



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>"

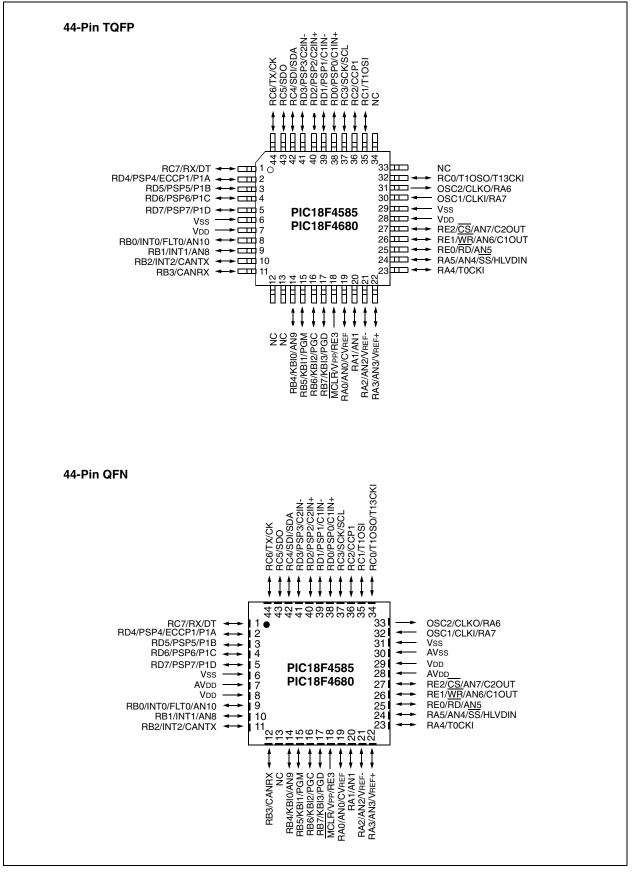
Details

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	25MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	48KB (24K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2585-e-so

Email: info@E-XFL.COM

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

Pin Diagrams (Continued)



NOTES:

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F2585
- PIC18F2680
- PIC18F4585
- PIC18F4680

This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high-endurance, Enhanced Flash program memory. In addition to these features, the PIC18F2585/2680/4585/4680 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F2585/2680/4585/4680 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- Alternate Run Modes: By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- Multiple Idle Modes: The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-fly Mode Switching:** The power managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- Lower Consumption in Key Modules: The power requirements for both Timer1 and the Watchdog Timer have been reduced by up to 80%, with typical values of 1.1 and 2.1 μA, respectively.
- Extended Instruction Set: In addition to the standard 75 instructions of the PIC18 instruction set, PIC18F2585/2680/4585/4680 devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment indirect and indexed addressing operations and the implementation of Indexed Literal Offset Addressing mode for many of the standard PIC18 instructions.

1.1.2 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F2585/2680/4585/4680 family offer ten different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes, using crystals or ceramic resonators
- Two External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O)
- Two External RC Oscillator modes with the same pin options as the External Clock modes
- An internal oscillator block which provides an 8 MHz clock (±2% accuracy) and an INTRC source (approximately 31 kHz, stable over temperature and VDD), as well as a range of 6 user selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees the two oscillator pins for use as additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the high-speed crystal and internal oscillator modes, which allows clock speeds of up to 40 MHz. Used with the internal oscillator, the PLL gives users a complete selection of clock speeds, from 31 kHz to 32 MHz – all without using an external crystal or clock circuit.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- Fail-Safe Clock Monitor: This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

TABLE 10-8:	SUMMARY OF REGISTERS ASSOCIATED WITH PORTD
-------------	--

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page		
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	52		
LATD ⁽¹⁾	LATD Data Output Register										
TRISD ⁽¹⁾	PORTD Data Direction Register										
TRISE ⁽¹⁾	IBF OBF IBOV PSPMODE — PORTE Data Direction bits								52		
ECCP1CON ⁽¹⁾	EPWM1M1	EPWM1M0	EDC1B1	EDC1B0	ECCP1M3	ECCP1M2	ECCP1M1	ECCP1M0	51		

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTD.

Note 1: These registers are available on PIC18F4X8X devices only.

12.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on CCP1 special event trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 12-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 12-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 12-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

- bit 7 RD16: 16-bit Read/Write Mode Enable bit
 - 1 = Enables register read/write of TImer1 in one 16-bit operation
 - 0 = Enables register read/write of Timer1 in two 8-bit operations
- bit 6 T1RUN: Timer1 System Clock Status bit
 - 1 = Device clock is derived from Timer1 oscillator
 - 0 = Device clock is derived from another source
- bit 5-4 T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits
 - 11 = 1:8 Prescale value
 - 10 = 1:4 Prescale value
 - 01 = 1:2 Prescale value
 - 00 = 1:1 Prescale value
- bit 3 T1OSCEN: Timer1 Oscillator Enable bit
 - 1 = Timer1 oscillator is enabled
 - 0 = Timer1 oscillator is shut off
 - The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 TISYNC: Timer1 External Clock Input Synchronization Select bit
 - When TMR1CS = 1:
 - 1 = Do not synchronize external clock input
 - 0 = Synchronize external clock input

<u>When TMR1CS = 0:</u>

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

- bit 1 TMR1CS: Timer1 Clock Source Select bit
 - 1 = External clock from pin RC0/T10SO/T13CKI (on the rising edge)
 - 0 = Internal clock (Fosc/4)
- bit 0 TMR1ON: Timer1 On bit
 - 1 = Enables Timer1
 - 0 = Stops Timer1

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

17.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before

reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

REGISTER 17-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C MODE)

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

bit 7 WCOL: Write Collision Detect bit

In Master Transmit mode:

- 1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
- 0 = No collision

In Slave Transmit mode:

- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 =No collision

In Receive mode (Master or Slave modes):

This is a "don't care" bit.

bit 6 SSPOV: Receive Overflow Indicator bit

In Receive mode:

- 1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
- 0 = No overflow

In Transmit mode:

This is a "don't care" bit in Transmit mode.

bit 5 SSPEN: Synchronous Serial Port Enable bit

- 1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins
 - Note: When enabled, the SDA and SCL pins must be properly configured as input or output.
- bit 4 CKP: SCK Release Control bit
 - In Slave mode:
 - 1 = Release clock
 - 0 = Holds clock low (clock stretch), used to ensure data setup time
 - In Master mode:

