



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	25MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4620-e-ml

Email: info@E-XFL.COM

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

3.1.3 CLOCK TRANSITIONS AND STATUS INDICATORS

The length of the transition between clock sources is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Three bits indicate the current clock source and its status. They are:

- OSTS (OSCCON<3>)
- IOFS (OSCCON<2>)
- T1RUN (T1CON<6>)

In general, only one of these bits will be set while in a given power-managed mode. When the OSTS bit is set, the primary clock is providing the device clock. When the IOFS bit is set, the INTOSC output is providing a stable, 8 MHz clock source to a divider that actually drives the device clock. When the T1RUN bit is set, the Timer1 oscillator is providing the clock. If none of these bits are set, then either the INTRC clock source is clocking the device, or the INTOSC source is not yet stable.

If the internal oscillator block is configured as the primary clock source by the FOSC3:FOSC0 Configuration bits, then both the OSTS and IOFS bits may be set when in PRI_RUN or PRI_IDLE modes. This indicates that the primary clock (INTOSC output) is generating a stable, 8 MHz output. Entering another power-managed RC mode at the same frequency would clear the OSTS bit.

- Note 1: Caution should be used when modifying a single IRCF bit. If VDD is less than 3V, it is possible to select a higher clock speed than is supported by the low VDD. Improper device operation may result if the VDD/FOSC specifications are violated.
 - 2: Executing a SLEEP instruction does not necessarily place the device into Sleep mode. It acts as the trigger to place the controller into either the Sleep mode or one of the Idle modes, depending on the setting of the IDLEN bit.

3.1.4 MULTIPLE SLEEP COMMANDS

The power-managed mode that is invoked with the SLEEP instruction is determined by the setting of the IDLEN bit at the time the instruction is executed. If another SLEEP instruction is executed, the device will enter the power-managed mode specified by IDLEN at that time. If IDLEN has changed, the device will enter the new power-managed mode specified by the new setting.

3.2 Run Modes

In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

3.2.1 PRI_RUN MODE

The PRI_RUN mode is the normal, full-power execution mode of the microcontroller. This is also the default mode upon a device Reset unless Two-Speed Start-up is enabled (see **Section 23.3 "Two-Speed Start-up"** for details). In this mode, the OSTS bit is set. The IOFS bit may be set if the internal oscillator block is the primary clock source (see **Section 2.7.1 "Oscillator Control Register"**).

3.2.2 SEC_RUN MODE

The SEC_RUN mode is the compatible mode to the "clock switching" feature offered in other PIC18 devices. In this mode, the CPU and peripherals are clocked from the Timer1 oscillator. This gives users the option of lower power consumption while still using a high-accuracy clock source.

SEC_RUN mode is entered by setting the SCS1:SCS0 bits to '01'. The device clock source is switched to the Timer1 oscillator (see Figure 3-1), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.

Note: The Timer1 oscillator should already be running prior to entering SEC_RUN mode. If the T1OSCEN bit is not set when the SCS1:SCS0 bits are set to '01', entry to SEC_RUN mode will not occur. If the Timer1 oscillator is enabled, but not yet running, device clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

On transitions from SEC_RUN mode to PRI_RUN, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-2). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the clock. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run.

