

Welcome to [E-XFL.COM](https://www.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 "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	56KB (32K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 45x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19197t-i-pt

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 60 (Continued)											
1E37h	CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	uuuu uuuu
1E38h	RF0PPS	—	—	—	RF0PPS4	RF0PPS3	RF0PPS2	RF0PPS1	RF0PPS0	--x xxxx	--uu uuuu
1E39h	RF1PPS	—	—	—	RF1PPS4	RF1PPS3	RF1PPS2	RF1PPS1	RF1PPS0	--xx xxxx	--uu uuuu
1E3Ah	RF2PPS	—	—	—	RF2PPS4	RF2PPS3	RF2PPS2	RF2PPS1	RF2PPS0	--xx xxxx	--uu uuuu
1E3Bh	RF3PPS	—	—	—	RF3PPS4	RF3PPS3	RF3PPS2	RF3PPS1	RF3PPS0	--xx xxxx	--uu uuuu
1E3Ch	RF4PPS	—	—	—	RF4PPS4	RF4PPS3	RF4PPS2	RF4PPS1	RF4PPS0	--xx xxxx	--uu uuuu
1E3Dh	RF5PPS	—	—	—	RF5PPS4	RF5PPS3	RF5PPS2	RF5PPS1	RF5PPS0	--xx xxxx	--uu uuuu
1E3Eh	RF6PPS	—	—	—	RF6PPS4	RF6PPS3	RF6PPS2	RF6PPS1	RF6PPS0	--xx xxxx	--uu uuuu
1E3Fh	RF7PPS	—	—	—	RF7PPS4	RF7PPS3	RF7PPS2	RF7PPS1	RF7PPS0	--xx xxxx	--uu uuuu
1E40h	RG0PPS	—	—	—	RG0PPS4	RG0PPS3	RG0PPS2	RG0PPS1	RG0PPS0	--xx xxxx	--uu uuuu
1E41h	RG1PPS	—	—	—	RG1PPS4	RG1PPS3	RG1PPS2	RG1PPS1	RG1PPS0	--xx xxxx	--uu uuuu
1E42h	RG2PPS	—	—	—	RG2PPS4	RG2PPS3	RG2PPS2	RG2PPS1	RG2PPS0	--xx xxxx	--uu uuuu
1E43h	RG3PPS	—	—	—	RG3PPS4	RG3PPS3	RG3PPS2	RG3PPS1	RG3PPS0	--xx xxxx	--uu uuuu
1E44h	RG4PPS	—	—	—	RG4PPS4	RG4PPS3	RG4PPS2	RG4PPS1	RG4PPS0	--xx xxxx	--uu uuuu
1E45h	—	Unimplemented								----	----
1E46h	RG6PPS	—	—	—	RG6PPS4	RG6PPS3	RG6PPS2	RG6PPS1	RG6PPS0	--xx xxxx	--uu uuuu
1E47h	RG7PPS	—	—	—	RG7PPS4	RG7PPS3	RG7PPS2	RG7PPS1	RG7PPS0	--xx xxxx	--uu uuuu
1E48h	RH0PPS	—	—	—	RH0PPS4	RH0PPS3	RH0PPS2	RH0PPS1	RH0PPS0	--xx xxxx	--uu uuuu
1E49h	RH1PPS	—	—	—	RH1PPS4	RH1PPS3	RH1PPS2	RH1PPS1	RH1PPS0	--xx xxxx	--uu uuuu
1E4Ah	RH2PPS	—	—	—	RH2PPS4	RH2PPS3	RH2PPS2	RH2PPS1	RH2PPS0	--xx xxxx	--uu uuuu
1E4Bh	RH3PPS	—	—	—	RH3PPS4	RH3PPS3	RH3PPS2	RH3PPS1	RH3PPS0	--xx xxxx	--uu uuuu
1E4Ch	—	Unimplemented								----	----
1E4Dh	—	Unimplemented								----	----
1E4Eh	—	Unimplemented								----	----
1E4Fh	—	Unimplemented								----	----

Legend: x = unknown, u = unchanged, c = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Unimplemented data memory locations, read as '0'.

