



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	64MHz
Connectivity	ECANbus, I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k80-e-pt

Email: info@E-XFL.COM

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

## 6.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Using the Access Bank for many of the core PIC18 instructions introduces a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode. Inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

#### 6.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset or the Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- Use of the Access Bank ('a' = 0)
- A file address argument that is less than or equal to 5Fh

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

## 6.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost one-half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they do not use the Access Bank (Access RAM bit = 1), or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 6-9.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 29.2.1** "Extended Instruction Syntax".

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	<b>INT0IF</b>	RBIF
EEADRH	EEPROM A	ddress Registe	er High Byte					
EEADR	EEPROM A	ddress Regist	er Low Byte					
EEDATA	EEPROM D	ata Register						
EECON2	EEPROM C	Control Registe	er 2 (not a ph	nysical regist	ter)			
EECON1	EEPGD	CFGS		FREE	WRERR	WREN	WR	RD
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	_	CCP5IP	CCP4IP	CCP3IP
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	_	CCP5IF	CCP4IF	CCP3IF
PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	_	CCP5IE	CCP4IE	CCP3IE

### TABLE 8-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

## REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	—	—	—	BCLIF	HLVDIF	TMR3IF	TMR3GIF
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	<b>OSCFIF:</b> Oscillator Fail Interrupt Flag bit 1 = Device oscillator failed, clock input has changed to INTOSC (bit must be cleared in software) 0 = Device clock is operating
bit 6-4	Unimplemented: Read as '0'
bit 3	BCLIF: Bus Collision Interrupt Flag bit
	<ul> <li>1 = A bus collision occurred (bit must be cleared in software)</li> <li>0 = No bus collision occurred</li> </ul>
bit 2	HLVDIF: High/Low-Voltage Detect Interrupt Flag bit
	<ul> <li>1 = A low-voltage condition occurred (bit must be cleared in software)</li> <li>0 = The device voltage is above the regulator's low-voltage trip point</li> </ul>
bit 1	TMR3IF: TMR3 Overflow Interrupt Flag bit
	<ul> <li>1 = TMR3 register overflowed (bit must be cleared in software)</li> <li>0 = TMR3 register did not overflow</li> </ul>
bit 0	<ul> <li>TMR3GIF: TMR3 Gate Interrupt Flag bit</li> <li>1 = Timer gate interrupt occurred (bit must be cleared in software)</li> <li>0 = No timer gate interrupt occurred</li> </ul>

## 11.3 PORTB, TRISB and LATB Registers

PORTB is an eight-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISB and LATB. All pins on PORTB are digital only.

CLRF	PORTB	; Initialize PORTB by ; clearing output ; data latches
CLRF	LATB	; Alternate method
		; to clear output ; data latches
MOVLW	OCFh	
		; initialize data ; direction
MOVWF	TRISB	; Set RB<3:0> as inputs ; RB<5:4> as outputs
		; RB<7:6> as inputs

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit, RBPU (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of the PORTB pins (RB<7:4>) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur. Any RB<7:4> pins that are configured as outputs are excluded from the interrupt-on-change comparison.

Comparisons with the input pins (of RB<7:4>) are made with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from power-managed modes. To clear the interrupt in the Interrupt Service Routine:

- 1. Perform any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction).
- 2. Wait one instruction cycle (such as executing a NOP instruction).

This ends the mismatch condition.

3. Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB will end the mismatch condition and allow flag bit, RBIF, to be cleared after a one TCY delay.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

The RB<3:2> pins are multiplexed as CTMU edge inputs. RB5 has an additional function for Timer3 and Timer1. It can be configured for Timer3 clock input or Timer1 external clock gate input.