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
SPBRGH	EUSART Bau	0000 0000	51, 206							
SPBRG	EUSART Bau	0000 0000	51, 206							
RCREG	EUSART Red	0000 0000	51, 213							
TXREG	EUSART Tra	0000 0000	51, 211							
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	51, 202
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	51, 203
EEADRH	—	—	—	—	—	—	EEPROM Add	r Register High	00	51, 73
EEADR	EEPROM Ad	dress Registe	r						0000 0000	51, 80, 73
EEDATA	EEPROM Da		0000 0000	51, 80, 73						
EECON2	EEPROM Co	ntrol Register	2 (not a physi	cal register)					0000 0000	51, 80, 73
EECON1	EEPGD	CFGS	_	FREE	WRERR	WREN	WR	RD	xx-0 x000	51, 81, 74
IPR2	OSCFIP	CMIP	_	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	11-1 1111	52, 119
PIR2	OSCFIF	CMIF	_	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	00-0 0000	52, 115
PIE2	OSCFIE	CMIE	_	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	00-0 0000	52, 117
IPR1	PSPIP ⁽²⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	52, 118
PIR1	PSPIF ⁽²⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	52, 114
PIE1	PSPIE ⁽²⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	52, 116
OSCTUNE	INTSRC	PLLEN ⁽³⁾	—	TUN4	TUN3	TUN2	TUN1	TUN0	00-0 0000	27, 52
TRISE ⁽²⁾	IBF	OBF	IBOV	PSPMODE	_	TRISE2	TRISE1	TRISE0	0000 -111	52, 104
TRISD ⁽²⁾	PORTD Data	Direction Cor	ntrol Register						1111 1111	52, 100
TRISC	PORTC Data	Direction Cor	ntrol Register						1111 1111	52, 97
TRISB	PORTB Data	Direction Cor	trol Register						1111 1111	52, 94
TRISA	TRISA7 ⁽⁵⁾	TRISA6 ⁽⁵⁾	Data Directio	n Control Reg	ister for PORT	A			1111 1111	52, 91
LATE ⁽²⁾	—	—	—	_	-	PORTE Data (Read and W	Latch Register	ch)	xxx	52, 103
LATD ⁽²⁾	PORTD Data	Latch Registe	er (Read and V	Vrite to Data L	atch)				xxxx xxxx	52, 100
LATC	PORTC Data	Latch Registe	er (Read and V	Vrite to Data L	atch)				xxxx xxxx	52, 97
LATB	PORTB Data	Latch Registe	er (Read and V	Vrite to Data L	atch)				xxxx xxxx	52, 94
LATA	LATA7 ⁽⁵⁾	LATA6 ⁽⁵⁾	PORTA Data	Latch Registe	er (Read and V	Vrite to Data L	atch)		xxxx xxxx	52, 91
PORTE	_	_	_	_	RE3 ⁽⁴⁾	RE2 ⁽²⁾	RE1 ⁽²⁾	RE0 ⁽²⁾	xxxx	52, 103
PORTD ⁽²⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	52, 100
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	XXXX XXXX	52, 97
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	52, 94
PORTA	RA7 ⁽⁵⁾	RA6 ⁽⁵⁾	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000	52, 91

|--|

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition Note

The SBOREN bit is only available when the BOREN1:BOREN0 Configuration bits = 01; otherwise, it is disabled and reads as '0'. See 1: Section 4.4 "Brown-out Reset (BOR)".

2: These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.

The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See Section 2.6.4 "PLL in 3: INTOSC Modes".

The RE3 bit is only available when Master Clear Reset is disabled (MCLRE Configuration bit = 0); otherwise, RE3 reads as '0'. This bit is 4: read-only.

RA6/RA7 and their associated latch and direction bits are individually configured as port pins based on various primary oscillator modes. 5: When disabled, these bits read as '0'.

6: Bit 7 and bit 6 are cleared by user software or by a POR.

7.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 7-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 7-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 7-1: READING A FLASH PROGRAM MEMORY WORD

	MOVLW MOVWF MOVUW MOVWF MOVLW MOVWF	CODE_ADDR_UPPER TBLPTRU CODE_ADDR_HIGH TBLPTRH CODE_ADDR_LOW TBLPTRL	;;	Load TBLPTR with the base address of the word
READ_WORD	א א מם זמיד			road into TARIAT and ingroment
	IBLKD +		'	Teau INCO TABLAT AND INCLEMENT
	MOVF	TABLAT, W	;	get data
	MOVWF	WORD_EVEN		
	TBLRD*+		;	read into TABLAT and increment
	MOVF	TABLAT, W	;	get data
	MOVWF	WORD_ODD		

13.0 TIMER2 MODULE

The Timer2 module timer incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 13-1), which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 13-1.