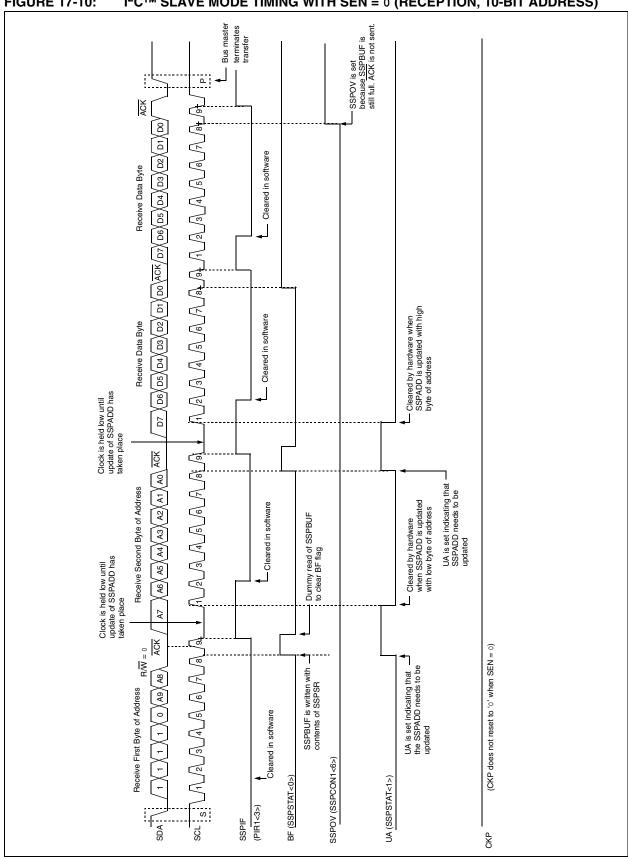
Unused in this mode.

bit 3-0 SSPM3:SSPM0: Synchronous Serial Port Mode Select bits

- $1111 = I^2C$ Slave mode, 10-bit address with Start and Stop bit interrupts enabled
- $1110 = I^2C$ Slave mode, 7-bit address with Start and Stop bit interrupts enabled
- $1011 = I^2C$ Firmware Controlled Master mode (Slave Idle)
- $1000 = I^2C$ Master mode, clock = FOSC/(4 * (SSPADD + 1))
- $0111 = I^2C$ Slave mode, 10-bit address
- $0110 = I^2C$ Slave mode, 7-bit address
 - **Note:** Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

Legend:

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



		SYNC = 0, BRGH = 0, BRG16 = 0													
BAUD RATE	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fos	Fosc = 8.000 MHz				
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)			
0.3	—	_	_	_	_				_	_	_	_			
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1201	-0.16	103			
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2403	-0.16	51			
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9615	-0.16	12			
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	_	_			
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	_	—			
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	_	_			

TABLE 18-3: BAUD RATES FOR ASYNCHRONOUS MODES

		SYNC = 0, BRGH = 0, BRG16 = 0											
BAUD RATE	Fos	c = 4.000	MHz	Fos	c = 2.000	MHz	Fos	c = 1.000	MHz				
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)				
0.3	0.300	0.16	207	300	-0.16	103	300	-0.16	51				
1.2	1.202	0.16	51	1201	-0.16	25	1201	-0.16	12				
2.4	2.404	0.16	25	2403	-0.16	12	—	_	_				
9.6	8.929	-6.99	6	—	_	—	—	_	_				
19.2	20.833	8.51	2	—	_	_	—	_	_				
57.6	62.500	8.51	0	—	_	_	—	_	_				
115.2	62.500	-45.75	0	_		—	—	_	—				

		SYNC = 0, BRGH = 1, BRG16 = 0													
BAUD RATE	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz					
(К)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)			
0.3	—	_	_	_	_	_	_	_	_		_	_			
1.2	—	—	—	—	—	—	—	—	—	—	—	—			
2.4	—	_	—	—	_	—	2.441	1.73	255	2403	-0.16	207			
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51			
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25			
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8			
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	_	—	_			

		SYNC = 0, BRGH = 1, BRG16 = 0											
BAUD RATE	Foso	c = 4.000	MHz	Fos	c = 2.000	MHz	Fos	c = 1.000	MHz				
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)				
0.3	_	_	_	_	_	_	300	-0.16	207				
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51				
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25				
9.6	9.615	0.16	25	9615	-0.16	12	_	_	_				
19.2	19.231	0.16	12	_	_	_	_	_	_				
57.6	62.500	8.51	3	_	_	_	—	_	_				
115.2	125.000	8.51	1	—	—	—	_	—	—				

		SYNC = 0, BRGH = 0, BRG16 = 1												
BAUD	Fosc = 40.000 MHz			Fosc = 20.000 MHz		Fosc = 10.000 MHz			Fosc = 8.000 MHz					
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)		
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	300	-0.04	1665		
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1201	-0.16	415		
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2403	-0.16	207		
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51		
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25		
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8		
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	_	_	—		

TABLE 18-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

		SYNC = 0, BRGH = 0, BRG16 = 1											
BAUD RATE	Fosc = 4.000 MHz			Fos	c = 2.000	MHz	Fosc = 1.000 MHz						
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)				
0.3	0.300	0.04	832	300	-0.16	415	300	-0.16	207				
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51				
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25				
9.6	9.615	0.16	25	9615	-0.16	12	_	_	—				
19.2	19.231	0.16	12	—	_	_	—	_	_				
57.6	62.500	8.51	3	—	_	_	—	_	_				
115.2	125.000	8.51	1	_	—	—		—	—				

		SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1												
BAUD RATE	Fosc = 40.000 MHz			Fosc = 20.000 MHz		Fosc = 10.000 MHz			Fosc = 8.000 MHz					
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)		
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	300	-0.01	6665		
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665		
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832		
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207		
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103		
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34		
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16		

		SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1										
BAUD RATE	Fos	c = 4.000	MHz	Fos	c = 2.000	MHz	Fosc = 1.000 MHz					
(K)	Actual Rate (K)	% Error	SPBRG Actual value Rate (decimal) (K)		% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)			
0.3	0.300	0.01	3332	300	-0.04	1665	300	-0.04	832			
1.2	1.200	0.04	832	1201	-0.16	415	1201	-0.16	207			
2.4	2.404	0.16	415	2403	-0.16	207	2403	-0.16	103			
9.6	9.615	0.16	103	9615	-0.16	51	9615	-0.16	25			
19.2	19.231	0.16	51	19230	-0.16	25	19230	-0.16	12			
57.6	58.824	2.12	16	55555	3.55	8	—	—	—			
115.2	111.111	-3.55	8	_	—	_	_	_	—			

© 2007 Microchip Technology Inc.

21.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 16-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram is of the module shown in Figure 21-1. The resistor ladder is segmented to provide two ranges of CVREF values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device VDD/VSS or an external voltage reference.