TABLE 11-7:	PORTD FUNCTIONS										
Pin Name	Function	TRIS Setting	I/O	I/O Type	Description						
RD0/C1INA/	RD0	0	0	DIG	LATD<0> data output.						
PSP0		1	Ι	ST	PORTD<0> data input.						
	C1INA	1	Ι	ANA	Comparator 1 Input A.						
-	PSP0	x	I/O	ST	Parallel Slave Port data.						
RD1/C1INB/ RD1 <sup>(1)</sup> 0 O DIG LATD<1> data output.		LATD<1> data output.									
PSP1		1	I	ST	PORTD<1> data input.						
-	C1INB <sup>(1)</sup>	1	I	ANA	Comparator 1 Input B.						
	PSP1 <sup>(1)</sup>	x	I/O	ST	Parallel Slave Port data.						
RD2/C2INA/	RD2	0	0	DIG	LATD<2> data output.						
PSP2		1	I	ST	PORTD<2> data input.						
ľ	C2INA	1	I	ANA	Comparator 2 Input A.						
ľ	PSP2	x	I/O	ST	Parallel Slave Port data.						
RD3/C2INB/	RD3	0	0	DIG	LATD<3> data output.						
CTMUI/PSP3		1	I	ST	PORTD<3> data input.						
-	C2INB	1	I	ANA	Comparator 2 Input B.						
ľ	CTMUI	x	I	_	CTMU pulse generator charger for the C2INB comparator input.						
-	PSP3	x	I/O	ST	Parallel Slave Port data.						
RD4/ECCP1/	RD4	0	0	DIG	LATD<4> data output.						
P1A/PSP4		1	I	ST	PORTD<4> data input.						
-	ECCP1	0	0	DIG	ECCP1 compare output and ECCP1 PWM output; takes priority ove port data.						
		1	I	ST	ECCP1 capture input.						
	P1A	0	0	DIG	ECCP1 Enhanced PWM output, Channel A. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.						
-	PSP4	x	I/O	ST	Parallel Slave Port data.						
RD5/P1B/PSP5	RD5	0	0	DIG	LATD<5> data output.						
		1	I	ST	PORTD<5> data input.						
	P1B	0	0	DIG	ECCP1 Enhanced PWM output, Channel B. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.						
	PSP5	x	I/O	ST	Parallel Slave Port data.						
RD6/TX2/CK2	RD6	0	0	DIG	LATD<6> data output.						
P1C/PSP6		1	I	ST	PORTD<6> data input.						
	TX2 <sup>(1)</sup>	0	0	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.						
	CK2 <sup>(1)</sup>	0	0	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.						
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.						
	P1C	0	0	DIG	ECCP1 Enhanced PWM output, Channel C. May be configured for tri-state during Enhanced PWM.						
ľ	PSP6	x	I/O	ST	Parallel Slave Port data.						

TABLE 11-7: PORTD FUNCTIONS

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input;

x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

**Note 1:** This is the pin assignment for 40 and 44-pin devices (PIC18F4XK80).

## 16.5 Timer3 Gates

Timer3 can be configured to count freely or the count can be enabled and disabled using the Timer3 gate circuitry. This is also referred to as the Timer3 gate count enable.

The Timer3 gate can also be driven by multiple selectable sources.

## 16.5.1 TIMER3 GATE COUNT ENABLE

The Timer3 Gate Enable mode is enabled by setting the TMR3GE bit (TxGCON<7>). The polarity of the Timer3 Gate Enable mode is configured using the T3GPOL bit (T3GCON<6>).

When Timer3 Gate Enable mode is enabled, Timer3 will increment on the rising edge of the Timer3 clock source. When Timer3 Gate Enable mode is disabled, no incrementing will occur and Timer3 will hold the current count. See Figure 16-2 for timing details.

## TABLE 16-1: TIMER3 GATE ENABLE SELECTIONS

T3CLK <sup>(†)</sup>	T3GPOL (T3GCON<6>)	T3G Pin	Timer3 Operation
1	0	0	Counts
$\uparrow$	0	1	Holds Count
$\uparrow$	1	0	Holds Count
$\uparrow$	1	1	Counts

† The clock on which TMR3 is running. For more information, see T3CLK in Figure 16-1.



## FIGURE 16-2: TIMER3 GATE COUNT ENABLE MODE

## 20.2.2 TIMER1/2/3/4 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 2, 3 or 4) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer to be used with each ECCP module is selected in the CCPTMRS register (Register 20-2).

#### 20.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCP1IF, should also be cleared following any such change in operating mode.

## 20.2.4 ECCP PRESCALER

There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCP1M<3:0>). Whenever the ECCP module is turned off, or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 20-1 provides the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

## EXAMPLE 20-1: CHANGING BETWEEN CAPTURE PRESCALERS

			Turn ECCP module off
MOVLW	NEW_CAPT_PS	;	Load WREG with the
		;	new prescaler mode
		;	value and ECCP ON
MOVWF	CCP1CON	;	Load ECCP1CON with
		;	this value

## FIGURE 20-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



<b>IGURE 21-5:</b>	SPI N	IODE W	AVEFO	RM (SLA	VE MO	DE WITH	CKE =	0)			
 SS Opilonsi	( .										
80% {CKP = 0 CXE = 0}	: : : :X		, 	·	·					· ·	: : : 
- VAR 20) - ROR	· ·	: : :	( ( () () () () () () () () () () () () () (		; ; ; , ,		( ( (	; ; ;	2 2 3	, , ,	
2508 (CRE = 5 (CRE = 5)	2 3 3							· · · · · · · · · · · · · · · · · · ·			
Verlie en Sisteration			5 2 7 7	: d	* \$****** \$ \$		5 2 7 7	: :: : :	* .5 1 1	: :	, 
\$6 <b>x</b> 3		Kana a		N 88.8		X68.3	X 88.0				
SD: (S3:82 = 33)										Mygeeneerine E C	: : :
ingasi Serrecia	· · ·			. 4.		<i></i>		: . <i>B</i> e	: : :	49.	
(3367 = 6)				* *	, , ,		5 2 5		* : * :	: : ://///////////////////////////////	
- S-SPHF Britanupî Filag	; ; ; ;		s 2 2	: : :	< ; ; ;		5 2 2 2	: : ::		ilia A Stend Col	. Produce
355 <b>0</b> 355 65	2 1	* :		; ,	:		i i	; ,		1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	

## FIGURE 21-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



## 23.3 A/D Acquisition Requirements

For the A/D Converter to meet its specified accuracy, the Charge Holding (CHOLD) capacitor must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 23-5. 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).