8.16 Register Definitions: Power Control

REGISTER 8-2: PCON0: POWER CONTROL REGISTER 0

R/W/HS-0/q	R/W/HS-0/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR
bit 7							bit 0

Legend:

HC = Bit is cleared by hardware

HS = Bit is set by hardware

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-m/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

q = Value depends on condition

- bit 7 **STKOVF**: Stack Overflow Flag bit
 1 = A Stack Overflow occurred
 0 = A Stack Overflow has not occurred or cleared by firmware
- bit 6 **STKUNF**: Stack Underflow Flag bit
 1 = A Stack Underflow occurred
 0 = A Stack Underflow has not occurred or cleared by firmware
- bit 5 **WDTWV**: WDT Window Violation Flag bit
 1 = A WDT Window Violation Reset has not occurred or set to '1' by firmware
 0 = A WDT Window Violation Reset has occurred (a CLRWDT instruction was executed either without arming the window or outside the window (cleared by hardware))
- bit 4 **RWDT**: Watchdog Timer Reset Flag bit
 1 = A Watchdog Timer Reset has not occurred or set to '1' by firmware
 0 = A Watchdog Timer Reset has occurred (cleared by hardware)
- bit 3 **RMCLR**: MCLR Reset Flag bit
 1 = A MCLR Reset has not occurred or set to '1' by firmware
 0 = A MCLR Reset has occurred (cleared by hardware)
- bit 2 **RI**: RESET Instruction Flag bit
 1 = A RESET instruction has not been executed or set to '1' by firmware
 0 = A RESET instruction has been executed (cleared by hardware)
- bit 1 **POR**: Power-on Reset Status bit
 1 = No Power-on Reset occurred
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
- bit 0 **BOR**: Brown-out Reset Status bit
 1 = No Brown-out Reset occurred
 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

REGISTER 10-3: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE
bit 7						bit 0	

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

- bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the Oscillator Fail Interrupt
0 = Disables the Oscillator Fail Interrupt
- bit 6 **CSWIE:** Clock Switch Complete Interrupt Enable bit
1 = The clock switch module interrupt is enabled
0 = The clock switch module interrupt is disabled
- bit 5-2 **Unimplemented:** Read as '0'
- bit 1 **ADTIE:** Analog-to-Digital Converter (ADC) Threshold Compare Interrupt Enable bit
1 = Enables the ADC threshold compare interrupt
0 = Disables the ADC threshold compare interrupt
- bit 0 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

REGISTER 10-11: PIR0: PERIPHERAL INTERRUPT STATUS REGISTER 0

U-0	U-0	R/W/HS-0/0	R-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IF	IOCIF	—	—	—	INTF ⁽¹⁾
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS= Hardware Set

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **TMR0IF:** Timer0 Overflow Interrupt Flag bit

- 1 = Timer0 register has overflowed (must be cleared in software)
- 0 = Timer0 register did not overflow

bit 4 **IOCIF:** Interrupt-on-Change Interrupt Flag bit (read-only)⁽²⁾

- 1 = One or more of the IOCAF-IOCEF register bits are currently set, indicating an enabled edge was detected by the IOC module.
- 0 = None of the IOCAF-IOCEF register bits are currently set

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **INTF:** INT External Interrupt Flag bit⁽¹⁾

- 1 = The INT external interrupt occurred (must be cleared in software)
- 0 = The INT external interrupt did not occur

Note 1: The External Interrupt GPIO pin is selected by INTPPS (Register 15-1).

2: The IOCIF bit is the logical OR of all the IOCAF-IOCEF flags. Therefore, to clear the IOCIF flag, application firmware must clear all of the lower level IOCAF-IOCEF register bits.

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

REGISTER 14-26: LATD: PORTD DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **LATD<7:0>**: RD<7:0> Output Latch Value bits⁽¹⁾

Note 1: Writes to PORTD are actually written to corresponding LATD register. Reads from PORTD register is return of actual I/O pin values.