21.1 Configuring the Comparator Voltage Reference

The voltage reference module is controlled through the CVRCON register (Register 21-1). The comparator voltage reference provides two ranges of output voltage, each with 16 distinct levels. The range to be

used is selected by the CVRR bit (CVRCON<5>). The primary difference between the ranges is the size of the steps selected by the CVREF Selection bits (CVR3:CVR0), with one range offering finer resolution. The equations used to calculate the output of the comparator voltage reference are as follows:

 $\frac{\text{If CVRR} = 1:}{\text{CVREF} = ((\text{CVR3:CVR0})/24) \times \text{CVRSRC}}$ $\frac{\text{If CVRR} = 0:}{\text{CVREF} = (\text{CVDD x 1/4}) + (((\text{CVR3:CVR0})/32) \times \text{CVRSRC})}$

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF- that are multiplexed with RA2 and RA3. The voltage source is selected by the CVRSS bit (CVRCON<4>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 27-3 in Section 27.0 "Electrical Characteristics").

REGISTER 21-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	CVREN	CVROE ⁽¹⁾	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	
	bit 7							bit 0	
bit 7	CVREN: (Comparator V	oltage Refe	rence Enab	le bit				
		circuit powe							
		circuit powe							
bit 6	CVROE: (Comparator V	REF Output	Enable bit ⁽¹)				
	 1 = CVREF voltage level is also output on the RA0/AN0/CVREF pin 0 = CVREF voltage is disconnected from the RA0/AN0/CVREF pin 								
	Note 1:	CVROE over be configure		TRISA<0> b out by settin	0		output, RA2	2 must also	
bit 5	CVRR: Co	omparator VR	EF Range S	election bit					
	1 = 0.00 C	VRSRC to 0.7	5 CVRSRC,	with CVRSR	c/24 step si	ze			
	0 = 0.25 C	VRSRC to 0.7	5 CVRSRC,	with CVRSR	c/32 step si	ze			
bit 4	CVRSS: C	Comparator Vi	REF Source	Selection b	it				
	•	arator referen		•	<i>,</i> , ,	REF-)			
		arator referen	,						
bit 3-0	CVR3:CV	R0: Compara	tor VREF Va	alue Selectio	on bits ($0 \le ($	CVR3:CVR	0) ≤ 15)		
	When CVI								
		(CVR3:CVR0)/24) • (CV	RSRC)					
	When CVI								
	OVREF = (CVRSRC/4) +	((CVH3:CV	'HU)/32) ● (C	VRSRC)				
	r								
	l egend:								

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

23.2.3.1 Programmable TX/RX and Auto-RTR Buffers

The ECAN module contains 6 message buffers that can be programmed as transmit or receive buffers. Any of these buffers can also be programmed to automatically handle RTR messages.

Note: These registers are not used in Mode 0.

REGISTER 23-22: BnCON: TX/RX BUFFER n CONTROL REGISTERS IN RECEIVE MODE $[0 \le n \le 5, TXnEN (BSEL0 \le n)] = 01^{(1)}$

	$[0 \le n \le 5,$	IXNEN (B	SELU <n>)</n>	= 0](.)				
	R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXFUL ⁽²⁾	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0
	bit 7							bit 0
bit 7	RXFUL: Re	eceive Full	Status bit ⁽²⁾					
			ntains a receitopen to receiv		,			
bit 6	RXM1: Rec	eive Buffe	r Mode bit					
			iges including nessages as		•	eptance filte	ers are ignor	ed)
bit 5	RXRTRRO	: Read-On	ly Remote Tra	ansmission	Request for	Received N	lessage bit	
			e is a remote e is not a rem			st		
bit 4-0	FILHIT4:FI	LHITO: Filt	er Hit bits					
	These bits i	ndicate wh	ich acceptan	ce filter ena	bled the last	message re	eception into	this buffer.
		•	Filter 15 (RXF Filter 14 (RXF	,				
				•				
		•	Filter 1 (RXF ⁻ Filter 0 (RXF(,				
	Note 1:	These reg	isters are ava	ailable in Mo	de 1 and 2	only.		
	2:	by softwar	set by the CA e after the bu and the buffe	ffer is read.	As long as l	RXFUL is se		

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

23.7 Message Reception

23.7.1 RECEIVING A MESSAGE

Of all receive buffers, the MAB is always committed to receiving the next message from the bus. The MCU can access one buffer while the other buffer is available for message reception or holding a previously received message.

Note: The entire contents of the MAB are moved into the receive buffer once a message is accepted. This means that regardless of the type of identifier (standard or extended) and the number of data bytes received, the entire receive buffer is overwritten with the MAB contents. Therefore, the contents of all registers in the buffer must be assumed to have been modified when any message is received.

When a message is moved into either of the receive buffers, the associated RXFUL bit is set. This bit must be cleared by the MCU when it has completed processing the message in the buffer in order to allow a new message to be received into the buffer. This bit provides a positive lockout to ensure that the firmware has finished with the message before the module attempts to load a new message into the receive buffer. If the receive interrupt is enabled, an interrupt will be generated to indicate that a valid message has been received.

Once a message is loaded into any matching buffer, user firmware may determine exactly what filter caused this reception by checking the filter hit bits in the RXBnCON or BnCON registers. In Mode 0, FILHIT<3:0> of RXBnCON serve as filter hit bits. In Mode 1 and 2. FILHIT<4:0> of BnCON serves as filter hit bits. The same registers also indicate whether the current message is an RTR frame or not. A received message is considered a standard identifier message if the EXID bit in the RXBnSIDL or the BnSIDL register is cleared. Conversely, a set EXID bit indicates an extended identifier message. If the received message is a standard identifier message, user firmware needs to read the SIDL and SIDH registers. In the case of an extended identifier message, firmware should read the SIDL, SIDH, EIDL and EIDH registers. If the RXBnDLC or BnDLC register contain non-zero data count, user firmware should also read the corresponding number of data bytes by accessing the RXBnDm or the BnDm registers. When a received message is an RTR and if the current buffer is not configured for automatic RTR handling, user firmware must take appropriate action and respond manually.

Each receive buffer contains RXM bits to set special Receive modes. In Mode 0, RXM<1:0> bits in RXBnCON define a total of four Receive modes. In Mode 1 and 2, RXM1 bit, in combination with the EXID mask and filter bit, define the same four Receive modes. Normally, these bits are set to '00' to enable reception of all valid messages as determined by the appropriate acceptance filters. In this case, the determination of whether or not to receive standard or extended messages is determined by the EXIDE bit in the acceptance filter register. In Mode 0, if the RXM bits are set to '01' or '10', the receiver will accept only messages with standard or extended identifiers, respectively. If an acceptance filter has the EXIDE bit set such that it does not correspond with the RXM mode, that acceptance filter is rendered useless. In Mode 1 and 2, setting EXID in the SIDL Mask register will ensure that only standard or extended identifiers are received. These two modes of RXM bits can be used in systems where it is known that only standard or extended messages will be on the bus. If the RXM bits are set to '11' (RXM1 = 1 in Mode 1 and 2), the buffer will receive all messages regardless of the values of the acceptance filters. Also, if a message has an error before the end of frame, that portion of the message assembled in the MAB before the error frame will be loaded into the buffer. This mode may serve as a valuable debugging tool for a given CAN network. It should not be used in an actual system environment as the actual system will always have some bus errors and all nodes on the bus are expected to ignore them.

