



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	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f458-e-p

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

The PIC18FXX8 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.

Figure 2-7 shows a timing diagram indicating the transition from the main oscillator to the Timer1 oscillator. 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), 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 2-8.



FIGURE 2-7: TIMING DIAGRAM FOR TRANSITION FROM OSC1 TO TIMER1 OSCILLATOR

FIGURE 2-8: TIMING DIAGRAM FOR TRANSITION BETWEEN TIMER1 AND OSC1 (HS, XT, LP)



3.1 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when a VDD rise is detected. To take advantage of the POR circuitry, connect the MCLR pin directly (or through a resistor) to VDD. This eliminates external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (refer to parameter D004). For a slow rise time, see Figure 3-2.

When the device starts normal operation (exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met. Brown-out Reset may be used to meet the voltage start-up condition.

3.2 MCLR

PIC18FXX8 devices have a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive MCLR pin low.

The behavior of the ESD protection on the MCLR pin differs from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both Resets and current draws outside of device specification during the Reset event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 3-2, is suggested.

FIGURE 3-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



3: R1 = 10002 to 1 k02 Will limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

3.3 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out (parameter #33), only on power-up from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit (PWRTEN in CONFIG2L register) is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature and process variation. See DC parameter #33 for details.

3.4 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter #32). This additional delay ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HS4 modes and only on Power-on Reset or wake-up from Sleep.

3.5 PLL Lock Time-out

With the PLL enabled, the time-out sequence following a Power-on Reset is different from other oscillator modes. A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the oscillator start-up time-out (OST).

3.6 Brown-out Reset (BOR)

A configuration bit, BOREN, can disable (if clear/ programmed), or enable (if set), the Brown-out Reset circuitry. If VDD falls below parameter D005 for greater than parameter #35, the brown-out situation resets the chip. A Reset may not occur if VDD falls below parameter D005 for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. The Power-up Timer will then be invoked and will keep the chip in Reset an additional time delay (parameter #33). If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute the additional time delay.



Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽²⁾	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 ⁽²⁾	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(2)	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 ⁽²⁾	FBCh	ECCPR1H ⁽⁵⁾	F9Ch	
FFBh	PCLATU	FDBh	PLUSW2 ⁽²⁾	FBBh	ECCPR1L ⁽⁵⁾	F9Bh	
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP1CON ⁽⁵⁾	F9Ah	
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL ⁽⁵⁾	F97h	
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCPAS ⁽⁵⁾	F96h	TRISE ⁽⁵⁾
FF5h	TABLAT	FD5h	TOCON	FB5h	CVRCON ⁽⁵⁾	F95h	TRISD ⁽⁵⁾
FF4h	PRODH	FD4h	—	FB4h	CMCON ⁽⁵⁾	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	
FEFh	INDF0 ⁽²⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	
FEEh	POSTINC0 ⁽²⁾	FCEh	TMR1L	FAEh	RCREG	F8Eh	
FEDh	POSTDEC0 ⁽²⁾	FCDh	T1CON	FADh	TXREG	F8Dh	LATE ⁽⁵⁾
FECh	PREINC0 ⁽²⁾	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD ⁽⁵⁾
FEBh	PLUSW0(2)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—
FE7h	INDF1 ⁽²⁾	FC7h	SSPSTAT	FA7h	EECON2	F87h	_
FE6h	POSTINC1 ⁽²⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1 ⁽²⁾	FC5h	SSPCON2	FA5h	IPR3	F85h	_
FE4h	PREINC1 ⁽²⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE ⁽⁵⁾
FE3h	PLUSW1 ⁽²⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD ⁽⁵⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h		FA0h	PIE2	F80h	PORTA

Note 1: Unimplemented registers are read as '0'.

- 2: This is not a physical register.
- 3: Contents of register are dependent on WIN2:WIN0 bits in the CANCON register.
- **4:** CANSTAT register is repeated in these locations to simplify application firmware. Unique names are given for each instance of the CANSTAT register due to the Microchip header file requirement.
- 5: These registers are not implemented on the PIC18F248 and PIC18F258.

6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 6-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.5 "Writing to Flash Program Memory"**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.





FIGURE 9-3: OSC2/CLKO/RA6 PIN BLOCK DIAGRAM



9.4 PORTD, TRISD and LATD Registers

Note:	This	port	is	only	available	on	the
	PIC18F448 and PIC18F458.						

