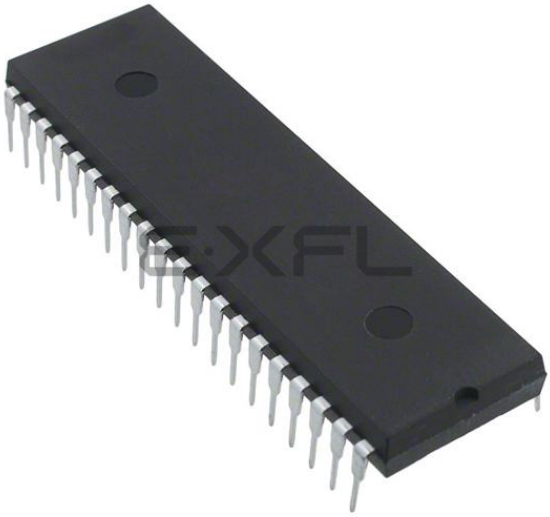


Welcome to E-XFL.COM

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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4220-i-p

TABLE 1-3: PIC18F4220/4320 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
MCLR/VPP/RE3	1	18	18			Master Clear (input) or programming voltage (input).
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P		Programming voltage input.
RE3				I	ST	Digital input.
OSC1/CLKI/RA7	13	30	32			Oscillator crystal or external clock input.
OSC1				I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise.
CLKI				I	CMOS	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7				I/O	TTL	General purpose I/O pin.
OSC2/CLKO/RA6	14	31	33			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6				I/O	TTL	General purpose I/O pin.
						PORTA is a bidirectional I/O port.
RA0/AN0	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
CVREF				O	Analog	Comparator reference voltage output.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT	6	23	23			
RA4				I/O	ST/OD	Digital I/O. Open drain when configured as output.
T0CKI				I	ST	Timer0 external clock input.
C1OUT				O	—	Comparator 1 output.
RA5/AN4/SS/LVDIN/C2OUT	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
SS				I	TTL	SPI slave select input.
LVDIN				I	Analog	Low-Voltage Detect input.
C2OUT				O	—	Comparator 2 output.
RA6						See the OSC2/CLKO/RA6 pin.
RA7						See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power; OD = Open-drain (no diode to VDD)

Note 1: Alternate assignment for CCP2 when CCP2MX is cleared.

2: Default assignment for CCP2 when CCP2MX (CONFIG3H<0>) is set.

TABLE 1-3: PIC18F4220/4320 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTE is a bidirectional I/O port.
RE0/AN5/$\overline{RD}$	8	25	25			
RE0				I/O	ST	Digital I/O.
AN5				I	Analog	Analog input 5.
$\overline{RD}$				I	TTL	Read control for Parallel Slave Port (see also $\overline{WR}$ and $\overline{CS}$ pins).
RE1/AN6/$\overline{WR}$	9	26	26			
RE1				I/O	ST	Digital I/O.
AN6				I	Analog	Analog input 6.
$\overline{WR}$				I	TTL	Write control for Parallel Slave Port (see $\overline{CS}$ and $\overline{RD}$ pins).
RE2/AN7/$\overline{CS}$	10	27	27			
RE2				I/O	ST	Digital I/O.
AN7				I	Analog	Analog input 7.
$\overline{CS}$				I	TTL	Chip select control for Parallel Slave Port (see related $\overline{RD}$ and $\overline{WR}$).
RE3	1	18	18	—	—	See $\overline{MCLR}$/VPP/RE3 pin.
VSS	12, 31	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	7, 28	7, 8 29	P	—	Positive supply for logic and I/O pins.
NC	—	—	13, 28	NC	NC	No connect.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power; OD = Open-drain (no diode to VDD)

Note 1: Alternate assignment for CCP2 when CCP2MX is cleared.

2: Default assignment for CCP2 when CCP2MX (CONFIG3H<0>) is set.