In Mode 1 and 2, when a programmable buffer is configured as a transmit buffer and one or more acceptance filters are associated with it, all incoming messages matching this acceptance filter criteria will be discarded. To avoid this scenario, user firmware must make sure that there are no acceptance filters associated with a buffer configured as a transmit buffer.

23.7.2 RECEIVE PRIORITY

When in Mode 0, RXB0 is the higher priority buffer and has two message acceptance filters associated with it. RXB1 is the lower priority buffer and has four acceptance filters associated with it. The lower number of acceptance filters makes the match on RXB0 more restrictive and implies a higher priority for that buffer. Additionally, the RXB0CON register can be configured such that if RXB0 contains a valid message and another valid message is received, an overflow error will not occur and the new message will be moved into RXB1 regardless of the acceptance criteria of RXB1. There are also two programmable acceptance filter masks available, one for each receive buffer (see Section 23.5 "CAN Message Buffers").

In Mode 1 and 2, there are a total of 16 acceptance filters available and each can be dynamically assigned to any of the receive buffers. A buffer with a lower number has higher priority. Given this, if an incoming message matches with two or more receive buffer acceptance criteria, the buffer with the lower number will be loaded with that message.

23.10 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.

23.10.1 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.

23.10.2 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 23-6) or subtracted from Phase Segment 2 (see Figure 23-7). 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.

23.10.3 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.

CPFSGT	Compare	f with W, Sk	ip if f > W						
Syntax:	CPFSGT	f {,a}							
Operands:	$0 \leq f \leq 255$	$0 \le f \le 255$							
	a ∈ [0,1]								
Operation:	(f) - (W),								
	skip if (f) > (. ,							
		comparison)							
Status Affected:	None	None							
Encoding:	0110	010a fff	f fff						
Description:	location 'f' t	he contents of o the contents an unsigned s	of the W by						
	contents of instruction i	nts of 'f' are gr WREG, then t s discarded ar stead, making astruction.	the fetched						
		he Access Bar he BSR is use (default)							
	If 'a' is '0' a set is enabl in Indexed I mode when Section 25	nd the extende ed, this instruc Literal Offset A lever f ≤ 95 (5F .2.3 "Byte-Ori	ction operates addressing Th). See ented and						
		ed Instruction set Mode" for							
Words:	1								
Cycles:	1(2) Note: 3 c	cycles if skip a	nd followed						
		a 2-word instr							
Q Cycle Activity:									
Q1	Q2	Q3	Q4						
Decode	Read	Process	No						
	register 'f'	Data	operation						
If skip:									
Q1	Q2	Q3	Q4						
No	No	No	No						
operation	operation	operation	operation						
If skip and followed	-		04						
Q1 No	Q2 No	Q3 No	Q4 No						
operation	operation	operation	operation						
No	No	No	No						
operation	operation	operation	operation						
<u>Example:</u>	HERE NGREATER GREATER	CPFSGT RE : :	G, 0						
Before Instruc	tion								
PC	= Ad	dress (HERE))						
PC W	= Ad = ?	dress (HERE))						
PC W After Instructio If REG	= Ad = ?)						
PC W After Instructio If REG PC	= Ad = ? on > W; = Ad	dress (GREA							
PC W After Instructio If REG	= Ad = ? > W; = Ad ≤ W;	dress (GREA	FER)						