REGISTER 14-27: ANSD: PORTD ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **ANSD<7:0>**: Analog Select between Analog or Digital Function on pins RD<7:0>, respectively
 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.
 0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

PIC16(L)F19195/6/7

REGISTER 14-42: LATF: PORTF DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set '0' = Bit is cleared

bit 7-0 **LATF<7:0>**: RF<7:0> Output Latch Value bits⁽¹⁾

Note 1: Writes to PORTF are actually written to corresponding LATF register. Reads from PORTF register is return of actual I/O pin values.

REGISTER 14-43: ANSELF: PORTF ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSF7	ANSF6	ANSF5	ANSF4	ANSF3	ANSF2	ANSF1	ANSF0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set '0' = Bit is cleared

bit 7-0 **ANSF<7:0>**: Analog Select between Analog or Digital Function on pins RF<7:0>, respectively
1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

REGISTER 14-44: WPUF: WEAK PULL-UP PORTF REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WPUF7	WPUF6	WPUF5	WPUF4	WPUF3	WPUF2	WPUF1	WPUF0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set '0' = Bit is cleared

bit 7-0 **WPUF<7:0>**: Weak Pull-up Register bits⁽¹⁾
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: The weak pull-up device is automatically disabled if the pin is configured as an output.

14.14.6 ANALOG CONTROL

The ANSELG register (Register 14-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELG bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELG bits has no effect on digital output functions. A pin with its TRIS bit clear and its ANSEL bit set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

<p>Note: The ANSELG bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software.</p>
--

14.14.7 WEAK PULL-UP CONTROL

The WPUG register (Register 14-5) controls the individual weak pull-ups for each PORT pin.

14.14.8 PORTG FUNCTIONS AND OUTPUT PRIORITIES

Each PORTF pin is multiplexed with other functions.

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic or by enabling an analog output, such as the DAC. See **Section 15.0 “Peripheral Pin Select (PPS) Module”** for more information.

Analog input functions, such as ADC and comparator inputs are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

REGISTER 16-1: PMD0: PMD CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0
SYSCMD	FVRMD	ACTMD	—	—	NVMMD	—	IOCMD
7							0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

- bit 7 **SYSCMD:** Disable Peripheral System Clock Network bit
See description in **Section 16.4 “System Clock Disable”**.
1 = System clock network disabled (a.k.a. FOSC)
0 = System clock network enabled
- bit 6 **FVRMD:** Disable Fixed Voltage Reference (FVR) bit
1 = FVR module disabled
0 = FVR module enabled
- bit 5 **ACTMD:** Disable Active Clock Tuning (ACT) bit
1 = ACT module disabled
0 = ACT module enabled
- bit 4-3 **Unimplemented:** Read as '0'
- bit 2 **NVMMD:** NVM Module Disable bit⁽¹⁾
1 = User memory reading and writing is disabled; NVMCON registers cannot be written; FSR access to these locations returns zero.
0 = NVM module enabled
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **IOCMD:** Disable Interrupt-on-Change bit, All Ports
1 = IOC module(s) disabled
0 = IOC module(s) enabled

Note 1: When enabling NVM, a delay of up to 1 μ s may be required before accessing data.

REGISTER 19-22: ADPREVH: ADC PREVIOUS RESULT REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
PREV<15:8>							
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **PREV<15:8>**: Previous ADC Results bits
If ADPSIS = 1:
Upper byte of FLTR at the start of current ADC conversion
If ADPSIS = 0:
Upper bits of ADRES at the start of current ADC conversion⁽¹⁾

Note 1: If ADPSIS = 0, ADPREVH and ADPREVL are formatted the same way as ADRES is, depending on the FM bit.

REGISTER 19-23: ADPREVL: ADC PREVIOUS RESULT REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
PREV<7:0>							
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **PREV<7:0>**: Previous ADC Results bits
If ADPSIS = 1:
Lower byte of FLTR at the start of current ADC conversion
If ADPSIS = 0:
Lower bits of ADRES at the start of current ADC conversion⁽¹⁾

Note 1: If ADPSIS = 0, ADPREVH and ADPREVL are formatted the same way as ADRES is, depending on the FM bit.

