DATA memory
BSF	EECON1, RD	;EE Read
BANKSEL	EEDAT	;
MOVF	EEDAT, W	;W = EEDAT
BANKSEL	PORTA	;Bank 0

10.1.3 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the specific sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

BANKSELEEADR MOVFDATA_EE_ADDR, W; MOVWFEEADR ;Data Memory Address to write MOVFDATA_EE_DATA, W; MOVWFEEDAT ;Data Memory Value to write BANKSELEECON1 ; BCF EECON1, EEPGD; Point to DATA memory BSF EECON1, WREN; Enable writes BCF INTCON, GIE ; Disable INTs. BTFSCINTCON, GIE;SEE AN576 GOTO\$-2 MOVLW55h ; Required Sequence MOVWFEECON2 ;Write 55h MOVLWAAh ; MOVWFEECON2 ;Write AAh BSF EECON1, WR ;Set WR bit to begin write BSF INTCON, GIE ; Enable INTs. SLEEP ;Wait for interrupt to signal write complete (optional) BCF EECON1, WREN; Disable writes BANKSEL0x00 ;Bank 0

EXAMPLE 10-2: DATA EEPROM WRITE

11.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCP1 pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPR1L
- CCP1CON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCP1 pin. Since the CCP1 pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCP1 pin output driver.

Note:	Clearing	the	CCP1CON	register	will
	relinquish	CCF	1 control of t	he CCP1	pin.

Figure 11-3 shows a simplified block diagram of PWM operation.

Figure 11-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 11.3.7** "Setup for PWM Operation".

FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM



The PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle).

FIGURE 11-4: CCP PWM OUTPUT



FIGURE 12-12	: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)
RX/DT pin TX/CK pin (SCKP = 0)	bit 0 x bit 1 x bit 2 x bit 3 x bit 4 x bit 5 x bit 6 x bit 7
TX/CK pin (SCKP = 1) Write to bit SREN	
SREN bit	ʻ0'
RCIF bit (Interrupt) ——— Read	
RXREG	g diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

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
BAUDCTL	ABDOVF	RCIDL	_	SCKP	BRG16	_	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	x000 000x	x000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	x000 000x	x000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4					1111	1111
TXREG	EUSART	Transmit Da	ata Register	r					0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

13.3 Enabling SPI I/O

To enable the serial port, SSP Enable bit SSPEN of the SSPCON register must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRISB and TRISC registers) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<7> bit cleared
- SCK (Master mode) must have TRISB<6> bit cleared
- SCK (Slave mode) must have TRISB<6> bit set
- SS must have TRISC<6> bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRISB and TRISC) registers to the opposite value.

13.4 Typical Connection

Figure 13-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data Slave sends dummy data
- Master sends data Slave sends data
- Master sends dummy data Slave sends data



FIGURE 13-2: SPI MASTER/SLAVE CONNECTION







FIGURE 13-12: CLOCK SYNCHRONIZATION TIMING



Addr	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
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	TOIE	INTE	RABIE	TOIF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	_	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	_	_	_	1111	1111
93h	SSPMSK ⁽²⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111
94h	SSPSTAT	SMP ⁽³⁾	CKE ⁽³⁾	D/A	Р	S	R/W	UA	BF	0000 0000	0000 0000
8Ch	PIE1	_	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IF	TMR1IF	-000 0000	-000 0000

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

Note 1: PIC16F677/PIC16F687/PIC16F689/PIC16F690 only.

2: SSPMSK register (Register 13-3) can be accessed by reading or writing to SSPADD register with bits SSPM<3:0> = 1001. See Registers 13-2 and 13-3 for more details.

3: Maintain these bits clear.

DECFSZ	Decrement f, Skip if 0
Syntax:	[label] DECFSZ f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	(f) - 1 \rightarrow (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

Syntax:	[label] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 \rightarrow (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

Increment f, Skip if 0

INCFSZ

GOTO	Unconditional Branch
Syntax:	[<i>label</i>] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow PC<10:0>$ PCLATH<4:3> \rightarrow PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[<i>label</i>] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .OR. $k \rightarrow$ (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

INCF	Increment f
Syntax:	[<i>label</i>] INCF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	(f) + 1 \rightarrow (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

IORWF	Inclusive OR W with f			
Syntax:	[<i>label</i>] IORWF f,d			
Operands:	$0 \le f \le 127$ $d \in [0,1]$			
Operation:	(W) .OR. (f) \rightarrow (destination)			
Status Affected:	Z			
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.			

16.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

16.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

16.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a highspeed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

16.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a fullspeed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming[™] (ICSP[™]).

16.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.







FIGURE 18-9: MAXIMUM IDD vs. VDD OVER Fosc (EXTRC MODE)

BOR IPD vs. VDD OVER TEMPERATURE







FIGURE 18-16:

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- Business of Microchip Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support