The source impedance affects the offset voltage at the analog input (due to pin leakage current). The maximum recommended impedance for analog sources is 2.5 k $\Omega$ . After the analog input channel is selected or changed, the channel must be sampled for at least the minimum acquisition time before starting a conversion.

**Note:** When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 23-1 can be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 23-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	$\leq$	1/2 LSb
Vdd	=	$3V \rightarrow Rss = 2 \ k\Omega$
Temperature	=	85°C (system max.)

## EQUATION 23-1: ACQUISITION TIME

TACQ	=	Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient
	=	TAMP + TC + TCOFF

## EQUATION 23-2: A/D MINIMUM CHARGING TIME

 $VHOLD = (VREF - (VREF/2048)) \cdot (1 - e^{(-TC/CHOLD(RIC + RSS + RS))})$ or  $TC = -(CHOLD)(RIC + RSS + RS) \ln(1/2048)$ 

#### EQUATION 23-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

TACQ	=	TAMP + TC + TCOFF
TAMP	=	0.2 μs
TCOFF	=	(Temp – 25°C)(0.02 μs/°C) (85°C – 25°C)(0.02 μs/°C) 1.2 μs
Tempera	ature c	oefficient is only required for temperatures $> 25^{\circ}$ C. Below $25^{\circ}$ C, TCOFF = 0 ms.
ТС	=	-(CHOLD)(RIC + RSS + RS) $\ln(1/2048) \mu s$ -(25 pF) (1 k $\Omega$ + 2 k $\Omega$ + 2.5 k $\Omega$ ) ln(0.0004883) $\mu s$ 1.05 $\mu s$
TACQ	=	0.2 μs + 1.05 μs + 1.2 μs 2.45 μs

## 24.0 COMPARATOR MODULE

The analog comparator module contains two comparators that can be independently configured in a variety of ways. The inputs can be selected from the analog inputs and two internal voltage references. The digital outputs are available at the pin level and can also be read through the control register. Multiple output and interrupt event generation are also available. A generic single comparator from the module is shown in Figure 24-1.

Key features of the module includes:

- · Independent comparator control
- · Programmable input configuration
- · Output to both pin and register levels
- · Programmable output polarity
- Independent interrupt generation for each comparator with configurable interrupt-on-change

## 24.1 Registers

The CMxCON registers (CM1CON and CM2CON) select the input and output configuration for each comparator, as well as the settings for interrupt generation (see Register 24-1).

The CMSTAT register (Register 24-2) provides the output results of the comparators. The bits in this register are read-only.





### EXAMPLE 27-3: TRANSMITTING A CAN MESSAGE USING BANKED METHOD

```
; Need to transmit Standard Identifier message 123h using TXBO buffer.
; To successfully transmit, CAN module must be either in Normal or Loopback mode.
; TXBO buffer is not in access bank. And since we want banked method, we need to make sure
; that correct bank is selected.
BANKSEL TXB0CON
                                 ; One BANKSEL in beginning will make sure that we are
                                 ; in correct bank for rest of the buffer access.
; Now load transmit data into TXB0 buffer.
MOVLW MY_DATA_BYTE1
                                 ; Load first data byte into buffer
MOVWF TXB0D0
                                 ; Compiler will automatically set "BANKED" bit
; Load rest of data bytes - up to 8 bytes into TXBO buffer.
. . .
; Load message identifier
MOVLW 60H
                                 ; Load SID2:SID0, EXIDE = 0
MOVWF TXB0SIDL
MOVLW 24H
                                 ; Load SID10:SID3
MOVWF TXB0SIDH
; No need to load TXB0EIDL:TXB0EIDH, as we are transmitting Standard Identifier Message only.
; Now that all data bytes are loaded, mark it for transmission.
MOVLW B'00001000'
                                 ; Normal priority; Request transmission
MOVWF TXB0CON
; If required, wait for message to get transmitted
BTFSC TXB0CON, TXREQ
                                 ; Is it transmitted?
BRA
       $-2
                                 ; No. Continue to wait...
; Message is transmitted.
```

## 27.4.3 MODE 2 – ENHANCED FIFO MODE

In Mode 2, two or more receive buffers are used to form the receive FIFO (first in, first out) buffer. There is no one-to-one relationship between the receive buffer and acceptance filter registers. Any filter that is enabled and linked to any FIFO receive buffer can generate acceptance and cause FIFO to be updated.