24.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

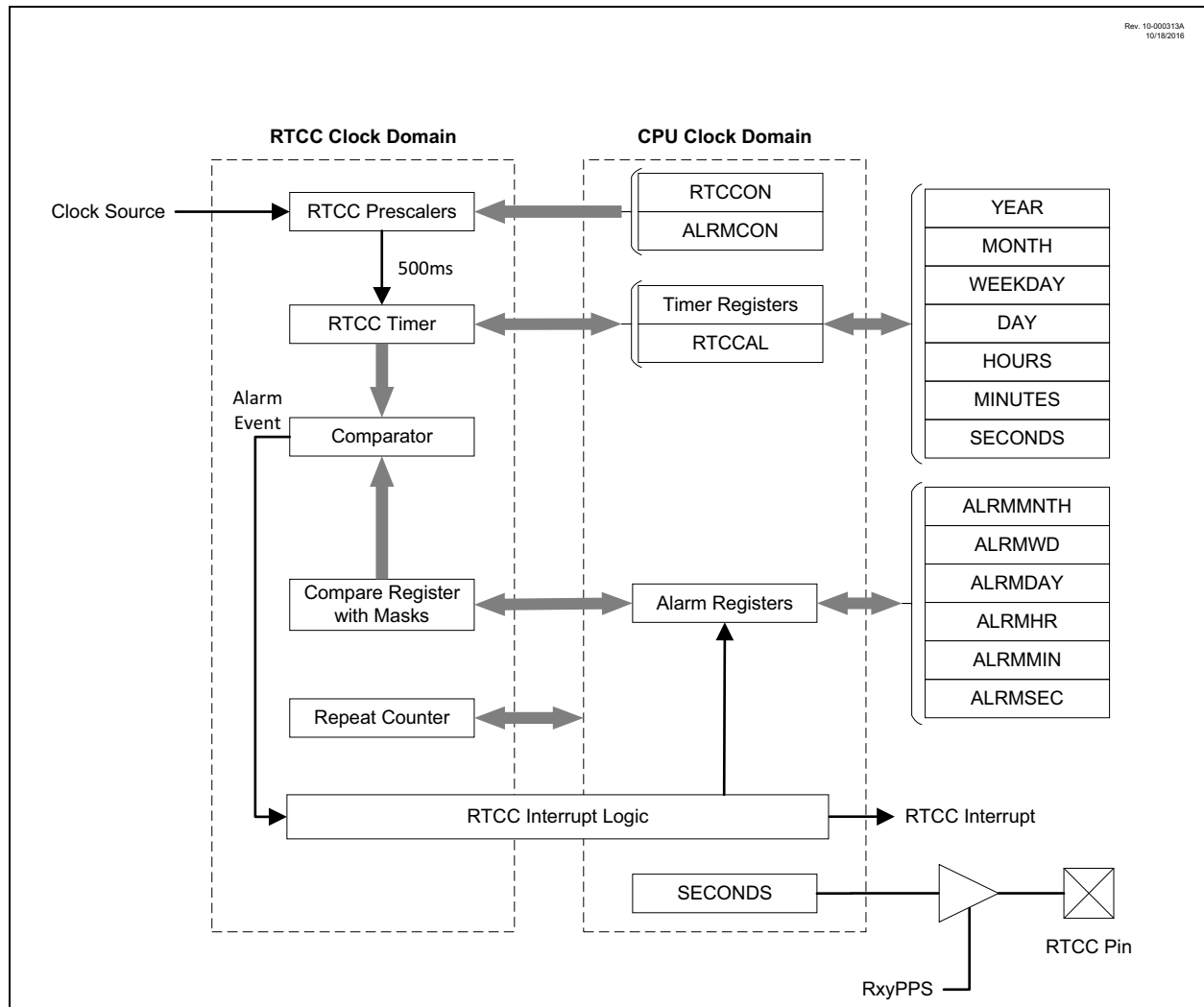
The PIC16(L)F19195/6/7 family of devices is equipped with a Real-Time Clock and Calendar (RTCC) module, designed to maintain accurate time measurement for extended periods, with little or no intervention from the CPU. The module is optimized for low-power operation in order to provide extended battery life. The key features include:

- Time: Hours, Minutes and Seconds
- 24-hour Format (Military Time)

- Calendar: Weekday, Date, Month and Year
- Year Range: 2000 to 2099
- Leap Year Correction
- Configurable Alarm
- BCD Format for Compact Firmware
- Half-second Synchronization and Visibility
- User Calibration with Auto-Adjust
- Multiple Clock Sources
- Low-Power Optimization

Figure 24-1 is a simplified block diagram of the RTCC module.

FIGURE 24-1: RTCC BLOCK DIAGRAM



REGISTER 24-8: MINUTES⁽¹⁾: MINUTE VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINH<2:0>			MINL<3:0>			
bit 7							bit 0

Legend:

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

bit 7 **Unimplemented:** Read as '0'
 bit 6-4 **MINH<2:0>:** Binary Coded Decimal value of minutes '10' digit; valid values from 0 to 5
 bit 3-0 **MINL<3:0>:** Binary Coded Decimal value of minutes '1' digit; valid values from 0 to 9

Note 1: Writes to the MINUTE registers are only allowed when RTCWREN = 1.

REGISTER 24-9: SECONDS⁽¹⁾: SECOND VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECH<2:0>			SECL<3:0>			
bit 7							bit 0

Legend:

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

bit 7 **Unimplemented:** Read as '0'
 bit 6-4 **SECH<2:0>:** Binary Coded Decimal value of seconds '10' digit; valid values from 0 to 5
 bit 3-0 **SECL<3:0>:** Binary Coded Decimal value of seconds '1' digit; valid values from 0 to 9

Note 1: Writes to the SECOND registers are only allowed when RTCWREN = 1.

REGISTER 25-1: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7	T0EN: Timer0 Enable bit 1 = The module is enabled and operating 0 = The module is disabled and in the lowest power mode
bit 6	Unimplemented: Read as '0'
bit 5	T0OUT: Timer0 Output bit (read-only) Timer0 output bit
bit 4	T016BIT: Timer0 Operating as 16-bit Timer Select bit 1 = Timer0 is a 16-bit timer 0 = Timer0 is an 8-bit timer
bit 3-0	T0OUTPS<3:0>: Timer0 output postscaler (divider) select bits 1111 = 1:16 Postscaler 1110 = 1:15 Postscaler 1101 = 1:14 Postscaler 1100 = 1:13 Postscaler 1011 = 1:12 Postscaler 1010 = 1:11 Postscaler 1001 = 1:10 Postscaler 1000 = 1:9 Postscaler 0111 = 1:8 Postscaler 0110 = 1:7 Postscaler 0101 = 1:6 Postscaler 0100 = 1:5 Postscaler 0011 = 1:4 Postscaler 0010 = 1:3 Postscaler 0001 = 1:2 Postscaler 0000 = 1:1 Postscaler

27.5 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except $F_{osc}/4$ and show clock-sync delays of at least two full cycles for both ON and Timer2/4_ers. When using $F_{osc}/4$, the clock-sync delay is at least one instruction period for Timer2/4_ers; ON applies in the next instruction period.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section 29.0 “Capture/Compare/PWM Modules”**. The signals are not a part of the Timer2/4 module.

27.5.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2/4 operation. The timer increments with each clock input when $ON = 1$ and does not increment when $ON = 0$. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 27-4. With $PRx = 5$, the counter advances until $TMRx = 5$, and goes to zero with the next clock.

29.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 26.0 “Timer1 Module with Gate Control”** for more information on configuring Timer1.

29.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIE6 register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIR6 register following any change in Operating mode.

Note: Clocking Timer1 from the system clock (Fosc) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock (Fosc/4) or from an external clock source.

29.1.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxMODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler. Example 29-1 demonstrates the code to perform this function.