13.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options; these are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/ postscaler (see Section 13.2 "Timer2 Interrupt").

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7	•	•					bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplem	nented bit, read	d as '0'	
-n = Value at I	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	Iown
bit 7	Unimplement	ted: Read as 'd)'				
bit 6-3	T2OUTPS3:T	20UTPS0: Tim	ner2 Output Po	stscale Select	bits		
	0000 = 1:1 Po	ostscale					
	0001 = 1:2 Po	ostscale					
	•						
	•						
	1111 = 1:16 F	Postscale					
bit 2	TMR2ON: Tin	ner2 On bit					
	1 = Timer2 is	on					
	0 = Timer2 is	off					
bit 1-0	T2CKPS1:T2	CKPS0: Timer	2 Clock Presca	ale Select bits			
	00 = Prescale	eris 1					
	01 = Prescale	eris 4 Aris 16					
	$TX = r^{1}escale$	115 10					

The CCPR2H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP2 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

EQUATION 15-3:

PWM Resolution (max) =
$$\frac{\log(\frac{Fosc}{FPWM})}{\log(2)}$$
 bits

Note: If the PWM duty cycle value is longer than the PWM period, the CCP2 pin will not be cleared.

TABLE 15-4:	EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

15.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 16.4.7 "Enhanced PWM Auto-Shutdown"**.

Auto-shutdown features are not available for CCP2.

15.4.4 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCPx module for PWM operation:

- 1. Set the PWM period by writing to the PR2 register.
- 2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
- 3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
- 4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
- 5. Configure the CCPx module for PWM operation.

16.4.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the ECCP module for PWM operation:

- 1. Configure the PWM pins, P1A and P1B (and P1C and P1D, if used), as inputs by setting the corresponding TRIS bits.
- 2. Set the PWM period by loading the PR2 register.
- 3. If auto-shutdown is required, do the following:
 - Disable auto-shutdown (ECCP1AS = 0)
 - Configure source (FLT0, Comparator 1 or Comparator 2)
 - Wait for non-shutdown condition
- Configure the ECCP module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
 - Select one of the available output configurations and direction with the P1M1:P1M0 bits.
 - Select the polarities of the PWM output signals with the CCP1M3:CCP1M0 bits.
- 5. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
- 6. For Half-Bridge Output mode, set the deadband delay by loading PWM1CON<6:0> with the appropriate value.
- 7. If auto-shutdown operation is required, load the ECCP1AS register:
 - Select the auto-shutdown sources using the ECCPAS2:ECCPAS0 bits.
 - Select the shutdown states of the PWM output pins using the PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits.
 - Set the ECCPASE bit (ECCP1AS<7>).
 - Configure the comparators using the CMCON register.
 - Configure the comparator inputs as analog inputs.
- 8. If auto-restart operation is required, set the PRSEN bit (PWM1CON<7>).
- 9. Configure and start TMR2:
 - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
 - Set the TMR2 prescale value by loading the T2CKPS bits (T2CON<1:0>).
 - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
- 10. Enable PWM outputs after a new PWM cycle has started:
 - Wait until TMRx overflows (TMRxIF bit is set).
 - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRIS bits.
 - Clear the ECCPASE bit (ECCP1AS<7>).

16.4.10 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2 will not increment and the state of the module will not change. If the ECCP pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from INTOSC and the postscaler may not be stable immediately.

In PRI_IDLE mode, the primary clock will continue to clock the ECCP module without change. In all other power-managed modes, the selected power-managed mode clock will clock Timer2. Other power-managed mode clocks will most likely be different than the primary clock frequency.

16.4.10.1 Operation with Fail-Safe Clock Monitor

If the Fail-Safe Clock Monitor is enabled, a clock failure will force the device into the power-managed RC_RUN mode and the OSCFIF bit (PIR2<7>) will be set. The ECCP will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

See the previous section for additional details.