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register for the port is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

PORTD uses Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide, microprocessor port (Parallel Slave Port or PSP) by setting the control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL. See **Section 10.0 "Parallel Slave Port"** for additional information.

PORTD is also multiplexed with the analog comparator module and the ECCP module.

EXAMP	LE 9-4:	INITIALIZING PORTD
CLRF	PORTD	; Initialize PORTD by
		; clearing output
		; data latches
CLRF	LATD	; Alternate method
		; to clear output
		; data latches
MOVLW	07h	; comparator off
MOVWF	CMCON	
MOVLW	0CFh	; Value used to
		; initialize data
		; direction
MOVWF	TRISD	; Set RD3:RD0 as inputs
		; RD5:RD4 as outputs
		; RD7:RD6 as inputs

FIGURE 9-9: PORTD BLOCK DIAGRAM IN I/O PORT MODE



REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 WCOL: Write Collision Detect bit (Transmit mode only)

 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF even if only transmitting data to avoid setting overflow (must be cleared in software).
- 0 = No overflow
 - **Note:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 SSPEN: Synchronous Serial Port Enable bit

- 1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

- bit 4 CKP: Clock Polarity Select bit
 - 1 = Idle state for clock is a high level
 - 0 = Idle state for clock is a low level
- bit 3-0 SSPM3:SSPM0: Synchronous Serial Port Mode Select bits
 - 0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin 0100 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control enabled
 - 0011 = SPI Master mode, clock = TMR2 output/2
 - 0010 = SPI Master mode, clock = Fosc/64
 - 0001 = SPI Master mode, clock = Fosc/16
 - 0000 = SPI Master mode, clock = Fosc/4
 - **Note:** Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

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		



18.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous Master mode is selected, reception is enabled by setting either enable bit SREN (RCSTA register) or enable bit CREN (RCSTA register). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence. Steps to follow when setting up a Synchronous Master Reception:

- 1. Initialize the SPBRG register for the appropriate baud rate (Section 18.1 "USART Baud Rate Generator (BRG)").
- 2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 3. Ensure bits CREN and SREN are clear.
- 4. If interrupts are desired, set enable bit RCIE.
- 5. If 9-bit reception is desired, set bit RX9.
- 6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
- 7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if the enable bit RCIE was set.
- 8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 9. Read the 8-bit received data by reading the RCREG register.
- 10. If any error occurred, clear the error by clearing bit CREN.

TABLE 18-9:	BEGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION
IADEE IV V.	

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 000x	0000 000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000u
RCREG	USART Receive Register									0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

Note 1: These registers or register bits are not implemented on the PIC18F248 and PIC18F258 and read as '0's.

FIGURE 18-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



© 2006 Microchip Technology Inc.

REGISTER 19-19: RXBnDm: RECEIVE BUFFER n DATA FIELD BYTE m REGISTERS

| R/W-x |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RXBnDm7 | RXBnDm6 | RXBnDm5 | RXBnDm4 | RXBnDm3 | RXBnDm2 | RXBnDm1 | RXBnDm0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **RXBnDm7:RXBnDm0:** Receive Buffer n Data Field Byte m bits (where $0 \le n < 1$ and 0 < m < 7) Each receive buffer has an array of registers. For example, Receive Buffer 0 has 8 registers: RXB0D0 to RXB0D7.

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

REGISTER 19-20: RXERRCNT: RECEIVE ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
bit 7							bit 0

bit 7-0 REC7:REC0: Receive Error Counter bits

This register contains the receive error value as defined by the CAN specifications. When RXERRCNT > 127, the module will go into an error passive state. RXERRCNT does not have the ability to put the module in "Bus-Off" state.

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

19.7 Baud Rate Setting

All nodes on a given CAN bus must have the same nominal bit rate. The CAN protocol uses Non-Returnto-Zero (NRZ) coding which does not encode a clock within the data stream. Therefore, the receive clock must be recovered by the receiving nodes and synchronized to the transmitters clock.

As oscillators and transmission time may vary from node to node, the receiver must have some type of Phase Lock Loop (PLL) synchronized to data transmission edges to synchronize and maintain the receiver clock. Since the data is NRZ coded, it is necessary to include bit stuffing to ensure that an edge occurs at least every six bit times to maintain the Digital Phase Lock Loop (DPLL) synchronization.