3.5.2 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock (defined in Configuration Register 1H) becomes ready. At that time, the OSTS bit is set and the device begins executing code.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 23.3 "Two-Speed Start-up"**) or Fail-Safe Clock Monitor (see **Section 23.4 "Fail-Safe Clock Monitor"**) are enabled in Configuration Register 1H, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Since the OSCCON register is cleared following all Resets, the INTRC clock source is selected. A higher speed clock may be selected by modifying the IRCF bits in the OSCCON register. Execution is clocked by the internal oscillator block until either the primary clock becomes ready, or a power-managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

3.5.3 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in a wake-up from the power-managed mode (see **Section 3.2 "Sleep Mode"** through **Section 3.4 "Run Modes"**).

If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 23.2 "Watchdog Timer (WDT)"**).

The WDT timer and postscaler are cleared by executing a `SLEEP` or `CLRWDT` instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the system clock source.

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. These are:

- PRI_IDLE mode, where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these cases, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes).

However, a fixed delay (approximately 10 μs) following the wake-up event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.6 INTOSC Frequency Drift

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz. However, this frequency may drift as V_{DD} or temperature changes, which can affect the controller operation in a variety of ways.

It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. This has the side effect that the INTRC clock source frequency is also affected. However, the features that use the INTRC source often do not require an exact frequency. These features include the Fail-Safe Clock Monitor, the Watchdog Timer and the RC_RUN/RC_IDLE modes when the INTRC clock source is selected.

Being able to adjust the INTOSC requires knowing when an adjustment is required, in which direction it should be made and in some cases, how large a change is needed. Three examples are shown but other techniques may be used.

3.6.1 EXAMPLE – USART

An adjustment may be indicated when the USART begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the system clock frequency is too high – try decrementing the value in the OSCTUNE register to reduce the system clock frequency. Errors in data may suggest that the system clock speed is too low – increment OSCTUNE.

3.6.2 EXAMPLE – TIMERS

This technique compares system clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator.

Both timers are cleared but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is greater than expected, then the internal oscillator block is running too fast – decrement OSCTUNE.

3.6.3 EXAMPLE – CCP IN CAPTURE MODE

A CCP module can use free-running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast – decrement OSCTUNE. If the measured time is much less than the calculated time, the internal oscillator block is running too slow – increment OSCTUNE.

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TOSU	2220	2320	4220	4320	---0 0000	---0 0000	---0 uuuu(3)
TOSH	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu(3)
TOSL	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu(3)
STKPTR	2220	2320	4220	4320	uu-0 0000	00-0 0000	uu-u uuuu(3)
PCLATU	2220	2320	4220	4320	---0 0000	---0 0000	---u uuuu
PCLATH	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
PCL	2220	2320	4220	4320	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	2220	2320	4220	4320	--00 0000	--00 0000	--uu uuuu
TBLPTRH	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
TABLAT	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
PRODH	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	2220	2320	4220	4320	0000 000x	0000 000u	uuuu uuuu(1)
INTCON2	2220	2320	4220	4320	1111 -1-1	1111 -1-1	uuuu -u-u(1)
INTCON3	2220	2320	4220	4320	11-0 0-00	11-0 0-00	uu-u u-uu(1)
INDF0	2220	2320	4220	4320	N/A	N/A	N/A
POSTINC0	2220	2320	4220	4320	N/A	N/A	N/A
POSTDEC0	2220	2320	4220	4320	N/A	N/A	N/A
PREINC0	2220	2320	4220	4320	N/A	N/A	N/A
PLUSW0	2220	2320	4220	4320	N/A	N/A	N/A
FSR0H	2220	2320	4220	4320	---- xxxx	---- uuuu	---- uuuu
FSR0L	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	2220	2320	4220	4320	N/A	N/A	N/A
POSTINC1	2220	2320	4220	4320	N/A	N/A	N/A
POSTDEC1	2220	2320	4220	4320	N/A	N/A	N/A
PREINC1	2220	2320	4220	4320	N/A	N/A	N/A
PLUSW1	2220	2320	4220	4320	N/A	N/A	N/A