FIFO length is user-programmable, from 2-8 buffers deep. FIFO length is determined by the very first programmable buffer that is configured as a transmit buffer. For example, if Buffer 2 (B2) is programmed as a transmit buffer, FIFO consists of RXB0, RXB1, B0 and B1, creating a FIFO length of 4. If all programmable buffers are configured as receive buffers, FIFO will have the maximum length of 8.

The following is the list of resources available in Mode 2:

- Three transmit buffers: TXB0, TXB1 and TXB2
- Two receive buffers: RXB0 and RXB1
- Six buffers programmable as TX or RX; receive buffers form FIFO: B0-B5
- Automatic RTR handling on B0-B5
- Sixteen acceptance filters: RXF0-RXF15
- Two dedicated acceptance mask registers; RXF15 programmable as third mask: RXM0-RXM1, RXF15
- Programmable data filter on standard identifier messages: SDFLC, useful for DeviceNet protocol

## 27.5 CAN Message Buffers

## 27.5.1 DEDICATED TRANSMIT BUFFERS

The PIC18F66K80 family devices implement three dedicated transmit buffers – TXB0, TXB1 and TXB2. Each of these buffers occupies 14 bytes of SRAM and are mapped into the SFR memory map. These are the only transmit buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each transmit buffer contains one Control register (TXBnCON), four Identifier registers (TXBnSIDL, TXBnSIDH, TXBnEIDL, TXBnEIDH), one Data Length Count register (TXBnDLC) and eight Data Byte registers (TXBnDm).

## 27.5.2 DEDICATED RECEIVE BUFFERS

The PIC18F66K80 family devices implement two dedicated receive buffers: RXB0 and RXB1. Each of these buffers occupies 14 bytes of SRAM and are mapped into SFR memory map. These are the only receive buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers. Each receive buffer contains one Control register (RXBnCON), four Identifier registers (RXBnSIDL, RXBnSIDH, RXBnEIDL, RXBnEIDH), one Data Length Count register (RXBnDLC) and eight Data Byte registers (RXBnDm).

There is also a separate Message Assembly Buffer (MAB) which acts as an additional receive buffer. MAB is always committed to receiving the next message from the bus and is not directly accessible to user firmware. The MAB assembles all incoming messages one by one. A message is transferred to appropriate receive buffers only if the corresponding acceptance filter criteria is met.

## 27.5.3 PROGRAMMABLE TRANSMIT/ RECEIVE BUFFERS

The ECAN module implements six new buffers: B0-B5. These buffers are individually programmable as either transmit or receive buffers. These buffers are available only in Mode 1 and 2. As with dedicated transmit and receive buffers, each of these programmable buffers occupies 14 bytes of SRAM and are mapped into SFR memory map.

Each buffer contains one Control register (BnCON), four Identifier registers (BnSIDL, BnSIDH, BnEIDL, BnEIDH), one Data Length Count register (BnDLC) and eight Data Byte registers (BnDm). Each of these registers contains two sets of control bits. Depending on whether the buffer is configured as transmit or receive, one would use the corresponding control bit set. By default, all buffers are configured as receive buffers. Each buffer can be individually configured as a transmit or receive buffer by setting the corresponding TXENn bit in the BSEL0 register.

When configured as transmit buffers, user firmware may access transmit buffers in any order similar to accessing dedicated transmit buffers. In receive configuration with Mode 1 enabled, user firmware may also access receive buffers in any order required. But in Mode 2, all receive buffers are combined to form a single FIFO. Actual FIFO length is programmable by user firmware. Access to FIFO must be done through the FIFO Pointer bits (FP<4:0>) in the CANCON register. It must be noted that there is no hardware protection against out of order FIFO reads.

BRA		Unconditio	Unconditional Branch					
Synta	ax:	BRA n						
Oper	ands:	-1024 ≤ n ≤	1023					
Oper	ation:	(PC) + 2 +	$2n \rightarrow PC$					
Statu	is Affected:	None						
Enco	oding:	1101	0nnn	nnnn	nnnn			
Desc	ription:	Add the 2's to the PC. incremente instruction, PC + 2 + 2 two-cycle in	Since the l d to fetch the new a n. This ins	PC will I the nex iddress	nave t will be			
Word	ls:	1	1					
Cycle	es:	2	2					
QC	ycle Activity:							
	Q1	Q2	Q3		Q4			
	Decode	Read literal 'n'	Proces Data	s '	Write to PC			
	No operation	No operation	No operatio	on o	No peration			
Example: Before Instructio PC After Instruction		= ad		ump ERE )				
	PC	= ad	ldress (J	ump)				