EXAMPLE 29-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCPxCON    ;Set Bank bits to point
                   ;to CCPxCON
CLRf    CCPxCON    ;Turn CCP module off
MOVLW  NEW_CAPT_PS ;Load the W reg with
                   ;the new prescaler
                   ;move value and CCP ON
MOVWF   CCPxCON    ;Load CCPxCON with this
                   ;value
```

29.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (Fosc/4), or by an external clock source.

When Timer1 is clocked by Fosc/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep when Timer1 is clocked by an external clock source.

29.2 Compare Mode

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMR1H:TMR1L register pair. When a match occurs, one of the following events can occur:

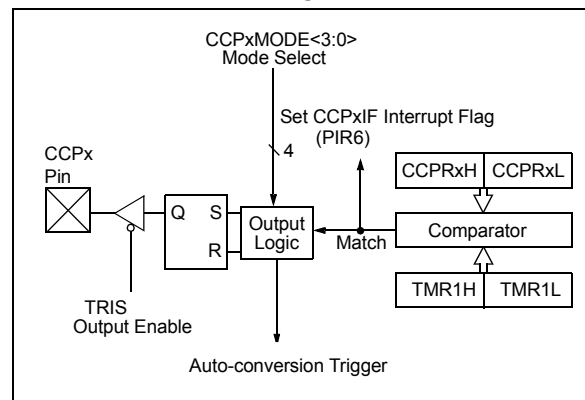
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate an Auto-conversion Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxMODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set, and an ADC conversion can be triggered, if selected.

All Compare modes can generate an interrupt and trigger and ADC conversion.

Figure 29-2 shows a simplified diagram of the compare operation.

FIGURE 29-2: COMPARE MODE OPERATION BLOCK DIAGRAM



31.5 Dead-Band Control

The dead-band control provides non-overlapping PWM signals to prevent shoot-through current in PWM switches. Dead-band operation is employed for Half-Bridge and Full-Bridge modes. The CWG contains two 6-bit dead-band counters. One is used for the rising edge of the input source control in Half-Bridge mode or for reverse dead-band Full-Bridge mode. The other is used for the falling edge of the input source control in Half-Bridge mode or for forward dead band in Full-Bridge mode.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers. See CWG1DBR and CWG1DBF registers, respectively.

31.5.1 DEAD-BAND FUNCTIONALITY IN HALF-BRIDGE MODE

In Half-Bridge mode, the dead-band counters dictate the delay between the falling edge of the normal output and the rising edge of the inverted output. This can be seen in Figure 31-9.

31.5.2 DEAD-BAND FUNCTIONALITY IN FULL-BRIDGE MODE

In Full-Bridge mode, the dead-band counters are used when undergoing a direction change. The MODE<0> bit of the CWG1CON0 register can be set or cleared while the CWG is running, allowing for changes from Forward to Reverse mode. The CWG1A and CWG1C signals will change upon the first rising input edge following a direction change, but the modulated signals (CWG1B or CWG1D, depending on the direction of the change) will experience a delay dictated by the dead-band counters. This is demonstrated in Figure 31-3.

31.6 Rising Edge and Reverse Dead Band