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 4-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
FSR1H	2220	2320	4220	4320	---- xxxx	---- uuuu	---- uuuu
FSR1L	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	2220	2320	4220	4320	---- 0000	---- 0000	---- uuuu
INDF2	2220	2320	4220	4320	N/A	N/A	N/A
POSTINC2	2220	2320	4220	4320	N/A	N/A	N/A
POSTDEC2	2220	2320	4220	4320	N/A	N/A	N/A
PREINC2	2220	2320	4220	4320	N/A	N/A	N/A
PLUSW2	2220	2320	4220	4320	N/A	N/A	N/A
FSR2H	2220	2320	4220	4320	---- xxxx	---- uuuu	---- uuuu
FSR2L	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	2220	2320	4220	4320	---x xxxx	---u uuuu	---u uuuu
TMR0H	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
TMR0L	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	2220	2320	4220	4320	1111 1111	1111 1111	uuuu uuuu
OSCCON	2220	2320	4220	4320	0000 q000	0000 q000	uuuu qquu
LVDCON	2220	2320	4220	4320	--00 0101	--00 0101	--uu uuuu
WDTCON	2220	2320	4220	4320	---- ---0	---- ---0	---- ---u
RCON(4)	2220	2320	4220	4320	0--1 11q0	0--q qquu	u--u qquu
TMR1H	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	2220	2320	4220	4320	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
PR2	2220	2320	4220	4320	1111 1111	1111 1111	1111 1111
T2CON	2220	2320	4220	4320	-000 0000	-000 0000	-uuu uuuu
SSPBUF	2220	2320	4220	4320	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
SSPCON1	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu
SSPCON2	2220	2320	4220	4320	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 4-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2220/2320/4220/4320) (CONTINUED)

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:
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	0000 q000	27, 47
LVDCON	—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	47, 233
WDTCON	—	—	—	—	—	—	—	SWDTEN	--- ---0	47, 247
RCON	IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0--1 11q0	45, 69, 98
TMR1H	Timer1 Register High Byte								xxxx xxxx	47, 125
TMR1L	Timer1 Register Low Byte								xxxx xxxx	47, 125
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	47, 121
TMR2	Timer2 Register								0000 0000	47, 127
PR2	Timer2 Period Register								1111 1111	47, 127
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	47, 127
SSPBUF	MSSP Receive Buffer/Transmit Register								xxxx xxxx	47, 156, 164
SSPADD	MSSP Address Register in I^2C™ Slave mode. MSSP Baud Rate Reload Register in I^2C Master mode.								0000 0000	47, 164
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	47, 156, 165
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	47, 157, 166
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	47, 167
ADRESH	A/D Result Register High Byte								xxxx xxxx	48, 220
ADRESL	A/D Result Register Low Byte								xxxx xxxx	48, 220
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	48, 211
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0qqq	48, 212
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	48, 213
CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	48, 134
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	48, 134
CCP1CON	P1M1(5)	P1M0(5)	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	48, 133, 141
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	48, 134
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	48, 134
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	48, 133
PWM1CON(5)	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	48, 149
ECCPAS(5)	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	48, 150
CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	48, 227
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	48, 221
TMR3H	Timer3 Register High Byte								xxxx xxxx	48, 131
TMR3L	Timer3 Register Low Byte								xxxx xxxx	48, 131
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	0000 0000	48, 129
SPBRG	USART Baud Rate Generator								0000 0000	48, 198
RCREG	USART Receive Register								0000 0000	48, 204, 203
TXREG	USART Transmit Register								0000 0000	48, 202, 203
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	48, 196
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	48, 197

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

Note 1: RA6 and associated bits are configured as port pins in RCIO, ECIO and INTIO2 (with port function on RA6) Oscillator mode only and read as '0' in all other oscillator modes.

2: RA7 and associated bits are configured as port pins in INTIO2 Oscillator mode only and read as '0' in all other modes.