16.4.11 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the CCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.



18.0 ENHANCED UNIVERSAL SYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a halfduplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous Master (half duplex) with selectable clock polarity
- Synchronous Slave (half duplex) with selectable clock polarity

The pins of the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT as a USART:

- SPEN bit (RCSTA<7>) must be set (= 1)
- TRISC<7> bit must be set (= 1)
- TRISC<6> bit must be set (= 1)

Note:	The EUSART control will automatically
	reconfigure the pin from input to output as
	needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 18-1, Register 18-2 and Register 18-3, respectively.

18.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTA<3> and TXSTA<5>) are set while the Transmit Shift Register is loaded with data. Note that the value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 18-10 for the timing of the Break character sequence.

18.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN bus master.

- Configure the EUSART for the desired mode. 1.
- 2. Set the TXEN and SENDB bits to set up the Break character.
- 3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
- Write '55h' to TXREG to load the Sync character 4 into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

18.2.6 **RECEIVING A BREAK CHARACTER**

The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in Section 18.2.4 "Auto-Wake-up on Sync Break Character". By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABD bit once the TXIF interrupt is observed.



FIGURE 18-10: SEND BREAK CHARACTER SEQUENCE

18.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RX pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

- 1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
- 2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.

- 3. Ensure bits, CREN and SREN, are clear.
- 4. If interrupts are desired, set enable bit, RCIE.
- 5. If 9-bit reception is desired, set bit, RX9.
- 6. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
- 7. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
- 8. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
- 9. Read the 8-bit received data by reading the RCREG register.
- 10. If any error occurred, clear the error by clearing bit, CREN.
- 11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.



FIGURE 18-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

TABLE 18-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART R	eceive Regi	ster						51
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	RGH EUSART Baud Rate Generator Register High Byte								51
SPBRG	PBRG EUSART Baud Rate Generator Register Low Byte								51
Legend: -	— = unimple	mented rea	d as '0' Sha	aded cells a	re not used :	for synchror	ous master	reception	

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

Note 1: These bits are unimplemented on 28-pin devices and read as '0'.

18.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode and bit, SREN, which is a "don't care" in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the chip from the lowpower mode. If the global interrupt is enabled, the program will branch to the interrupt vector. To set up a Synchronous Slave Reception:

- Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
- 2. If interrupts are desired, set enable bit, RCIE.
- 3. If 9-bit reception is desired, set bit, RX9.
- 4. To enable reception, set enable bit, CREN.
- 5. Flag bit, RCIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCIE, was set.
- 6. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
- 7. Read the 8-bit received data by reading the RCREG register.
- 8. If any error occurred, clear the error by clearing bit, CREN.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART F	Receive Regi	ster						51
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	51
SPBRGH	3RGH EUSART Baud Rate Generator Register High Byte								
SPBRG	'BRG EUSART Baud Rate Generator Register Low Byte								

TABLE 18-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

Note 1: These bits are unimplemented on 28-pin devices and read as '0'.

REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-q ⁽¹⁾	R/W-q ⁽¹⁾	R/W-q ⁽¹⁾
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
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 7-6	Unimplemented:	Read as '0'
	•••••••••••••••••••••••••••••••••••••••	

bit 5	VCFG1: Voltage Reference Configuration bit (VREF- source)
	1 = VREF- (AN2)
	0 = Vss
bit 4	VCFG0: Voltage Reference Configuration bit (VREF+ source)
	1 = VREF+ (AN3)
	0 = VDD

bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7 ⁽²⁾	AN6 ⁽²⁾	AN5 ⁽²⁾	AN4	AN3	AN2	AN1	ANO
0000 (1)	А	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0001	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0010	А	А	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0011	D	А	Α	Α	Α	Α	Α	Α	А	Α	А	Α	Α
0100	D	D	Α	Α	Α	Α	Α	Α	А	Α	А	Α	Α
0101	D	D	D	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0110	D	D	D	D	Α	Α	Α	Α	А	Α	Α	Α	Α
0111(1)	D	D	D	D	D	А	Α	A	А	А	А	Α	Α
1000	D	D	D	D	D	D	Α	Α	Α	Α	Α	Α	Α
1001	D	D	D	D	D	D	D	Α	Α	Α	Α	Α	Α
1010	D	D	D	D	D	D	D	D	Α	Α	Α	Α	Α
1011	D	D	D	D	D	D	D	D	D	Α	Α	Α	Α
1100	D	D	D	D	D	D	D	D	D	D	Α	Α	Α
1101	D	D	D	D	D	D	D	D	D	D	D	Α	Α
1110	D	D	D	D	D	D	D	D	D	D	D	D	Α
1111	D	D	D	D	D	D	D	D	D	D	D	D	D
A = Applog input $D = Digital 1/Q$													

A = Analog input

D = Digital I/O

Note 1: The POR value of the PCFG bits depends on the value of the PBADEN Configuration bit. When PBADEN = 1, PCFG<2:0> = 000; when PBADEN = 0, PCFG<2:0> = 111.

2: AN5 through AN7 are available only on 40/44-pin devices.

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and Vss), or the voltage level on the RA3/AN3/ VREF+ and RA2/AN2/VREF-/CVREF pins.

The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted.

Each port pin associated with the A/D converter can be configured as an analog input, or as a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/DONE bit (ADCON0 register) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 19-1.



FIGURE 19-1: A/D BLOCK DIAGRAM

19.6 A/D Conversions

Figure 19-4 shows the operation of the A/D converter after the GO/DONE bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 19-5 shows the operation of the A/D converter after the GO/DONE bit has been set and the ACQT2:ACQT0 bits are set to '010' and selecting a 4 TAD acquisition time before the conversion starts.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

Note: The GO/DONE bit should **NOT** be set in the same instruction that turns on the A/D.

19.7 Discharge

The discharge phase is used to initialize the value of the capacitor array. The array is discharged before every sample. This feature helps to optimize the unitygain amplifier, as the circuit always needs to charge the capacitor array, rather than charge/discharge based on previous measure values.





FIGURE 19-5: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



Bit Clear	f			E		
BCF f, b	BCF f, b {,a}					
$\begin{array}{l} 0\leq f\leq 255\\ 0\leq b\leq 7\\ a\in [0,1] \end{array}$				c		
$0 \rightarrow f < b >$				S		
None				E		
1001	bbba	ffff	ffff			
Bit 'b' in reg If 'a' is '0', t If 'a' is '1', t GPR bank. If 'a' is '0' a set is enab in Indexed mode wher Section 24 Bit-Oriente Literal Offe	Bit 'b' in register 'f' is cleared. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed					
1				I		
1						
Q2	Q3	3	Q4			
Read register 'f'	Proce Dat	ess a re	Write gister 'f'			
BCF F ion EG = C7 n EG = 47	FLAG_RE 7h ′h	G, 7,	0	E		
	Bit Clear BCF f, b $0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$ $0 \rightarrow f < b >$ None 1001 Bit 'b' in reg If 'a' is '0', i If 'a' is '0' a set is enab in Indexed mode when Section 24 Bit-Oriente Literal Offe 1 1 2 Read register 'f' BCF I ion EG = C7 n EG = 47	Bit Clear fBCFf, b {,a} $0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$ $0 \rightarrow f < b >$ None1001bbbaBit 'b' in register 'f' isIf 'a' is '0', the AccessIf 'a' is '0', the AccessIf 'a' is '0' and the exist is enabled, this isin Indexed Literal Ofmode whenever $f \le$ Section 24.2.3 "ByBit-Oriented InstructionLiteral Offset Model11Q2Q3ReadProcessorregister 'f'DateBCFFLAG_REionEG =CG =47h	Bit Clear fBCFf, b {,a} $0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$ $0 \rightarrow f < b >$ None1001bbbafffBit 'b' in register 'f' is cleared.If 'a' is '0', the Access Bank isIf 'a' is '0', the Access Bank isIf 'a' is '0' and the extended inset is enabled, this instructionin Indexed Literal Offset Addredmode whenever $f \le 95$ (5Fh).Section 24.2.3 "Byte-OrientedBit-Oriented Instructions inLiteral Offset Mode" for deta11Q2Q3Readregister 'f'Dataregister 'f'DatarefBCFFLAG_REG, 7,ionEG =C7hneG =47h	Bit Clear fBCFf, b {,a} $0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$ $0 \rightarrow f cb>$ None1001bbbaffffffffBit 'b' in register 'f' is cleared.If 'a' is '0', the Access Bank is selected.If 'a' is '0', the Access Bank is select the GPR bank.If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). SeeSection 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.11Q2Q3Q4ReadProcessWrite register 'f'BCFFLAG_REG, 7, 0ionEG =C7hneG =47h		