BSF	Bit Set f			
Syntax:	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	ffff	ffff
Description:	Bit 'b' in re	gister 'f' i	s set.	
	If 'a' is '0', f If 'a' is '1', f GPR bank.	the BSR		
If 'a' is '0' and the extended instruction set is enabled, this instruction operate in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.				
	Bit-Orient	ed Instru	ctions in	Indexed
Words:	Bit-Orient	ed Instru	ctions in	Indexed
Words: Cycles:	Bit-Oriento	ed Instru	ctions in	Indexed
Cycles:	Bit-Oriente Literal Off 1	ed Instru	ctions in	Indexed
	Bit-Oriente Literal Off 1	ed Instru	ctions in e" for deta	Indexed
Cycles: Q Cycle Activity:	<b>Bit-Orient</b> Literal Off 1 1	ed Instru set Mode	ctions in e" for deta	Indexed ils.
Cycles: Q Cycle Activity: Q1	Bit-Oriente Literal Off 1 1 2 2 Read register 'f'	ed Instru set Mode Q3 Proce Data	ctions in e" for deta	Indexed ils. Q4 Write

MOVSS	Move Indexed to Indexed					
Syntax:	MOVSS [z <sub>s</sub> ], [z <sub>d</sub> ]					
Operands:	$\begin{array}{l} 0 \leq z_s \leq 127 \\ 0 \leq z_d \leq 127 \end{array}$					
Operation:	$((FSR2) + z_S) \to ((FSR2) + z_d)$					
Status Affected:	None					
Encoding: 1st word (source) 2nd word (dest.) Description						
	addresses of the source and destination registers are determined by adding the 7-bit literal offsets, ' $z_s$ ' or ' $z_d$ ', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).					
	The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.					
	If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.					
Words:	2					
Cycles:	2					
Q Cycle Activity:						

/CIE	es:	2		
ξC	ycle Activity:			
	Q1	Q2	Q3	
	Decode	Determine	Determine	I

Decode	Determine	Determine	Read
	source addr	source addr	source reg
Decode	Determine	Determine	Write
	dest addr	dest addr	to dest reg

Q4

Example:	MOVSS	[05h],	[06h]

Before Instruction		
FSR2	=	80h
Contents of 85h Contents	=	33h
of 86h	=	11h
After Instruction		
FSR2 Contents	=	80h
of 85h Contents	=	33h
of 86h	=	33h

PUSHL	Store Literal at FSR2, Decrement FSR2					
Syntax:	PUSHL k					
Operands:	$0 \leq k \leq 255$					
Operation:	$k \rightarrow (FSR2)$ FSR2 – 1 –	,,				
Status Affected:	None					
Encoding:	1111	1010	kkkł	k kkkk		
Description:	The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation. This instruction allows users to push					
	values onto a software stack.					
Words:	1					
Cycles:	1					
Q Cycle Activity:						
Q1	Q2	C	13	Q4		
Decode	Read 'k'	Proc da		Write to destination		
Example:	PUSHL 0	8h				

Before Instruction FSR2H:FSR2L Memory (01ECh)	= =	01ECh 00h
After Instruction FSR2H:FSR2L Memory (01ECh)	= =	01EBh 08h

			Standard C Operating t		re -40°C	$\leq$ TA $\leq$ -	+85°C for Industrial +125°C for Extended
Param No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
		Internal Program Memory Programming Specifications <sup>(1)</sup>					
D110	Vpp	Voltage on MCLR/VPP/RE5 pin	VDD + 1.5	—	10	V	(Note 3, Note 4)
D113	IDDP	Supply Current during Programming	—	—	10	mA	
		Data EEPROM Memory					(Note 2)
D120	ED	Byte Endurance	100K	1000K	—	E/W	-40°C to +125°C
D121	Vdrw	VDD for Read/Write	1.8	—	5.5	V	Using EECON to read/write PIC18FXXKXX devices
			1.8	—	3.6	V	Using EECON to read/write PIC18LFXXKXX devices
D122	Tdew	Erase/Write Cycle Time	—	4	_	ms	
D123	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(2)</sup>	1M	10M	—	E/W	-40°C to +125°C
		Program Flash Memory					
D130	Еρ	Cell Endurance	1K	10K	_	E/W	-40°C to +125°C
D131	Vpr	VDD for Read	1.8	—	5.5	V	PIC18FXXKXX devices
			1.8	—	3.6	V	PIC18LFXXKXX devices
D132B	Vpew	Voltage for Self-Timed Erase or Write Operations					
		VDD	1.8	—	5.5	V	PIC18FXXKXX devices
D133A	Tiw	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	—	10	mA	
D140	TWE	Writes per Erase Cycle	—	—	1		For each physical address

## TABLE 31-1: MEMORY PROGRAMMING REQUIREMENTS

only and are not tested.Note 1: These specifications are for programming the on-chip program memory through the use of table write

instructions.
2: Refer to Section 8.8 "Using the Data EEPROM" for a more detailed discussion on data EEPROM endurance.

**3:** Required only if Single-Supply Programming is disabled.

4: The MPLAB<sup>®</sup> ICD 2 does not support variable VPP output. Circuitry to limit the ICD2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD2.