CWG1DBR controls the rising edge dead-band time at the leading edge of CWG1A (Half-Bridge mode) or the leading edge of CWG1B (Full-Bridge mode). The CWG1DBR value is double-buffered. When EN = 0, the CWG1DBR register is loaded immediately when CWG1DBR is written. When EN = 1, then software must set the LD bit of the CWG1CON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

31.7 Falling Edge and Forward Dead Band

CWG1DBF controls the dead-band time at the leading edge of CWG1B (Half-Bridge mode) or the leading edge of CWG1D (Full-Bridge mode). The CWG1DBF value is double-buffered. When EN = 0, the CWG1DBF register is loaded immediately when CWG1DBF is written. When EN = 1 then software must set the LD bit of the CWG1CON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

Refer to Figure 31-6 and Figure 31-7 for examples.

33.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 33-1: I²C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPxADD.
Write Request	Slave receives a matching address with R/W bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/W bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

33.4.5 START CONDITION

The I²C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 33-12 shows wave forms for Start and Stop conditions.

33.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

33.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 33-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/W bit set. The slave logic will then hold the clock and prepare to clock out data.

33.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

33.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 33-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 33-39).

FIGURE 33-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)

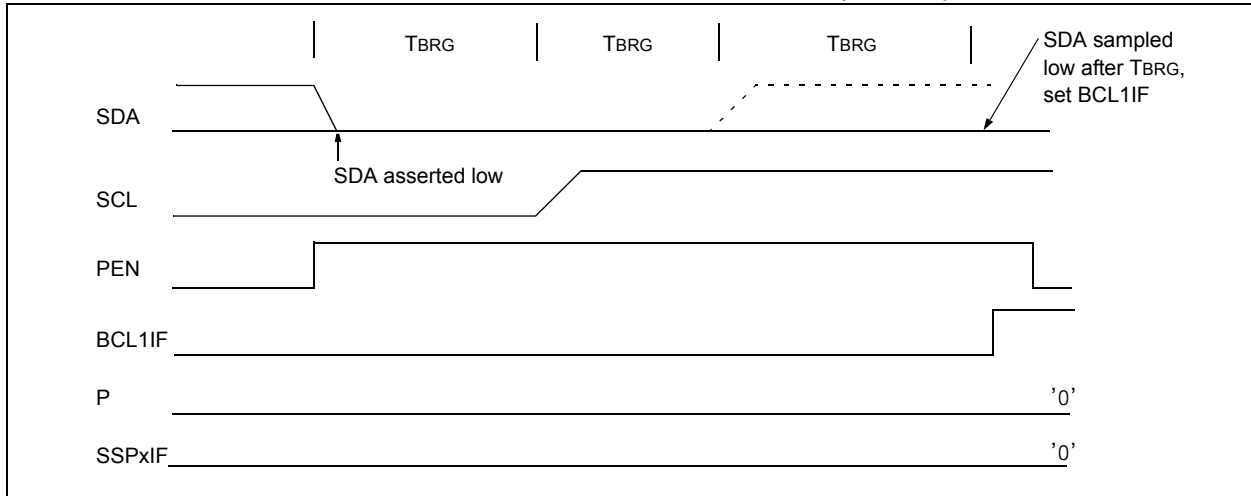
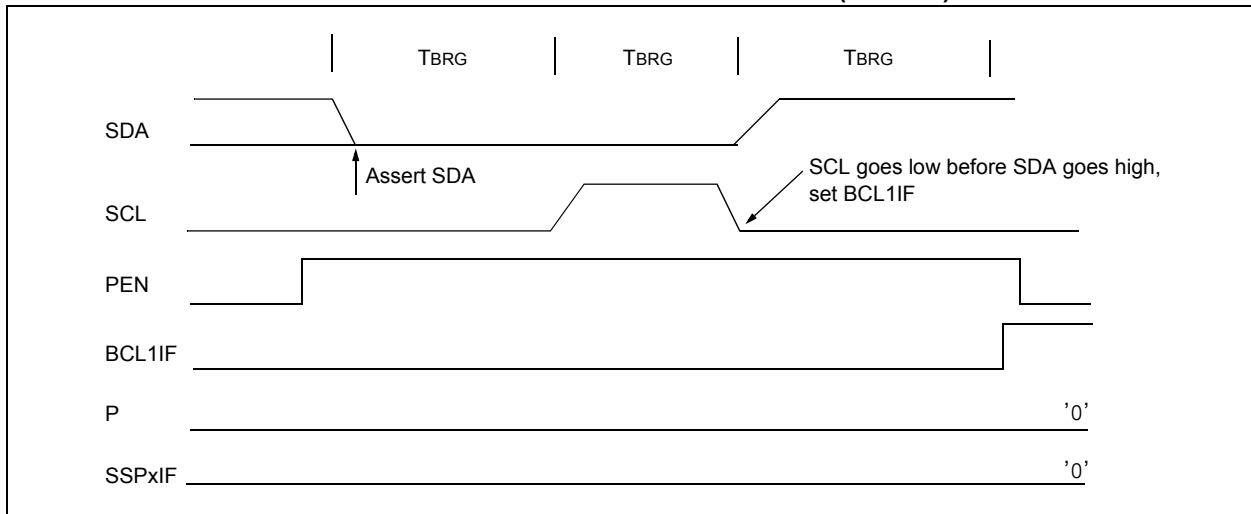


FIGURE 33-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



34.4.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXxSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXxSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCxSTA register enables the EUSART.

34.4.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 34.4.1.3 “Synchronous Master Transmission”**), except in the case of the Sleep mode.

If two words are written to the TXxREG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in the TXxREG register.
3. The TXxIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXxREG register will transfer the second character to the TSR and the TXxIF bit will now be set.
5. If the PEIE and TXxIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

34.4.2.2 Synchronous Slave Transmission Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for the CK pin (if applicable).
3. Clear the CREN and SREN bits.
4. If interrupts are desired, set the TXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit transmission is desired, set the TX9 bit.