BN Branch if Negative								
Synta	ax:	BN n	BN n					
Oper	ands:	-128 ≤ n ≤ ′	$-128 \le n \le 127$					
Oper	ation:	if Negative (PC) + 2 + 2	bit is '1', 2n → PC					
Statu	is Affected:	None						
Enco	oding:	1110	0110 nn	nn nnnn				
Desc	ription:	If the Negar program wi The 2's con added to the incremente instruction, PC + 2 + 2r two-cycle ir	If the Negative bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.					
Word	ls:	1	1					
Cycle	es:	1(2)						
Q C If Ju	ycle Activity: Imp:							
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'n'	Process Data	Write to PC				
	No operation	No operation	No operation	No operation				
lf No	o Jump:							
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'n'	Process Data	No operation				
Exan	nple:	HERE	BN Jump					
	Before Instruc PC After Instructio If Negativ PC If Negativ PC	tion = ad on = 1; /e = 1; /e = 0; /e = 0;	dress (HERE dress (Jump dress (HERE)))+2))				

24.2.2 EXTENDED INSTRUCTION SET

ADD	ADDFSR Add Literal to FSR						
Synta	ax:	ADDFSR	ADDFSR f, k				
Oper	ands:	$0 \le k \le 63$					
		f ∈ [0, 1,	2]				
Oper	ation:	FSR(f) + I	$\star \rightarrow FSR($	f)			
Statu	is Affected:	None					
Enco	oding:	1110	1000	ffk	k	kkkk	
Desc	ription:	The 6-bit	The 6-bit literal 'k' is added to the				
		contents of	contents of the FSR specified by 'f'.				
Word	ls:	1					
Cycle	es:	1					
Q Cycle Activity:							
	Q1	Q2	Q3			Q4	
	Decode	Read	Proce	SS	۷	Vrite to	
		literal 'k'	Data	a		FSR	

ADDFSR 2, 23h

Syntax:	ADDULNK k			
Operands:	$0 \le k \le 63$			
Operation:	$FSR2 + k \rightarrow FSR2$,			
	$(TOS) \rightarrow PC$			
Status Affected:	None			
Encoding:	1110 1000 11kk kkkk			
	contents of FSR2. A RETURN is then executed by loading the PC with the TOS. The instruction takes two cycles to execute; a NOP is performed during the second cycle. This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.			
Words:	1			
Cycles:	2			

Add Literal to FSR2 and Return

Q Cycle Activity:

ADDULNK

Q1	Q2	Q3	Q4
Decode	Read	Process	Write to
	literal 'k'	Data	FSR
No	No	No	No
Operation	Operation	Operation	Operation

Example: ADDULNK 23h

Before Instru	ction	
FSR2	=	03FFh
PC	=	0100h
After Instruct	ion	
FSR2	=	0422h
PC	=	(TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction syntax then becomes: {label} instruction argument(s).