The bit timing of the PIC18FXX8 is implemented using a DPLL that is configured to synchronize to the incoming data and provides the nominal timing for the transmitted data. The DPLL breaks each bit time into multiple segments made up of minimal periods of time called the *Time Quanta* (TQ).

Bus timing functions executed within the bit time frame, such as synchronization to the local oscillator, network transmission delay compensation and sample point positioning, are defined by the programmable bit timing logic of the DPLL.

All devices on the CAN bus must use the same bit rate. However, all devices are not required to have the same master oscillator clock frequency. For the different clock frequencies of the individual devices, the bit rate has to be adjusted by appropriately setting the baud rate prescaler and number of time quanta in each segment.

The *Nominal Bit Rate* is the number of bits transmitted per second, assuming an ideal transmitter with an ideal oscillator, in the absence of resynchronization. The nominal bit rate is defined to be a maximum of 1 Mb/s.



The Nominal Bit Time is defined as:

TBIT = 1/Nominal Bit Rate

The nominal bit time can be thought of as being divided into separate, non-overlapping time segments. These segments (Figure 19-7) include:

- Synchronization Segment (Sync_Seg)
- Propagation Time Segment (Prop_Seg)
- Phase Buffer Segment 1 (Phase_Seg1)
- Phase Buffer Segment 2 (Phase_Seg2)

The time segments (and thus, the nominal bit time) are, in turn, made up of integer units of time called time quanta or TQ (see Figure 19-7). By definition, the nominal bit time is programmable from a minimum of 8 TQ to a maximum of 25 TQ. Also, by definition, the minimum nominal bit time is 1 μ s corresponding to a maximum 1 Mb/s rate. The actual duration is given by the relationship:

Nominal Bit Time = TQ * (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2)

The time quantum is a fixed unit derived from the oscillator period. It is also defined by the programmable baud rate prescaler, with integer values from 1 to 64, in addition to a fixed divide-by-two for clock generation. Mathematically, this is

TQ (
$$\mu$$
s) = (2 * (BRP + 1))/Fosc (MHz)
or
TQ (μ s) = (2 * (BRP + 1)) * Tosc (μ s)

where FOSC is the clock frequency, TOSC is the corresponding oscillator period and BRP is an integer (0 through 63) represented by the binary values of BRGCON1<5:0>.



© 2006 Microchip Technology Inc.

19.8 Synchronization

To compensate for phase shifts between the oscillator frequencies of each of the nodes on the bus, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Sync_Seg). The circuit will then adjust the values of Phase Segment 1 and Phase Segment 2, as necessary. There are two mechanisms used for synchronization.

19.8.1 HARD SYNCHRONIZATION

Hard synchronization is only done when there is a recessive to dominant edge during a bus Idle condition, indicating the start of a message. After hard synchronization, the bit time counters are restarted with Sync_Seg. Hard synchronization forces the edge which has occurred to lie within the synchronization segment of the restarted bit time. Due to the rules of synchronization, if a hard synchronization occurs, there will not be a resynchronization within that bit time.

19.8.2 RESYNCHRONIZATION

As a result of resynchronization, Phase Segment 1 may be lengthened or Phase Segment 2 may be shortened. The amount of lengthening or shortening of the phase buffer segments has an upper bound given by the Synchronization Jump Width (SJW). The value of the SJW will be added to Phase Segment 1 (see Figure 19-8) or subtracted from Phase Segment 2 (see Figure 19-9). The SJW is programmable between 1 TQ and 4 TQ.

Clocking information will only be derived from recessive to dominant transitions. The property, that only a fixed maximum number of successive bits have the same value, ensures resynchronization to the bit stream during a frame. The phase error of an edge is given by the position of the edge relative to Sync_Seg, measured in Tq. The phase error is defined in magnitude of Tq as follows:

- e = 0 if the edge lies within Sync_Seg.
- e > 0 if the edge lies before the sample point.
- e < 0 if the edge lies after the sample point of the previous bit.

If the magnitude of the phase error is less than or equal to the programmed value of the synchronization jump width, the effect of a resynchronization is the same as that of a hard synchronization.

If the magnitude of the phase error is larger than the synchronization jump width and if the phase error is positive, then Phase Segment 1 is lengthened by an amount equal to the synchronization jump width.