3: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

4: If PBADEN = 0, PORTB<4:0> are configured as digital inputs and read unknown and if PBADEN = 1, PORTB<4:0> are configured as analog inputs and read as '0' following a Reset.

5: These registers and/or bits are not implemented on the PIC18F2X20 devices and read as 0x00.

6: The RE3 port bit is available as an input only pin only in 40-pin devices when Master Clear functionality is disabled (CONFIG3H<7> = 0).

5.14 RCON Register

The Reset Control (RCON) register contains flag bits that allow differentiation between the sources of a device Reset. These flags include the $\overline{TO}$, $\overline{PD}$, $\overline{POR}$, $\overline{BOR}$ and $\overline{RI}$ bits. This register is readable and writable.

Note 1: If the BOREN Configuration bit is set (Brown-out Reset enabled), the $\overline{BOR}$ bit is '1' on a Power-on Reset. After a Brown-out Reset has occurred, the $\overline{BOR}$ bit will be cleared and must be set by firmware to indicate the occurrence of the next Brown-out Reset.

2: It is recommended that the $\overline{POR}$ bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

REGISTER 5-3: RCON: RESET CONTROL REGISTER

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
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 **IPEN:** Interrupt Priority Enable bit
- 1 = Enable priority levels on interrupts
- 0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
- 1 = The RESET instruction was not executed (set by firmware only)
- 0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
- 1 = Set by power-up, CLRWDT instruction or SLEEP instruction
- 0 = A WDT time-out occurred

bit 2 **$\overline{PD}$:** Power-down Detection Flag bit
- 1 = Set by power-up or by the CLRWDT instruction
- 0 = Cleared by execution of the SLEEP instruction

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
- 1 = A Power-on Reset has not occurred (set by firmware only)
- 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
- 1 = A Brown-out Reset has not occurred (set by firmware only)
- 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

REGISTER 7-1: EECON1: DATA EEPROM CONTROL REGISTER 1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Legend:	S = Settable bit		
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 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
1 = Access Flash program memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Access Configuration or Calibration registers
0 = Access Flash program or data EEPROM memory

bit 5 **Unimplemented:** Read as '0'

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
0 = Perform write-only

bit 3 **WRERR:** EEPROM Error Flag bit(1)
1 = A write operation was prematurely terminated ($\overline{\text{MCLR}}$ or WDT Reset during self-timed erase or program operation)
0 = The write operation completed normally

bit 2 **WREN:** Erase/Write Enable bit
1 = Allows write cycles
0 = Inhibits write cycles

bit 1 **WR:** Write Control bit
1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle is completed

bit 0 **RD:** Read Control bit
1 = Initiates a memory read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Read completed

Note 1: When a WRERR occurs, the EEPGD or FREE bits are not cleared. This allows tracing of the error condition.

9.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag 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. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
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 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all high-priority interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

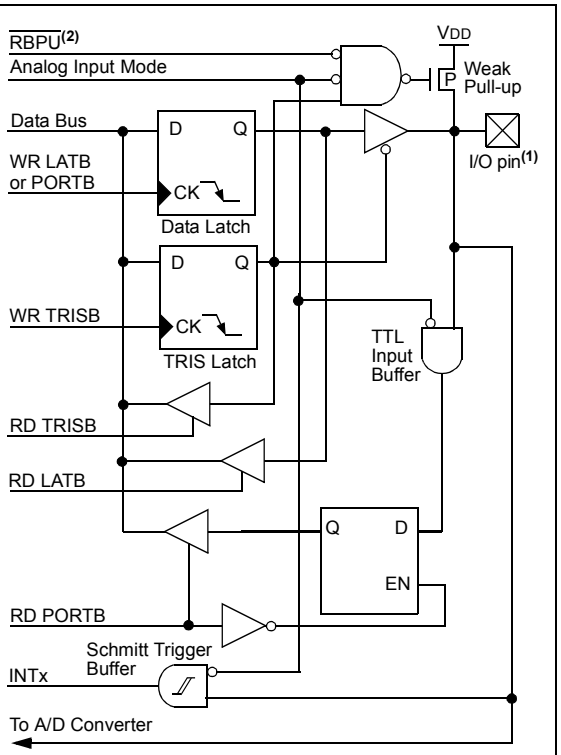
bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