Syntax:CPFSLT f {,a}Operands: $0 \le f \le 255$ $a \in [0,1]$ Operation:(f) - (W), skip if (f) < (W) (unsigned comparison)Status Affected:NoneEncoding: 0110 $000a$ ffffDescription:Compares the contents of data memory location 'f to the contents of W by performing an unsigned subtraction. If the contents of 'f are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instruction. If 'a' is '1', the BSR is used to select the GPR bank (default).Words:1Cycles:1(2) Note:Note:3 cycles if skip and followed by a 2-word instruction.Q Cycle Activity:Q1Q1Q2Q3Q4DecodeRead register 'f'DataoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationoperationNo	CPF	SLT	Compare	f with W, Sk	ip if f < W				
$a \in [0,1]$ Operation: $(f) - (W), \\skip if (f) < (W) \\(unsigned comparison)$ Status Affected: Encoding: $0110 000a ffff ffff$ Description: Compares the contents of data memory location 'f to the contents of W by performing an unsigned subtraction. If the contents of 'f are less than the contents of 'f are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). Words: $1 Cycles: 1(2) Note: 3 cycles if skip and followed by a 2-word instruction. Q Cycle Activity: Q1 Q2 Q3 Q4 Decode Read Process No \\ operation operation operation \\ operation operation \\ operation operation \\ operation$	Synta	ax:	CPFSLT f	f {,a}					
$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	Oper	rands:							
Encoding:0110000affffffffDescription:Compares the contents of data memory location 'f to the contents of W by performing an unsigned subtraction. If the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is 'o', the Access Bank is selected. If 'a' is 'o', the Access Bank is selected. Decode If 'a' is 'o', the Access Bank is selected. Decode If ac 'a' s cycles if skip and followed by a 2-word instruction.Q Cycle Activity:Q1Q2Q3Q4Q1Q2Q3Q4DecodeRead register 'f'Data DataoperationIf skip:Q1Q2Q3Q4NoNoNoNooperationIf skip and followed by 2-word instruction:Q1Q2Q3Q4NoNoNoNoNooperationoperationoperationoperationIf skip and followed by 2-word instruction:Q1Q2Q3Q4NoNoNoNoNooperationoperationoperationoperationDecodeReadPCIf Skip and followed by 2-word instruction:Q1Q2Q3Q4NoNoNoNooperationoperationoperation </td <td>Oper</td> <td>ration:</td> <td>skip if (f) <</td> <td colspan="6">skip if $(f) < (W)$</td>	Oper	ration:	skip if (f) <	skip if $(f) < (W)$					
Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is 'o', the Access Bank is selected. If 'a' is 'o', the Access Bank is select the GPR bank (default). Words: 1 Cycles: 1(2) Note: 3 cycles if skip and followed by a 2-word instruction. Q Cycle Activity: Q1 Q2 Q3 Q4 Decode Read Process No register 'f' Data operation If skip: Q1 Q2 Q3 Q4 No operation operation operation If skip and followed by 2-word instruction: Q1 Q2 Q3 Q4 No operation operation operation operation operation operation If skip and followed by 2-word instruction: Q1 Q2 Q3 Q4 No operation operation operation operation operation operation If skip and followed by 2-word instruction: Q1 Q2 Q3 Q4 No operation operation operation operation No operation operation operation operation PC = Address (HERE) W = ? After Instruction If REG W; PC Address (LESS) If REG W; HERE N; HERE N; HERE N; HERE CPFSLT REG, 1 NLESS HERE Address (LESS) Herei HERE N; HERE CPFSLT CPC Address (LESS)	Statu	is Affected:	None						
$\begin{array}{rcl} & \mbox{location 'f' to the contents of W by} \\ \mbox{performing an unsigned subtraction.} \\ & \mbox{If the contents of 'I' are less than the} \\ & \mbox{contents of W, then the fetched} \\ & \mbox{instruction is discarded and a NOP is} \\ & \mbox{executed instead, making this a} \\ & \mbox{two-cycle instruction.} \\ & \mbox{If 'a' is 'o', the Access Bank is selected.} \\ & \mbox{If 'a' is 'i', the BSR is used to select the} \\ & \mbox{GPR bank (default).} \\ \\ \hline & \mbox{Words: 1} \\ \\ \hline & \mbox{Cycle Activity: } \\ \hline & \mbox{Q1} & \mbox{Q2} & \mbox{Q3} & \mbox{Q4} \\ \hline & \mbox{Decode} & \mbox{Read} & \mbox{Process} & \mbox{No} \\ & \mbox{operation} & \mbox{operation} \\ \\ \hline & \mbox{If skip: } \\ \hline & \mbox{Q1} & \mbox{Q2} & \mbox{Q3} & \mbox{Q4} \\ \hline & \mbox{Decode} & \mbox{Read} & \mbox{Process} & \mbox{No} \\ & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \\ \hline & \mbox{If skip: } \\ \hline & \mbox{Q1} & \mbox{Q2} & \mbox{Q3} & \mbox{Q4} \\ \hline & \mbox{No} & \mbox{No} & \mbox{No} \\ & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{If skip and followed by 2-word instruction: } \\ \hline & \mbox{Q1} & \mbox{Q2} & \mbox{Q3} & \mbox{Q4} \\ \hline & \mbox{No} & \mbox{No} & \mbox{No} & \mbox{No} \\ & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & \mbox{operation} & \mbox{operation} \\ \hline & \mbox{operation} & \mbox{operation} & operat$	Enco	oding:	0110	000a fff	f ffff				
Words:1Cycles:1(2)Note:3 cycles if skip and followed by a 2-word instruction.Q Cycle Activity: $Q1$ Q1Q2Q3Q4DecodeReadProcessNo operationIf skip: $Q1$ Q1Q2Q3Q4No operationNo operationIf skip: $Q1$ Q2Q3Q4 $Q2$ Q3Q4No operationNo operationIf skip and followed by 2-word instruction:Q1Q2Q3Q4No operationNo operationNo operationNo operationNo operationNo operationPC W=Example:HERE LESSBefore Instruction W=PC W=After Instruction If REG<	Desc	pription:	location 'f' t performing If the conter contents of instruction i executed in two-cycle ir If 'a' is '0', t If 'a' is '1', t	o the contents an unsigned s nts of 'f' are les W, then the fe is discarded ar istead, making nstruction. he Access Bar he BSR is used	of W by ubtraction. ss than the tched a NOP is this a k is selected.				
Cycles:1(2)Note:3 cycles if skip and followed by a 2-word instruction.Q Cycle Activity: $Q1$ Q1Q2Q3Q4DecodeRead register 'f'DataoperationIf skip: $Q1$ Q1Q2Q3Q4NoNoNoNooperationoperationoperationoperationoperationoperationoperationoperationQ1Q2Q3Q4NoNess:LESS:Before InstructionPCW= ?After InstructionIf REGM= Address (LESS)If REG> W;	Word	ls.		(deladit).					
Note: 3 cycles if skip and followed by a 2-word instruction.Q Cycle Activity:Q1Q2Q3Q4DecodeRead register ifProcessNo operationIf skip:Q1Q2Q3Q4NoNoNoNoNo operationIf skip:Q1Q2Q3Q4NoNoNoNo operationNoIf skip and followed by 2-word instruction:Q1Q2Q3Q1Q2Q3Q4NoNoNoNo operationIf skip and followed by 2-word instruction:Q1Q2Q3Q1Q2Q3Q4NoNoNoNo operationIf skip and followed by 2-word instruction:Q1Q2Q3Q4NoNoNoNo operationIf skip and followed by 2-word instruction:Q1Q2Q3Q1Q2Q3Q4NoNoNoNo operationNoNoNoNoNooperationoperationoperationoperationoperationoperationNo </td <td></td> <td></td> <td>-</td> <td></td> <td></td>			-						
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	-		Note: 3 c	•					
$\begin{tabular}{ c c c c c c c } \hline Pc & Read & Process & No & operation \\ \hline register 'f' & Data & operation \\ \hline Operation & operation \\ \hline Data & operation \\ \hline Operation & operation & operation \\ \hline No & No & No & No \\ \hline operation & operation & operation \\ \hline No & No & No & No \\ \hline operation & operation & operation \\ \hline No & No & No & No \\ \hline operation & operation & operation \\ \hline Data & operation \\ \hline Data & operation \\ \hline Operation & operation \\ \hline Operation & operation & operation \\ \hline No & No & No & No \\ \hline operation & operation & operation \\ \hline No & No & No & No \\ \hline operation & operation & operation \\ \hline PC & = Address (HERE) \\ W & = ? \\ \hline After Instruction \\ \hline If REG & < W; \\ PC & = Address (LESS) \\ \hline If REG & \geq W; \\ \hline \end{tabular}$	QC				.				
$\begin{tabular}{ c c c c c c } \hline register `f' & Data & operation \\ \hline register `f' & Data & operation \\ \hline If skip: \\ \hline $Q1 & Q2 & Q3 & Q4 \\ \hline $No & No & No & operation & operation & operation \\ \hline $operation & operation & operation & operation \\ \hline $If skip and followed by 2-word instruction: \\ \hline $Q1 & Q2 & Q3 & Q4 \\ \hline $No & No & No & No & operation & o$				1					
If skip: Q1 Q2 Q3 Q4 No No No No No operation operation operation operation operation If skip and followed by 2-word instruction: Q1 Q2 Q3 Q4 No No No No operation operation Q1 Q2 Q3 Q4 No No No No operation operation operation operation operation operation No No No No No operation No No No No No operation No No No No No operation operation operation operation operation operation No No No No No operation operation operation operation operation operation Example: HERE CPFSLT REG, 1 NLESS LESS LESS W =		Decode							
$\begin{tabular}{ c c c c c c } \hline No & No & operation & operation & operation & operation \\ \hline operation & operation & operation & operation \\ \hline If skip and followed by 2-word instruction: \\ \hline Q1 & Q2 & Q3 & Q4 \\ \hline No & No & No & No & operation &$	lf sk	tip:							
$\begin{tabular}{ c c c c c c } \hline \end{tabular} & tabula$		Q1	Q2	Q3	Q4				
If skip and followed by 2-word instruction: Q1 Q2 Q3 Q4 No No No No operation operation operation No No No No No operation operation No operation operation operation Example: HERE CPFSLT REG, 1 NLESS : LESS : Before Instruction PC = Address (HERE) W = ? After Instruction If REG < W; PC = Address (LESS) If REG \geq W;		No	No	No	No				
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$					operation				
$\begin{tabular}{ c c c c c c } \hline No & No & No & operation & oper$	If Sk		-	-	04				
operation operation operation operation No No No No operation operation operation operation Example: HERE CPFSLT REG, 1 NLESS : LESS : Before Instruction PC PC = After Instruction If REG <									
operationoperationoperationoperationExample:HERECPFSLT REG, 1NLESS:LESS:Before InstructionPCPC=After InstructionIf REG<									
$\begin{array}{cccc} & \text{HERE} & \text{CPFSLT REG, 1} \\ & \text{NLESS} & : \\ & \text{LESS} & : \\ \end{array}$ Before Instruction $\begin{array}{c} PC & = & \text{Address (HERE)} \\ W & = & ? \\ \text{After Instruction} \\ & \text{If REG} & < & W; \\ PC & = & \text{Address (LESS)} \\ & \text{If REG} & \geq & W; \end{array}$									
$\begin{array}{rcl} \mathrm{NLESS} & : & & \\ \mathrm{LESS} & : & & \\ \end{array}$ Before Instruction $\begin{array}{rcl} \mathrm{PC} & = & \mathrm{Address} & (\mathrm{HERE}) \\ \mathrm{W} & = & ? & \\ \end{array}$ After Instruction $\begin{array}{rcl} \mathrm{If} \ \mathrm{REG} & < & \mathrm{W}; \\ \mathrm{PC} & = & \mathrm{Address} & (\mathrm{LESS}) \\ \mathrm{If} \ \mathrm{REG} & \geq & \mathrm{W}; \end{array}$		operation	operation	operation	operation				
PC = Address (HERE) W = ? After Instruction If REG < W; PC = Address (LESS) If REG ≥ W;	<u>Exar</u>	nple:	NLESS	:	1				
W = ? After Instruction If REG < W; PC = Address (LESS) If REG ≥ W;		Before Instruc	tion						
After Instruction If REG < W; PC = Address (⊥ESS) If REG ≥ W;				dress (HERE))				
PC = Address (LESS) If REG ≥ W;									
If REG \geq W;		If REG	< W;						
)				
					5)				