Example:

Before Instruction

After Instruction

FSR2

FSR2 = 03FFh

= 0422h

25.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft[®] Windows[®] 32-bit operating system were chosen to best make these features available in a simple, unified application.

25.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC[®] Flash MCUs and dsPIC[®] Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

25.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming[™] (ICSP[™]) protocol, offers costeffective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

25.10 MPLAB PM3 Device Programmer

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







44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	MILLIMETERS			
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

Sleep

OSC1 and OSC2 Pin States
Software Simulator (MPLAB SIM)
Special Event Trigger. See Compare (ECCP Mode).
Special Event Trigger. See Compare (ECCP Module).
Special Features of the CPU
Special Function Registers
Map62
SPI Mode (MSSP)
Associated Registers169
Bus Mode Compatibility169
Effects of a Reset169
Enabling SPI I/O165
Master Mode166
Master/Slave Connection165
Operation164
Operation in Power-Managed Modes
Serial Clock161
Serial Data In161
Serial Data Out161
Slave Mode167
Slave Select161
Slave Select Synchronization167
SPI Clock166
Typical Connection165
<u>SS</u>
SSPOV
SSPOV Status Flag191
SSPSTAT Register
R/W Bit
Stack Full/Underflow Resets
Standard Instructions
STATUS Register
SUBFSR
SUBFWB
SUBLW
SUBULNK
SUBWF
SUBWFB
SWAPF
т

Table Reads/Table Writes	
TBLRD	
TBLWT	306
Time-out in Various Situations (table)	45
Timer0	123
Associated Registers	125
Operation	
Overflow Interrupt	
Prescaler	
Prescaler Assignment (PSA Bit)	
Prescaler Select (T0PS2:T0PS0 Bits)	
Prescaler. See Prescaler, Timer0.	
Reads and Writes in 16-Bit Mode	124
Source Edge Select (T0SE Bit)	
Source Select (T0CS Bit)	
Switching Prescaler Assignment	
Timer1	127
16-Bit Read/Write Mode	129
Associated Registers	131
Interrunt	130
Operation	
Oscillator	127 120
	127, 129
Layour Considerations	