6. Enable transmission by setting the TXEN bit.
7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
8. Start transmission by writing the Least Significant eight bits to the TXxREG register.

34.6 Register Definitions: EUSART Control

REGISTER 34-1: TXxSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7	<p>CSRC: Clock Source Select bit</p> <p><u>Asynchronous mode:</u> Unused in this mode – value ignored</p> <p><u>Synchronous mode:</u> 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)</p>
bit 6	<p>TX9: 9-bit Transmit Enable bit</p> <p>1 = Selects 9-bit transmission 0 = Selects 8-bit transmission</p>
bit 5	<p>TXEN: Transmit Enable bit⁽¹⁾</p> <p>1 = Transmit enabled 0 = Transmit disabled</p>
bit 4	<p>SYNC: EUSART Mode Select bit</p> <p>1 = Synchronous mode 0 = Asynchronous mode</p>
bit 3	<p>SENDB: Send Break Character bit</p> <p><u>Asynchronous mode:</u> 1 = Send SYNCH BREAK on next transmission – Start bit, followed by 12 '0' bits, followed by Stop bit; cleared by hardware upon completion 0 = SYNCH BREAK transmission disabled or completed</p> <p><u>Synchronous mode:</u> Unused in this mode – value ignored</p>
bit 2	<p>BRGH: High Baud Rate Select bit</p> <p><u>Asynchronous mode:</u> 1 = High speed 0 = Low speed</p> <p><u>Synchronous mode:</u> Unused in this mode – value ignored</p>
bit 1	<p>TRMT: Transmit Shift Register Status bit</p> <p>1 = TSR empty 0 = TSR full</p>
bit 0	<p>TX9D: Ninth bit of Transmit Data</p> <p>Can be address/data bit or a parity bit.</p>

Note 1: SREN/CREN overrides TXEN in Sync mode.

TABLE 39-24: SPI MODE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Characteristic	Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2scH, TssL2scL	\overline{SS} ↓ to SCK↓ or SCK↑ input	2.25*Tcy	—	—	ns	
SP71*	Tsch	SCK input high time (Slave mode)	Tcy + 20	—	—	ns	
SP72*	TscL	SCK input low time (Slave mode)	Tcy + 20	—	—	ns	
SP73*	TdIV2scH, TdIV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
SP74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
SP75*	TdoR	SDO data output rise time	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP76*	TdoF	SDO data output fall time	—	10	25	ns	
SP77*	TssH2boZ	\overline{SS} ↑ to SDO output high-impedance	10	—	50	ns	
SP78*	Tscr	SCK output rise time (Master mode)	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP79*	TscF	SCK output fall time (Master mode)	—	10	25	ns	
SP80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	3.0V ≤ VDD ≤ 5.5V
			—	—	145	ns	1.8V ≤ VDD ≤ 5.5V
SP81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	1 Tcy	—	—	ns	
SP82*	TssL2doV	SDO data output valid after \overline{SS} ↓ edge	—	—	50	ns	
SP83*	Tsch2ssH, TscL2ssH	\overline{SS} ↑ after SCK edge	1.5 Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

PRELIMINARY