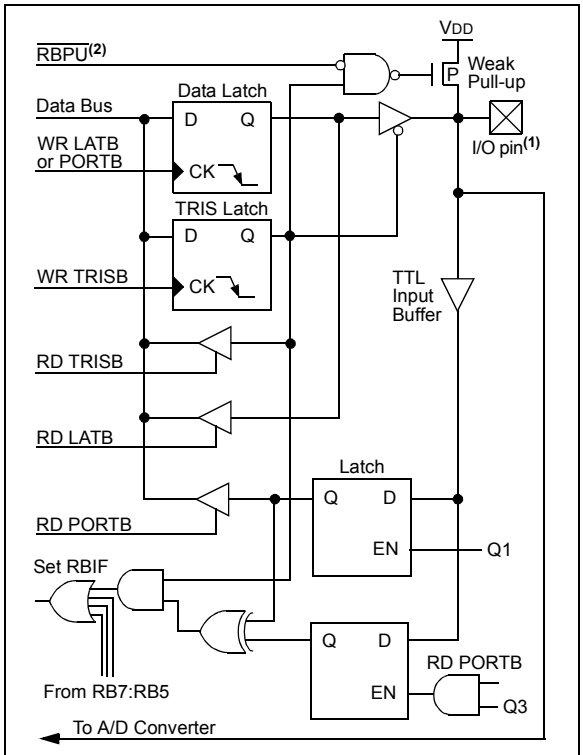
Note 1: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

FIGURE 10-7: BLOCK DIAGRAM OF RB2:RB0 PINS



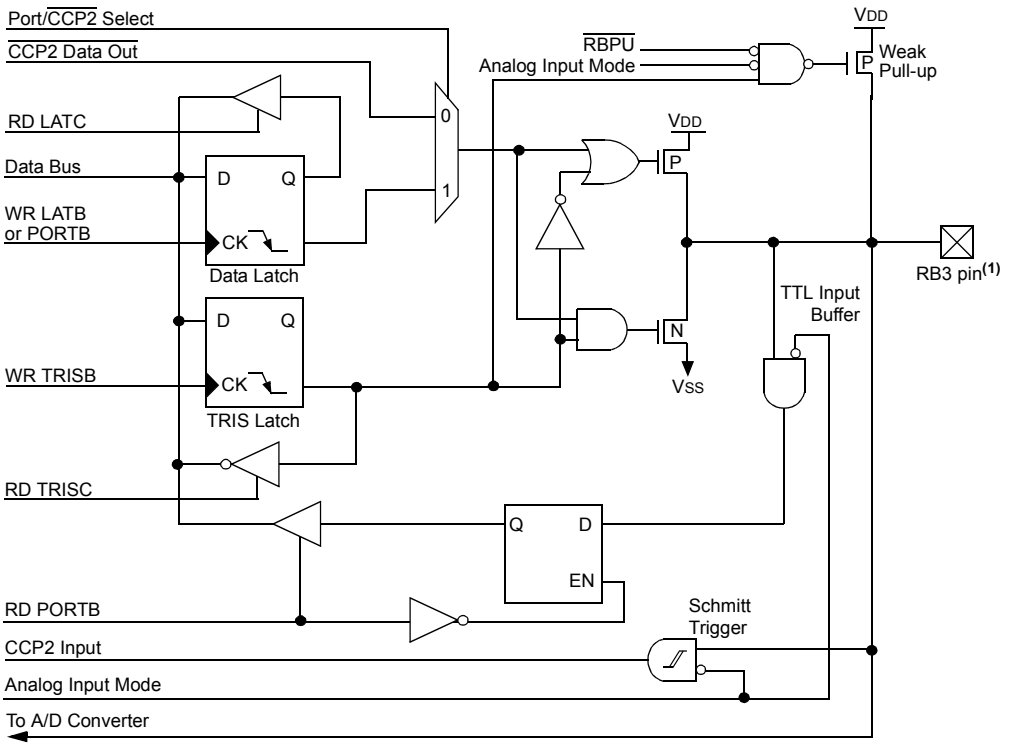
Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (INTCON2<7>).