MO\	/SS	Move Ind	exed to	Inde	exed	l		
Synta	ax:	MOVSS [z _s], [z _d]					
Oper	ands:	0 ≤ z _s ≤ 12 0 ≤ z _d ≤ 12						
Oper	ation:	((FSR2) + :	$z_s) \rightarrow ((F$	SR2)	+ z _d))		
Statu	s Affected:	None						
Enco	ding:							
1st w	ord (source)	1110	1011	1zz	ZZ	zzzz _s		
2nd v	word (dest.)	1111	xxxx	XZZ	ZZ	zzzzd		
Desc	ription	moved to t addresses registers a 7-bit literal respectivel registers c the 4096-b (000h to FI The MOVSS PCL, TOSI destination If the result an indirect	The contents of the source register are moved to the destination register. The 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					
		instruction	will exec	ute as	a N	OP.		
Word	ls:	2						
Cycle	es:	2						
QC	ycle Activity:							
	Q1	Q2	Q	3	1	Q4		
	Decode	Determine	Determ	nine		Read		

Decode	Determine	Determine	Read
	source addr	source addr	source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg
		0031 2001	to dest leg

Example:	MOVSS	[05h],	[06h]
Before Instruction FSR2	on _	80h	
Contents	=		
of 85h Contents	=	33h	
of 86h	=	11h	
After Instruction FSR2	۱ 	80h	
Contents	=	0011	
of 85h Contents	=	33h	
of 86h	=	33h	

PUSHL Store Literal at FSR2, Decrement FSR2 Syntax: PUSHL k Operands: $0 \leq k \leq 255$ Operation: $k \rightarrow (FSR2),$ $FSR2 - 1 \rightarrow FSR2$ Status Affected: None Encoding: 1010 kkkk kkkk 1111 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 Q3 Q4 Decode Read 'k' Process Write to destination data Example: PUSHL 08h Before Instruction FSR2H:FSR2L Memory (01ECh) 01ECh = 00h = A

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

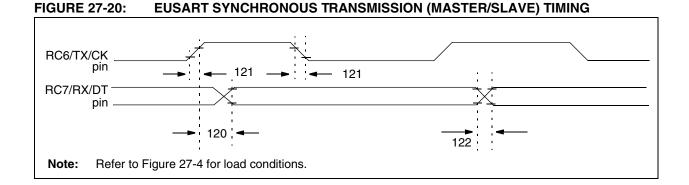


TABLE 27-22: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
120	TCKH2DTV	SYNC XMIT (MASTER & SLAVE)					
		Clock High to Data Out Valid	PIC18FXXXX	_	40	ns	
			PIC18LFXXXX		100	ns	VDD = 2.0V
121	TCKRF	Clock Out Rise Time and Fall Time	PIC18FXXXX		20	ns	
		(Master mode)	PIC18LFXXXX	Ι	50	ns	VDD = 2.0V
122	TDTRF	Data Out Rise Time and Fall Time	PIC18FXXXX		20	ns	
			PIC18LFXXXX	_	50	ns	VDD = 2.0V

FIGURE 27-21: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

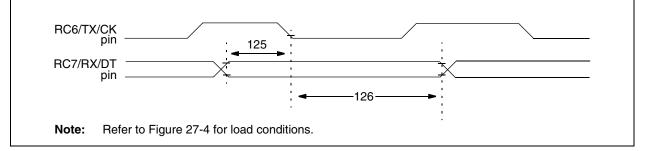


TABLE 27-23: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125		SYNC RCV (MASTER & SLAVE) Data Hold before CK \downarrow (DT hold time)	10	_	ns	
126	TCKL2DTL	Data Hold after CK \downarrow (DT hold time)	15	_	ns	

NOTES:

Data Memory	
Access Bank	69
and the Extended Instruction Set	91
Bank Select Register (BSR)	67
General Purpose Registers	
Map for PIC18F2X8X/4X8X	
Special Function Registers	70
DAW	380
DC and AC Characteristics	-
DC Characteristics	426
Power-Down and Supply Current	418
Supply Voltage	417
DCFSNZ	381
DECF	380
DECFSZ	381
Development Support	
Device Differences	461
Device Overview	
Features (table)	
New Core Features	7
Device Reset Timers	-
Oscillator Start-up Timer (OST)	
PLL Lock Time-out	
Power-up Timer (PWRT)	
Direct Addressing	
Disable Mode	324