## TABLE 31-23: EUSART/AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Мах	Units	Conditions
120	TCKH2DTV	<u>SYNC XMIT (MASTER and SLAVE)</u> Clock High to Data Out Valid	_	40	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	TDTRF	Data Out Rise Time and Fall Time	_	20	ns	

## FIGURE 31-20: EUSART/AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



## TABLE 31-24: EUSART/AUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (MASTER and SLAVE)				
		Data Hold before CKx $\downarrow$ (DTx hold time)	10	—	ns	
126	TCKL2DTL	Data Hold after CKx $\downarrow$ (DTx hold time)	15	_	ns	

NOTES:

## INDEX

Α

A/D	
A/D Converter Interrupt, Configuring	366
Acquisition Requirements	
ADRESH Register	
Analog Port Pins, Configuring	
Associated Registers	
Automatic Acquisition Time	
Configuring the Module	
Conversion Clock (TAD)	368
Conversion Requirements	580
Conversion Status (GO/DONE Bit)	
Conversions	
Converter Characteristics	
Differential Converter	
Operation in Power-Managed Modes	
Use of the Special Event Triggers	370
Absolute Maximum Ratings	537
AC (Timing) Characteristics	
Load Conditions for Device Timing Specifications	
Parameter Symbology	
Temperature and Voltage Specifications	
Timing Conditions	
ACKSTAT	
ACKSTAT Status Flag	322
ADCON0 Register	
GO/DONE Bit	364
ADDFSR	
ADDLW	
ADDULNK	
ADDWF	489
ADDWFC	490
ADRESL Register	364
Analog-to-Digital Converter. See A/D.	
ANDLW	49N
ANDWF	
	491
Assembler	
MPASM Assembler	
Auto-Wake-up on Sync Break Character	348
<b>D</b>	
В	
Baud Rate Generator	318
BC	
BCF	
BF	
BF Status Flag	322
Bit Timing Configuration Registers	
BRGCON1	452
BRGCON2	452
BRGCON3	
Block Diagrams	
A/D	265
Analog Input Model	
Baud Rate Generator	
CAN Buffers and Protocol Engine	392
Capture Mode Operation	269
Comparator Analog Input Model	
Comparator Configurations	
Comparator Module	
Comparator Voltage Reference	382
Comparator Voltage Reference Output Buffer	202
	303
Compare Mode Operation	270