If the magnitude of the phase error is larger than the resynchronization jump width and if the phase error is negative, then Phase Segment 2 is shortened by an amount equal to the synchronization jump width.

19.8.3 SYNCHRONIZATION RULES

- Only one synchronization within one bit time is allowed.
- An edge will be used for synchronization only if the value detected at the previous sample point (previously read bus value) differs from the bus value immediately after the edge.
- All other recessive to dominant edges, fulfilling rules 1 and 2, will be used for resynchronization with the exception that a node transmitting a dominant bit will not perform a resynchronization as a result of a recessive to dominant edge with a positive phase error.

FIGURE 19-8: LENGTHENING A BIT PERIOD (ADDING SJW TO PHASE SEGMENT 1)



20.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. The seven possible options for TAD are:

- 2 Tosc
- 4 Tosc
- 8 Tosc
- 16 Tosc
- 32 Tosc
- 64 Tosc
- Internal RC oscillator.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 $\mu s.$

Table 20-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

20.3 Configuring Analog Port Pins

The ADCON1, TRISA and TRISE registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS 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 TRIS bits.

- Note 1: When reading the port register, all pins configured as analog input channels 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 AN4:AN0 pins) may cause the input buffer to consume current that is out of the device's specification.

TABLE 20-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Device Frequency					
Operation	ADCS2:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz		
2 Tosc	000	100 ns ⁽²⁾	400 ns ⁽²⁾	1.6 μs	6 µs		
4 Tosc	100	200 ns ⁽²⁾	800 ns ⁽²⁾	3.2 μs	12 µs		
8 Tosc	001	400 ns ⁽²⁾	1.6 μs	6.4 μs	24 μs ⁽³⁾		
16 Tosc	101	800 ns ⁽²⁾	3.2 μs	12.8 μs	48 μs ⁽³⁾		
32 Tosc	010	1.6 μs	6.4 μs	25.6 μs ⁽³⁾	96 μs ⁽³⁾		
64 Tosc	110	3.2 μs	12.8 μs	51.2 μs ⁽³⁾	192 μs ⁽³⁾		
RC	011	2-6 μs ⁽¹⁾	2-6 μs ⁽¹⁾	2-6 μs ⁽¹⁾	2-6 μs ⁽¹⁾		

Legend: Shaded cells are outside of recommended range.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

TABLE 20-2: TAD vs. DEVICE OPERATING FREQUENCIES (FOR EXTENDED, LF DEVICES)

AD Clock Source (TAD)		Device Frequency					
Operation	Operation ADCS2:ADCS0		2 MHz	1.25 MHz	333.33 kHz		
2 Tosc	000	500 ns ⁽²⁾	1.0 μs ⁽²⁾	1.6 μs ⁽²⁾	6 µs		
4 Tosc	100	1.0 μs ⁽²⁾	2.0 μs ⁽²⁾	3.2 μs ⁽²⁾	12 µs		
8 Tosc	001	2.0 μs ⁽²⁾	4.0 μs	6.4 μs	24 μs ⁽³⁾		
16 Tosc	101	4.0 μs ⁽²⁾	8.0 μs	12.8 μs	48 μs ⁽³⁾		
32 Tosc	010	8.0 μs	16.0 μs	25.6 μs ⁽³⁾	96 μs ⁽³⁾		
64 Tosc	110	16.0 μs	32.0 μs	51.2 μs ⁽³⁾	192 μs ⁽³⁾		
RC	011	3-9 μs ⁽¹⁾	3-9 μs ⁽¹⁾	3-9 μs ⁽¹⁾	3-9 μs ⁽¹⁾		

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 6 μs.

- **2:** These values violate the minimum required TAD time.
- 3: For faster conversion times, the selection of another clock source is recommended.

Note 1: The RC source has a typical TAD time of 4 μ s.

NOTES:



WAKE-UP FROM SLEEP THROUGH INTERRUPT^(1,2) **FIGURE 24-2:**

2:

GIE = 1 assumed. In this case, after wake-up, the processor jumps to the interrupt routine. If GIE = 0, execution will continue in-line. Tost = 1024 Tosc (drawing not to scale). This delay will not occur for RC and EC Oscillator modes. CLKO is not available in these oscillator modes but shown here for timing reference. 3:

4:





FIGURE 24-6: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



25.0 INSTRUCTION SET SUMMARY

The PIC18 instruction set adds many enhancements to the previous PICmicro instruction sets, while maintaining an easy migration from these PICmicro instruction sets.