FIGURE 10-8: BLOCK DIAGRAM OF RB4 PIN



Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (INTCON2<7>).

FIGURE 10-9: BLOCK DIAGRAM OF RB3/CCP2 PIN



Note 1: I/O pins have diode protection to VDD and VSS.

NOTES:

NOTES:

17.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.

b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

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 SSPADD<6:0> and counts down to 0. 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' (Register 17-31). 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 17-32).

FIGURE 17-31: BUS COLLISION DURING A STOP CONDITION (CASE 1)

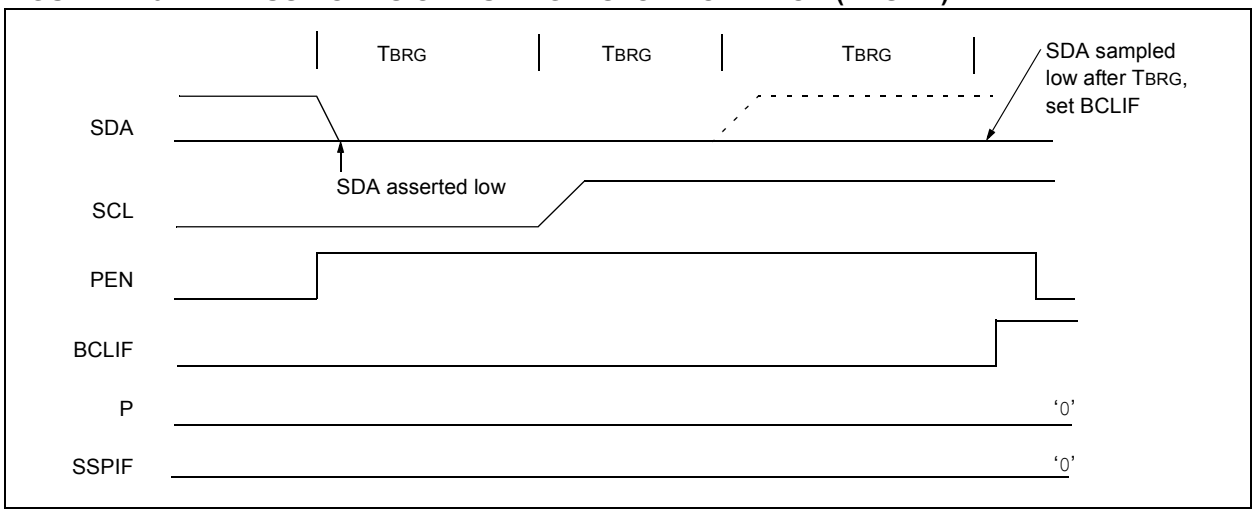
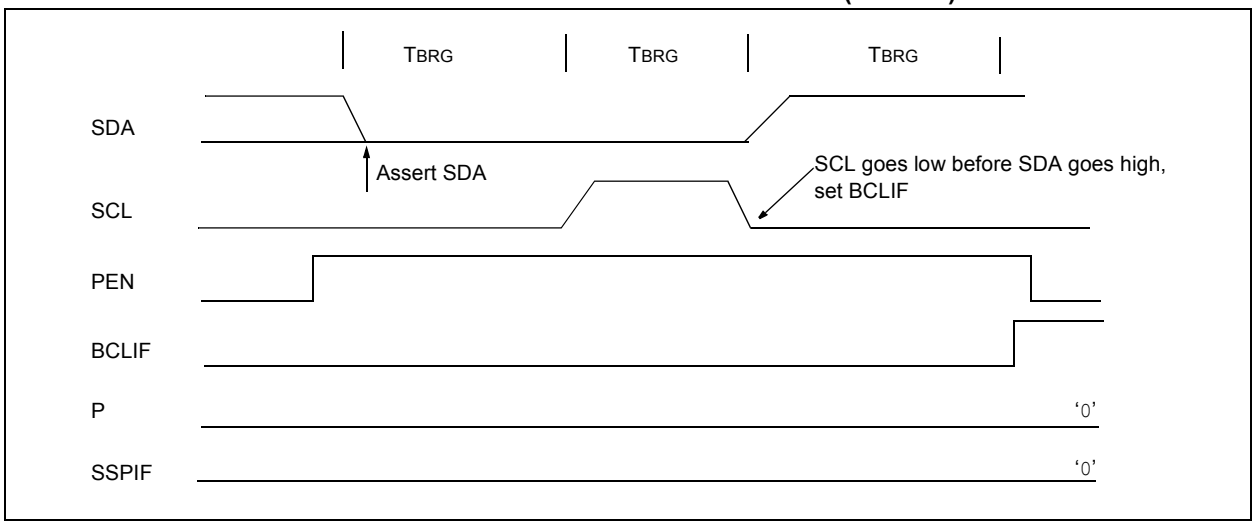