Crystal/Ceramic Resonator Operation (HS, HSPLL.	58
CTMU	
CTMU Current Source Calibration Circuit	
CTMU Temperature Measurement Circuit	
CTMU Typical Connections and Internal Configura	
for Pulse Delay Generation	
CTMU Typical Connections and Internal Configura	
for Time Measurement	
Data Signal Modulator	
Device Clock	
Differential Channel Measurement	
EUSART Receive	
EUSART Transmit	
External Components for the SOSC Oscillator	
External Power-on Reset Circuit (Slow VDD Power-u 81	ib).
Fail-Safe Clock Monitor (FSCM)	477
Full-Bridge Application	275
Generic I/O Port Operation	
Half-Bridge Applications	
High/Low-Voltage Detect with External Input	
Interrupt Logic	
INTIO1 Oscillator Mode	60
INTIO2 Oscillator Mode	
MSSP (I <sup>2</sup> C Master Mode)	316
MSSP (I <sup>2</sup> C Mode)	296
MSSP (SPI Mode)	
On-Chip Reset Circuit	
PIC18F2XK80	
PIC18F4XK80	
PIC18F6XK80	
PIC18F6XK80 PLL	
PLL	59
PLL PORTD and PORTE (Parallel Slave Port)	59 192
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode)	59 192 271
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified)	59 192 271 262
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode	59 192 271 262 57
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode	59 192 271 262 57 57
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory	59 192 271 262 57 57 133
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering	59 192 271 262 57 57 133 284
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory	59 192 271 262 57 133 284 357
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement Single Comparator	59 192 271 262 57 133 284 357 376
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement	59 192 271 262 57 133 284 357 376 129
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement Single Comparator Table Read Operation	59 192 271 262 57 133 284 357 376 129 130
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement Single Comparator Table Read Operation Table Write Operation Table Writes to Flash Program Memory	59 192 271 262 57 133 284 357 376 129 130 135
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement Single Comparator Table Read Operation Table Write Operation	59 192 271 262 57 133 284 357 376 129 130 135 206
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode	59 192 271 262 57 133 284 357 376 129 130 135 206 206
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer0 in 8-Bit Mode	59 192 271 262 57 57 133 284 357 376 129 130 135 206 206 213
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1	59 192 271 262 57 133 284 357 376 129 130 135 206 213 222
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write Sto Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 222 226 234
PLL PORTD and PORTE (Parallel Slave Port) PWM (Enhanced Mode) PWM Operation (Simplified) RC Oscillator Mode RCIO Oscillator Mode Reads from Flash Program Memory Simplified Steering Single Channel Measurement Single Comparator Table Read Operation Table Read Operation Table Write Operation Table Writes to Flash Program Memory Timer0 in 16-Bit Mode Timer1 Timer2 Timer3 Timer4 Transmit Buffers	59 192 271 262 57 133 284 357 376 129 130 135 206 213 222 226 234 442
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization	59 192 271 262 57 133 284 357 376 129 130 135 206 213 222 226 234 442 77
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization	59 192 271 262 57 133 284 357 376 129 130 135 206 213 222 226 234 442 77 173
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Write Sto Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 222 226 234 442 77 173 472
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write Operation         Table Write Sto Flash Program Memory         Timer0 in 16-Bit Mode         Timer1 in 8-Bit Mode         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN	59 192 271 262 57 133 284 357 376 129 130 135 206 213 226 234 442 234 442 77 173 472
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 222 226 234 442 77 173 472 492
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN         BNN	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 222 226 234 442 77 173 472 493 493
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write Operation         Table Write Sto Flash Program Memory         Timer0 in 16-Bit Mode         Timer1 in 8-Bit Mode         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN         BNN         BNOV	59 192 271 262 57 133 284 357 376 129 130 135 206 213 226 234 442 234 442 77 173 472 493 493
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write of Flash Program Memory         Timer0 in 16-Bit Mode         Timer0 in 8-Bit Mode         Timer1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN         BNOV         BNZ	59 192 271 262 57 133 284 357 376 129 130 135 206 213 226 234 442 234 442 77 173 472 493 493
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1 inter1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN         BNOV         BNOV         BNZ         BOR. See Brown-out Reset.	59 192 271 262 57 133 284 357 376 129 130 135 206 213 2226 234 442 77 173 472 493 494 494
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Read Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 226 234 442 226 234 442 493 493 494 494
PLL         PORTD and PORTE (Parallel Slave Port)         PWM (Enhanced Mode)         PWM Operation (Simplified)         RC Oscillator Mode         RCIO Oscillator Mode         Reads from Flash Program Memory         Simplified Steering         Single Channel Measurement         Single Comparator         Table Read Operation         Table Write Operation         Table Write Operation         Table Writes to Flash Program Memory         Timer0 in 16-Bit Mode         Timer1 inter1         Timer2         Timer3         Timer4         Transmit Buffers         Ultra Low-Power Wake-up Initialization         Using Open-Drain Output         Watchdog Timer         BN         BNOV         BNOV         BNZ         BOR. See Brown-out Reset.	59 192 271 262 57 57 133 284 357 376 129 130 135 206 213 226 234 442 226 234 442 493 494 494 494

	. 174
SLRCON (Slew Rate Control) SSPCON1 (MSSP Control 1, I <sup>2</sup> C Mode)	.298
SSPCON1 (MSSP Control 1, SPI Mode)	
SSPCON2 (MSSP Control 2, I <sup>2</sup> C Master Mode)	. 299
SSPCON2 (MSSP Control 2, I <sup>2</sup> C Slave Mode)	. 300
SSPMSK (I <sup>2</sup> C Slave Address Mask)	. 300
SSPSTAT (MSSP Status, I <sup>2</sup> C Mode)	
SSPSTAT (MSSP Status, SPI Mode)	. 288
STATUS	. 122
STKPTR (Stack Pointer)	
T0CON (Timer0 Control)	. 205
T1CON (Timer1 Control)	
T1GCON (Timer1 Gate Control)	
T2CON (Timer2 Control)	
T3CON (Timer3 Control)	. 223
T3GCON (Timer3 Gate Control)	
T4CON (Timer4 Control)	. 233
TXBIE (Transmit Buffers Interrupt Enable)	
TXBnCON (Transmit Buffer n Control)	
TXBnDLC (Transmit Buffer n Data Length Code)	
TXBnDm (Transmit Buffer n Data Field Byte m)	
TXBnEIDH (Transmit Buffer n Extended Identifier,	
Byte) TXBnEIDL (Transmit Buffer n Extended Identifier,	
Byte)	
TXBnSIDH (Transmit Buffer n Standard Identifier,	
Byte)	
TXBnSIDL (Transmit Buffer n Standard Identifier,	
	LOW
Byte)	
	.401
Byte)	.401 .403
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control)	.401 .403 .334 .473
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable)	.401 .403 .334 .473 .172
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET	.401 .403 .334 .473 .172 .513
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457 .457
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457 .457 .457
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .457
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457 .457 .457 .457 .514
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .513 .457 .457 .457 .457 .457 .514 .514
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .457 .514 .514 .515
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457 .457 .457 .457 .514 .514 .515 .103
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 ,457 .457 .457 .457 .457 .514 .514 .515 .103 .104
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .457 .514 .514 .515 .103 .104 .601
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .457 .514 .515 .103 .104 .601 .515
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .514 .515 .103 .104 .601 .515 .516
Byte) TXERRCNT (Transmit Error Count) TXSTAx (Transmit Status and Control) WDTCON (Watchdog Timer Control) WPUB (Weak Pull-up PORTB Enable) RESET Resets	.401 .403 .334 .473 .172 .513 .457 .457 .457 .457 .457 .514 .515 .103 .104 .515 .516 .516