Ε

ECAN Technology	273
Baud Rate Setting	332
Bit Time Partitioning	
Bit Timing Configuration Registers	338
Calculating TQ, Nominal Bit Rate	
and Nominal Bit Time	335
CAN Baud Rate Registers	311
CAN Control and Status Registers	275
CAN Controller Register Map	
CAN I/O Control Register	314
CAN Interrupt Registers	315
CAN Interrupts	339
Acknowledge	341
Bus Activity Wake-up	341
Bus-Off	341
Code Bits	340
Error	341
Message Error	340
Receive	340
Receiver Bus Passive	341
Receiver Overflow	341
Receiver Warning	341
Transmit	340
Transmitter Bus Passive	341
Transmitter Warning	341
CAN Message Buffers	326
Dedicated Receive	326
Dedicated Transmit	326
Programmable Auto-RTR	327
Programmable Transmit/Receive	326
CAN Message Transmission	327
Aborting	327
Initiating	327
Priority	328
CAN Modes of Operation	324
CAN Registers	275

	324
Dedicated CAN Receive Buffer Registers	
Dedicated CAN Transmit Buffer Registers	
Disable Mode	
Error Detection	
Acknowledge	
Bit	
Error Modes and Counters	
Error States	
Form	
Stuff Bit	
Error Modes State (diagram)	
Error Recognition Mode Filter-Mask Truth (table)	
Functional Modes	
Mode 0 (Legacy Mode)	
Mode 1 (Enhanced Legacy Mode)	325
Mode 2 (Enhanced FIFO Mode)	
Information Processing Time (IPT)	
Lengthening a Bit Period	
Listen Only Mode	
Loopback Mode Message Acceptance Filters and Masks 302,	
Message Acceptance Platers and Masks	330
and Filter Operation	004
Message Reception Enhanced FIFO Mode	
Priority	
Time-Stamping Normal Mode	
	324
Oppillator Talaranga	007
Oscillator Tolerance	
Overview	273
Overview Phase Buffer Segments	273 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers	273 335 294
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments	273 335 294 337
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment	273 335 294 337 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point	273 335 294 337 335 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period	273 335 294 337 335 335 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization	273 335 294 337 335 335 335 337 336
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard	273 335 294 337 335 335 337 336 336
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization	273 335 294 337 335 335 337 336 336 336 336
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules	273 335 294 337 335 335 337 336 336 336 336 336
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment	273 335 294 337 335 335 337 336 336 336 336 336 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta	273 335 294 337 335 335 336 336 336 336 336 335 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table)	273 335 294 337 335 335 336 336 336 336 336 335 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table)	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes	273 335 294 337 335 335 337 336 336 336 336 336 335 335 340 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode	273 335 294 337 335 335 337 336 336 336 336 336 335 335 340 174 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode	273 335 294 337 335 335 337 336 336 336 336 336 335 335 340 174 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on	273 335 294 337 335 335 336 336 336 336 336 335 335 340 174 174 408
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions	273 335 294 337 335 335 336 336 336 336 336 335 340 174 174 408 31
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions	273 335 294 337 335 335 335 336 336 336 336 336 336 335 340 174 174 408 31
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1)	273 335 294 337 335 335 335 336 336 336 336 336 336 335 340 174 174 408 31
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module).	273 335 294 337 335 335 335 336 336 336 336 336 335 335
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 174 408 31 415 173
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration Pin Configurations for ECCP1	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 174 408 31 415 173
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration Pin Configurations for ECCP1	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 408 31 415 173 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration Pin Configurations for ECCP1 PWM Mode. See PWM (ECCP1 Module). Timer Resources	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 408 31 415 173 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Programming Time Segments Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions Standard PWM Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration Pin Configurations for ECCP1 PWM Mode. See PWM (ECCP1 Module). Timer Resources Enhanced PWM Mode. See PWM (ECCP1 Module).	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 408 31 415 173 174
Overview Phase Buffer Segments Programmable TX/RX and Auto-RTR Buffers Propagation Segment Sample Point Shortening a Bit Period Synchronization Hard Resynchronization Rules Synchronization Segment Time Quanta Values for ICODE (table) ECCP1 Capture and Compare Modes Standard PWM Mode Effect on Standard PIC Instructions 91, Effects of Power Managed Modes on Various Clock Sources Electrical Characteristics Enhanced Capture/Compare/PWM (ECCP1) Capture Mode. See Capture (ECCP1 Module). Outputs and Configuration Pin Configurations for ECCP1 PWM Mode. See PWM (ECCP1 Module). Timer Resources	273 335 294 337 335 335 337 336 336 336 336 336 336 335 335 340 174 408 31 415 173 174

Example Frequencies/Resolutions	176
Full-Bridge Application Example	
Full-Bridge Mode	
Half-Bridge Mode	
Half-Bridge Output Mode Applications	
Example	
Output Configurations	
Output Relationships (Active-High)	
Output Relationships (Active-Low)	
Period	
Programmable Dead-Band Delay	
Setup	
Start-up Considerations	
TMR2 to PR2 Match	

Q

Q Clock	 170,	176
a chook	 ,	

R

RAM. See Data Memory.	
RC Oscillator	25
RCIO Oscillator Mode	25
RC_IDLE Mode	
RC_RUN Mode	35
RCALL	391
RCON Register	
Bit Status During Initialization	
Reader Response	
Register File	
Register File Summary	
Registers	
ADCON0 (A/D Control 0)	247
ADCON1 (A/D Control 1)	
ADCONI (A/D Control 0)	
ADCON2 (A/D Control 2)	
BAUDCON (Baud Rate Control)	
BIE0 (Buffer Interrupt Enable 0)	
BnCON (TX/RX Buffer n Control,	
Receive Mode)	
BnCON (TX/RX Buffer n Control,	
Transmit Mode)	
BnDLC (TX/RX Buffer n Data Length Code	
in Receive Mode)	300
BnDLC (TX/RX Buffer n Data Length Code	
in Transmit Mode)	301
BnDm (TX/RX Buffer n Data Field Byte m	
in Receive Mode)	
BnDm (TX/RX Buffer n Data Field Byte m	
in Transmit Mode)	
BnEIDH (TX/RX Buffer n Extended Identifier,	
High Byte in Receive Mode)	
BnEIDH (TX/RX Buffer n Extended Identifier,	
High Byte in Transmit Mode)	
BnEIDL (TX/RX Buffer n Extended Identifier,	
Low Byte in Receive Mode)	
BnEIDL (TX/RX Buffer n Extended Identifier,	
Low Byte in Transmit Mode)	299
BnSIDH (TX/RX Buffer n Standard Identifier,	
High Byte in Receive Mode)	296
BnSIDH (TX/RX Buffer n Standard Identifier,	
High Byte in Transmit Mode)	296
BnSIDL (TX/RX Buffer n Standard Identifier,	200
Low Byte in Receive Mode)	207
BnSIDL (TX/RX Buffer n Standard Identifier,	
•	207
Low Byte in Transmit Mode)	
BRGCON1 (Baud Rate Control 1)	
BRGCON2 (Baud Rate Control 2)	