FIGURE 17-32: BUS COLLISION DURING A STOP CONDITION (CASE 2)



REGISTER 18-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-Bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receiving next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

18.3 USART Asynchronous Mode

In this mode, the USART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware but can be implemented in software (and stored as the ninth data bit). Asynchronous mode functions in all power-managed modes except Sleep mode when call clock sources are stopped. When in PRI_IDLE mode, no changes to the Baud Rate Generator values are required; however, other power-managed mode clocks may operate at another frequency than the primary clock. Therefore, the Baud Rate generator values may need adjusting.

Asynchronous mode is selected by clearing bit, SYNC (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

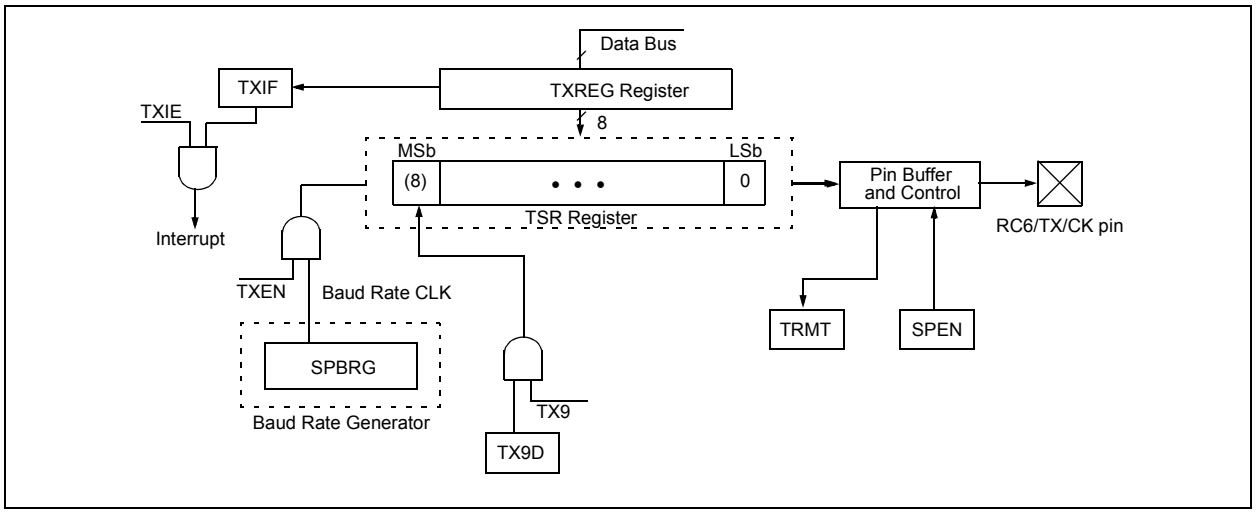
- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

18.3.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. Flag bit TXIF is not cleared immediately upon loading the Transmit Buffer register, TXREG. TXIF becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results. While flag bit TXIF indicated the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, therefore, the user must poll this bit in order to determine whether the TSR register is empty.