Overflow Interrupt	127
Resetting, Using the CCP Special Event Trigger	130
Special Event Trigger (ECCP)	148
TMR1H Register	127
TMR11 Register	127
Lise as a Real-Time Clock	130
	100
	133
Associated Registers	134
Interrupt	134
Operation	133
Output	134
PR2 Register14	44, 149
TMR2 to PR2 Match Interrupt 14	44, 149
Timer3	135
16-Bit Read/Write Mode	137
Associated Registers	137
Operation	126
	130
Oscillator	35, 137
	35, 137
Special Event Trigger (CCP)	137
TMR3H Register	135
TMR3L Register	135
Timing Diagrams	
A/D Conversion	360
Acknowledge Seguence	194
Asynchronous Recention	214
Asynchronous Tronomission	217
Asynchionous Transmission	212
Asynchronous Transmission	
(Back to Back)	212
Automatic Baud Rate Calculation	210
Auto-Wake-up Bit (WUE) During	
Normal Operation	215
Auto-Wake-up Bit (WLIE) During Sleep	045
Auto-wake-up bit (woll) builing Sleep	215
Baud Rate Generator with Clock Arbitration	215
Baud Rate Generator with Clock Arbitration	215 188 210
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration	215 188 210
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition	215 188 210 197
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR)	215 188 210 197 345
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated	215 188 210 197 345
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1)	215 188 210 197 345 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated	215 188 210 197 345 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2)	215 188 210 197 345 198 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start	215 188 210 197 345 198 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0)	215 188 210 197 345 198 198 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop	215 188 210 197 198 198 198
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1)	215 188 210 197 345 198 198 197 199
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop	215 188 210 197 345 198 198 197 199
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2)	215 188 210 197 345 198 198 197 199
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition During a Start Condition (Case 1) Bus Collision During a Start Condition (Case 1) Bus Collision During a Start Condition (Case 1) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2)	215 188 210 197 345 198 198 197 199
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2)	215 188 210 197 345 198 198 197 199 199
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2)	215 188 210 197 198 198 197 199 199 196
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge	215 188 210 197 198 198 197 199 199 196 195
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules)	215 188 210 197 198 198 198 197 199 199 195 347
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210 197 197 198 198 198 197 199 199 195 347 344
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210 197 197 198 198 198 198 199 199 199 195 347 344 181
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge CLKO and I/O Clock Synchronization 	215 188 210 197 345 198 198 198 198 199 199 199 199 195 347 344 181
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210 197 345 198 198 198 198 197 199 199 199 195 347 344 181 57
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210 197 345 198 198 198 198 198 197 199 199 199 347 344 37 350
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) CLKO and I/O Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave)	215 188 210 197 198 198 198 198 197 199 199 195 347 344 57 359
 Baud Rate Generator with Clock Arbitration BRG Overflow Sequence	215 188 210 197 345 198 198 198 198 197 199 199 195 347 344 57 359
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) Clock Synchronization Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave)	215 188 210 197 197 198 198 198 198 197 199 199 199 195 344 344 57 359 358
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave) EUSART Synchronous Transmission (Master/Slave)	215 188 210 197 197 198 198 198 198 197 199 199 199 199 195 344 359 358 358 349
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave) EUSART Synchronous Transmission (Master/Slave) Example SPI Master Mode (CKE = 0) Example SPI Master Mode (CKE = 1)	215 188 210 197 345 198 198 198 198 197 199 199 199 199 195 347 344 57 359 358 350
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave) EUSART Synchronous Transmission (Master/Slave) Example SPI Master Mode (CKE = 0) Example SPI Master Mode (CKE = 0)	215 188 210 197 197 198 198 198 198 197 199 199 199 199 199 199 195 344 357 358 359 350 351
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) Clckk Synchronization Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave) EUSART Synchronous Transmission (Master/Slave) Example SPI Master Mode (CKE = 0) Example SPI Master Mod	215 188 210 197 197 198 198 198 198 197 199 199 199 199 199 199 195 344 357 359 358 351 353
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (SCL = 0) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) CLKO and I/O Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Receive (Master/Slave) EUSART Synchronous Transmission (Master/Slave) Example SPI Master Mode (CKE = 0) Example SPI Master Mode (CKE =	215 188 210 197 345 198 198 198 198 198 197 199 199 195 347 344 359 358 358 351 353 353 353
Baud Rate Generator with Clock Arbitration BRG Overflow Sequence BRG Reset Due to SDA Arbitration During Start Condition Brown-out Reset (BOR) Bus Collision During a Repeated Start Condition (Case 1) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Repeated Start Condition (Case 2) Bus Collision During a Start Condition (Case 1) Bus Collision During a Stop Condition (Case 1) Bus Collision During a Stop Condition (Case 2) Bus Collision During a Stop Condition (Case 2) Bus Collision During Start Condition (SDA Only) Bus Collision for Transmit and Acknowledge Capture/Compare/PWM (All CCP Modules) CLKO and I/O Clock Synchronization Clock/Instruction Cycle EUSART Synchronous Transmission (Master/Slave) Example SPI Master Mode (CKE = 0) Example SPI Master Mode (CKE = 1) Example SPI Slave Mode (CKE = 1) Example SPI Slave Mode (CKE = 1) Example SPI Slave Mode (CKE = 1) <tr< td=""><td> 215 188 210 197 345 198 198 198 198 198 197 199 199 199 195 347 344 359 358 351 351 351 352 352 </td></tr<>	215 188 210 197 345 198 198 198 198 198 197 199 199 199 195 347 344 359 358 351 351 351 352