BRGCON3 (Baud Rate Control 3)	. 313
BSEL0 (Buffer Select 0)	
CANCON (CAN Control)	
CANSTAT (CAN Status)	. 277
CCP1CON (Capture/Compare/PWM Control)	. 163
CIOCON (CAN I/O Control)	
CMCON (Comparator Control)	. 257
COMSTAT (CAN Communication Status)	. 281
CONFIG1H (Configuration 1 High)	. 344
CONFIG2H (Configuration 2 High)	
CONFIG2L (Configuration 2 Low)	
CONFIG3H (Configuration 3 High)	
CONFIG4L (Configuration 4 Low)	
CONFIG5H (Configuration 5 High)	
CONFIG5L (Configuration 5 Low)	
CONFIG6H (Configuration 6 High)	
CONFIG6L (Configuration 6 Low)	
CONFIG7H (Configuration 7 High)	
CONFIG7L (Configuration 7 Low)	
CVRCON (Comparator Voltage	
Reference Control)	263
Device ID Register 1	351
Device ID Register 2	
ECANCON (Enhanced CAN Control)	
ECCP1AS (ECCP1 Auto-Shutdown Control)	183
ECCP1CON (Enhanced	. 100
Capture/Compare/PWM Control)	173
ECCP1DEL (PWM Configuration)	182
EECON1 (Data EEPROM Control 1)	
HLVDCON (HLVD Control)	
INTCON (Interrupt Control)	
INTCON2 (Interrupt Control 2)	
INITCONI2 (Interrupt Control 2)	117
INTCON3 (Interrupt Control 3)	
IPR1 (Peripheral Interrupt Priority 1)	. 124
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2)	. 124 . 125
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126	. 124 . 125 , 317
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0)	. 124 . 125 , 317 . 307
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1)	. 124 . 125 , 317 . 307 . 308
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2)	. 124 . 125 , 317 . 307 . 308 . 309
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123 PIR1 (Peripheral Interrupt	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 , 316
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123 PIR1 (Peripheral Interrupt Request (Flag) 1)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 , 316
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3)123 PIR1 (Peripheral Interrupt Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 30 27 . 121 . 122 . 316 . 118
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123 PIR1 (Peripheral Interrupt Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123 PIR1 (Peripheral Interrupt Enable 3) 124 Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) 120 RCON (Reset Control)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 123 PIR1 (Peripheral Interrupt Enable 3) 124 Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) 120 RCON (Reset Control) 42 RCSTA (Receive Status and Control)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 127 PIR3 (Peripheral Interrupt Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) 120 RCON (Reset Control) 42 RCSTA (Receive Status and Control) RXB0CON (Receive Buffer 0 Control)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 127 PIR2 (Peripheral Interrupt Enable 3) 128 PIR3 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) 120 RCON (Reset Control) 42 RCSTA (Receive Status and Control) RXB0CON (Receive Buffer 0 Control) RXB1CON (Receive Buffer 1 Control)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) PIR1 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Enable 3) PIR3 (Peripheral Interrupt Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) RCON (Reset Control) 42 RCSTA (Receive Status and Control) RXB0CON (Receive Buffer 0 Control) RXB1CON (Receive Buffer 1 Control) RXBnDLC (Receive Buffer n	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) 127 PIR2 (Peripheral Interrupt Enable 3) 128 PIR3 (Peripheral Interrupt Enable 3) 129 PIR3 (Peripheral Interrupt Request (Flag) 1) PIR3 (Peripheral Interrupt Request (Flag) 3) 120 RCON (Reset Control) 42 RCSTA (Receive Status and Control) RXB0CON (Receive Buffer 0 Control) RXB1CON (Receive Buffer 1 Control) RXBnDLC (Receive Buffer n Data Length Code)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) PIR1 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Enable 3) PIR3 (Peripheral Interrupt Enable 3) Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) RCON (Reset Control) ACON (Reset Control) QRCON (Receive Buffer 0 Control) RXBOCON (Receive Buffer 1 Control) RXBNDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3) 126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) PIR1 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) PIR3 (Peripheral Interrupt Request (Flag) 3) RCON (Reset Control) ACCON (Reset Control) ACCON (Receive Buffer 0 Control) RXB1CON (Receive Buffer 1 Control) RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 30 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) PIR1 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) PIR3 (Peripheral Interrupt Request (Flag) 4) PIR3 (Peripheral Interrupt Request (Piripheral Interrupt Reques	. 124 . 125 . 317 . 307 . 308 . 309 . 310 27 . 121 . 122 . 316 . 118 . 119 . 229 . 287 . 289 . 292 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)126 MSEL0 (Mask Select 0) MSEL1 (Mask Select 1) MSEL2 (Mask Select 2) MSEL3 (Mask Select 3) OSCCON (Oscillator Control) OSCCON (Oscillator Control) OSCTUNE (Oscillator Tuning) PIE1 (Peripheral Interrupt Enable 1) PIE2 (Peripheral Interrupt Enable 2) PIE3 (Peripheral Interrupt Enable 3) PIR1 (Peripheral Interrupt Enable 3) PIR2 (Peripheral Interrupt Request (Flag) 1) PIR2 (Peripheral Interrupt Request (Flag) 2) PIR3 (Peripheral Interrupt Request (Flag) 3) PIR3 (Peripheral Interrupt Request (Flag) 4) PIR3 (Peripheral Interrupt Request (Flag) 5) PIR3 (Peripheral Interrupt Request (Flag) 6) PIR3 (Peripheral Interrupt Request (Flag) 7) PIR3 (Peripheral Interrupt Request (Piripheral Interr	. 124 . 125 . 317 . 307 . 308 . 309 . 310 27 . 121 . 122 . 316 . 118 . 119 . 229 . 287 . 289 . 292 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289 . 289 . 292 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289 . 289 . 292 . 292
IPR1 (Peripheral Interrupt Priority 1) IPR2 (Peripheral Interrupt Priority 2) IPR3 (Peripheral Interrupt Priority 3)	. 124 . 125 . 317 . 307 . 308 . 309 . 310 27 . 121 . 122 . 316 . 118 . 119 . 315 . 127 . 229 . 287 . 289 . 292 . 291 . 291