Note 1:	The TSR register is not mapped in data memory so it is not available to the user.
2:	Flag bit, TXIF, is set when enable bit, TXEN, is set.

FIGURE 18-1: USART TRANSMIT BLOCK DIAGRAM



18.5 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any power-managed mode. Slave mode is entered by clearing bit, CSRC (TXSTA<7>).

18.5.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode.

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

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) Flag bit, TXIF, will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit, TXIF, will now be set.
e) If enable bit, TXIE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. Clear bits, CREN and SREN.
3. If interrupts are desired, set enable bit, TXIE.
4. If 9-bit transmission is desired, set bit, TX9.
5. Enable the transmission by setting enable bit, TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 18-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
TXREG	USART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

Note 1: The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X20 devices; always maintain these bits clear.

CPFSGT **Compare f with W, Skip if f > W**

Syntax: [*label*] CPFSGT f [,a]

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	010a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction.
If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE      CPFSGT REG
NGREATER  :
GREATER   :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG > W;
PC = Address (GREATER)
If REG ≤ W;
PC = Address (NGREATER)

CPFSLT **Compare f with W, Skip if f < W**

Syntax: [*label*] CPFSLT f [,a]

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank will be selected. If 'a' is '1', the BSR will not be overridden (default).

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CPFSLT REG
NLESS   :
LESS    :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

SUBWFB **Subtract W from f with Borrow**

Syntax: [*label*] SUBWFB f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) – (W) – ($\overline{C}$) → dest

Status Affected: N, OV, C, DC, Z

Encoding:

0101	10da	ffff	ffff

Description: Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG  =  0x19    (0001 1001)
    W    =  0x0D    (0000 1101)
    C    =  0x01
After Instruction
    REG  =  0x0C    (0000 1011)
    W    =  0x0D    (0000 1101)
    C    =  0x01
    Z    =  0x00
    N    =  0x00    ; result is positive
```

Example 2: `SUBWFB REG, 0, 0`

```
Before Instruction
    REG  =  0x1B    (0001 1011)
    W    =  0x1A    (0001 1010)
    C    =  0x00
After Instruction
    REG  =  0x1B    (0001 1011)
    W    =  0x00
    C    =  0x01
    Z    =  0x01    ; result is zero
    N    =  0x00
```

Example 3: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG  =  0x03    (0000 0011)
    W    =  0x0E    (0000 1101)
    C    =  0x01
After Instruction
    REG  =  0xF5    (1111 0100)
                    ; [2's comp]
    W    =  0x0E    (0000 1101)
    C    =  0x00
    Z    =  0x00
    N    =  0x01    ; result is negative
```

TBLRD **Table Read**

Syntax: [*label*] TBLRD (*; *+; *-; +*)

Operands: None

Operation: if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT,
TBLPTR - No Change;
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT,
(TBLPTR) +1 → TBLPTR;
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT,
(TBLPTR) -1 → TBLPTR;
if TBLRD +*,
(TBLPTR) +1 → TBLPTR,
(Prog Mem (TBLPTR)) → TABLAT

Status Affected: None

Encoding:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used. The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2 Mbyte address range.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD **Table Read (cont'd)**

Example1:

```
TBLRD  *+ ;
```

Before Instruction

TABLAT	=	0x55
TBLPTR	=	0x00A356
MEMORY(0x00A356)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x00A357

Example2:

```
TBLRD  +* ;
```

Before Instruction

TABLAT	=	0xAA
TBLPTR	=	0x01A357
MEMORY(0x01A357)	=	0x12
MEMORY(0x01A358)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x01A358