## S

SCK	
SDI	
SDO	
SEC_IDLE Mode	71
SEC_RUN Mode	66
Selective Peripheral Module Control	72
Serial Clock, SCK	
Serial Data In (SDI)	
Serial Data Out (SDO)	
Serial Peripheral Interface. See SPI Mode.	
SETF	517
Shoot-Through Current	
Slave Select (SS)	
SLEEP	518
Sleep Mode	70

Software Simulator (MPLAB SIM)	535
Special Event Trigger. See Compare (CCP Module).	
Special Event Trigger. See Compare (ECCP Mode).	
SPI Mode (MSSP)	
Associated Registers	
Bus Mode Compatibility	
Effects of a Reset	
Enabling SPI I/O	
Master Mode Master/Slave Connection	
Operation Operation in Power-Managed Modes	290
Serial Clock	
Serial Data In	
Serial Data Out	
Slave Mode	
Slave Node	
Slave Select Synchronization	
SPI Clock	
SSPBUF Register	
SSPSR Register	
Typical Connection	
<u>SS</u>	
SSPOV	
SSPOV Status Flag	
SSPSTAT Register	
R/W Bit	304
Stack Full/Underflow Resets	
SUBFSR	
SUBFWB	518
SUBLW	
SUBULNK	529
SUBWF	
SUBWFB	520
SWAPF	520
т	
-	400
Table Pointer Operations (table) Table Reads/Table Writes	
Table Reads/Table Writes	
TBLRD	
Time-out in Various Situations (table)	. 84
Time-out in Various Situations (table) Timer0	. 84 205
Time-out in Various Situations (table) Timer0 Associated Registers	. 84 205 207
Time-out in Various Situations (table) Timer0 Associated Registers Operation	. 84 205 207 206
Time-out in Various Situations (table) Timer0 Associated Registers	. 84 205 207 206 207

Table Pointer Operations (table)	. 132
Table Reads/Table Writes	. 105
TBLRD	. 521
TBLWT	. 522
Time-out in Various Situations (table)	84
Timer0	. 205
Associated Registers	. 207
Operation	. 206
Overflow Interrupt	. 207
Prescaler	. 207
Switching Assignment	. 207
Prescaler Assignment (PSA Bit)	. 207
Prescaler Select (T0PS2:T0PS0 Bits)	. 207
Reads and Writes in 16-Bit Mode	. 206
Source Edge Select (T0SE Bit)	. 206
Source Select (T0CS Bit)	. 206
Timer1	. 209
16-Bit Read/Write Mode	. 214
Associated Registers	. 220
Clock Source Selection	. 212
Gate	. 216
Interrupt	. 215
Operation	. 212
Oscillator	209
Oscillator, as Secondary Clock	56
Resetting, Using the ECCP Special Event Trigger.	. 216
SOSC Oscillator	. 214
Layout Considerations	. 215
Use as a Clock Source	. 215

## **PRODUCT IDENTIFICATION SYSTEM**

To order or obtain information, such as pricing or delivery, refer to the factory or the listed sales office.

PART NO.	x <u>/xx xxx</u>	Examples:
Device	Temperature Package Pattern Range	<ul> <li>a) PIC18F66K80-I/MR 301 = Industrial temp., QFN package, Extended VDD limits, QTP pattern #301.</li> <li>b) PIC18F66K80-I/PT = Industrial temp., TQFP</li> </ul>
Device <sup>(1,2)</sup>	PIC18F25K80, PIC18F26K80, PIC18F45K80, PIC18F46K80, PIC18F65K80, PIC18F66K80 VDD range 1.8V to 5V PIC18LF25K80, PIC18LF26K80, PIC18LF45K80, PIC18LF46K80, PIC18F65K80, PIC18F66K80 VDD range 1.8V to 3.6V	package, Extended VDD limits.
Temperature Range	I = $-40^{\circ}$ C to $+85^{\circ}$ C (Industrial) E = $-40^{\circ}$ C to $+125^{\circ}$ C (Extended)	
Package	P=PDIP Plastic Dual In-LineML=QFN Plastic Quad Flat, No Lead PackageSO=SOIC Plastic Small OutlineSP=SPDIP Skinny Plastic Dual In-LineSS=SSOP Plastic Shrink Small OutlinePT=TQFP Plastic Thin Quad Flatpack	Note 1:F=Standard Voltage RangeLF=Low Voltage Range2:T=in tape and reel, TQFPpackages only.
Pattern	a) QTP, SQTP, Code or Special Requirements (blank otherwise)	