Most instructions are a single program memory word (16 bits) but there are three instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- Byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

The PIC18 instruction set summary in Table 25-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 25-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The destination of the result (specified by 'd')
- 3. The accessed memory (specified by 'a')

The file register designator 'f' specifies which file register is to be used by the instruction.

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

All **bit-oriented** instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The bit in the file register (specified by 'b')
- 3. The accessed memory (specified by 'a')

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator 'f' represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'k')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the CALL or RETURN instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word, except for three double-word instructions. These three instructions were made double-word instructions so that all the required information is available in these 32 bits. In the second word, the 4 MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s. Two-word branch instructions (if true) would take 3 μ s.

Figure 25-1 shows the general formats that the instructions can have.

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

The Instruction Set Summary, shown in Table 25-2, lists the instructions recognized by the Microchip MPASM[™] Assembler.

Section 25.2 "Instruction Set" provides a description of each instruction.

25.1 Read-Modify-Write Operations

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

For example, a "CLRF PORTB" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared.

BRA	Unconditio	Unconditional Branch				
Syntax:	[label] BR	[<i>label</i>] BRA n				
Operands:	-1024 ≤ n ≤	-1024 ≤ n ≤ 1023				
Operation:	(PC) + 2 + 2	$(PC) + 2 + 2n \rightarrow PC$				
Status Affected:	None					
Encoding:	1101	0nnn	nnnn	nnnn		
Description:	Add the 2's the PC. Sin incremented instruction, PC + 2 + 2r two-cycle in	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.				
Words:	1					
Cycles:	2					
Q Cycle Activity:						
Q1	Q2	Q3		Q4		
Decode	Read literal 'n'	Process Data	s Wri	te to PC		
No	No	No		No		
operation	operation	operatio	n op	eration		
Example: HERE BRA Jump Before Instruction						
PC After Instruction	n = ad	uress (Hi	EKE)			

= address (Jump)

PC

BSF	Bit Set f							
Syntax:	[<i>label</i>] BSF f,b[,a]							
Operands:	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$							
Operation:	$1 \rightarrow \text{f}$							
Status Affected:	None							
Encoding:	1000	bbba	fff	f	ffff			
Description:	Access Bank will be selected, overriding the BSR value. If 'a' = 1, th the bank will be selected as per the BSR value.				= 1, then er the			
Words:	1							
Cycles:	1							
Q Cycle Activity:								
Q1	Q2	Q	3		Q4			
Decode	Read register 'f'	f' Data Vrite			Write gister 'f'			
Example:	BSF	FLAG_RI	EG, 7					
Before Instruction FLAG_REG = 0x0A								
After Instructio FLAG_R	on EG = 0x	k8A		FLAG_REG = 0x8A				

NEG	F	Negate f					
Synta	ax:	[label] N	IEGF 1	[,a]			
Oper	rands:	$\begin{array}{l} 0 \leq f \leq 255 \\ a \in \ [0,1] \end{array}$	0 ≤ f ≤ 255 a ∈ [0,1]				
Oper	ration:	$(\overline{f}) + 1 \rightarrow f$	f				
Statu	is Affected:	N, OV, C, E)C, Z				
Enco	oding:	0110	110a	ffff	ffff		
Desc	ription:	Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'. 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.					
Word	ds:	1					
Cycle	es:	1					
QC	ycle Activity:						
	Q1	Q2	Q	3	Q4		
	Decode	ReadProcessWriteregister 'f'Dataregister					
Example: NEGF REG, 1							
	Before Instruc REG After Instructio	tion = 0011	1010 [0	x3A]			
	REG	= 1100	0110 [(0xC6]			

NOP		No Operation						
Synta	ax:	[label] NOP						
Oper	ands:	None						
Oper	ation:	No operation						
Statu	s Affected:	None						
Enco	ding:	0000 1111	0000 xxxx	000 xxx	0 :x	0000 xxxx		
Desc	ription:	No operation.						
Word	ls:	1						
Cycle	es:	1						
QC	Q Cycle Activity:							
	Q1	Q2	Q3	3		Q4		
	Decode	No operation	No No operation		No eration			

Example:

None.