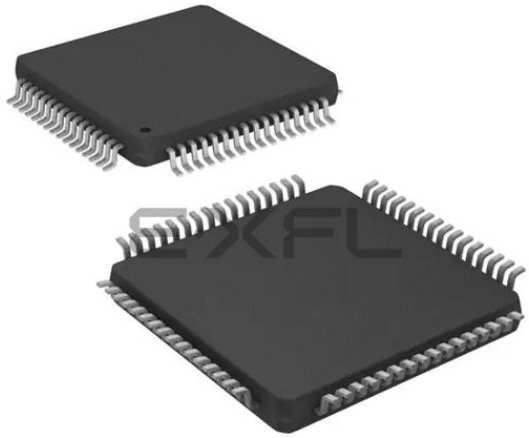


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°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/pic17c752t-16e-pt

2.0 DEVICE VARIETIES

Each device has a variety of frequency ranges and packaging options. Depending on application and production requirements, the proper device option can be selected using the information in the PIC17C7XX Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC17C7XX Product Identification System" at the back of this data sheet to specify the correct part number. When discussing the functionality of the device, memory technology and voltage range does not matter.

There are two memory type options. These are specified in the middle characters of the part number.

1. **C**, as in PIC17**C**756A. These devices have EPROM type memory.
2. **CR**, as in PIC17**CR**756A. These devices have ROM type memory.

All these devices operate over the standard voltage range. Devices are also offered which operate over an extended voltage range (and reduced frequency range). Table 2-1 shows all possible memory types and voltage range designators for a particular device. These designators are in **bold** typeface.

TABLE 2-1: DEVICE MEMORY VARIETIES

Memory Type	Voltage Range	
	Standard	**Extended**
EPROM	PIC17**C**XXX	PIC17**LC**XXX
ROM	PIC17**CR**XXX	PIC17**LCR**XXX

Note: Not all memory technologies are available for a particular device.

2.1 UV Erasable Devices

The UV erasable version, offered in CERQUAD package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP^{sm}) Devices

Microchip offers a unique programming service, where a few user defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry code, password or ID number.

2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, thus giving customers a low cost option for high volume, mature products.

ROM devices do not allow serialization information in the program memory space.

For information on submitting ROM code, please contact your regional sales office.

Note: Presently, NO ROM versions of the PIC17C7XX devices are available.

EXAMPLE 6-2: SAVING STATUS AND WREG IN RAM (NESTED)

```
; The addresses that are used to store the CPUSTA and WREG values must be in the data memory
; address range of 1Ah - 1Fh. Up to 6 locations can be saved and restored using the MOVFP
; instruction. This instruction neither affects the status bits, nor corrupts the WREG register.
; This routine uses the FRS0, so it controls the FS1 and FS0 bits in the ALUSTA register.
;
Nobank_FSR      EQU    0x40
Bank_FSR        EQU    0x41
ALU_Temp        EQU    0x42
WREG_TEMP       EQU    0x43
BSR_S1          EQU    0x01A        ; 1st location to save BSR
BSR_S2          EQU    0x01B        ; 2nd location to save BSR (Label Not used in program)
BSR_S3          EQU    0x01C        ; 3rd location to save BSR (Label Not used in program)
BSR_S4          EQU    0x01D        ; 4th location to save BSR (Label Not used in program)
BSR_S5          EQU    0x01E        ; 5th location to save BSR (Label Not used in program)
BSR_S6          EQU    0x01F        ; 6th location to save BSR (Label Not used in program)
;
INITIALIZATION                      ;
        CALL    CLEAR_RAM           ; Must Clear all Data RAM
;
INIT_POINTERS                       ; Must Initialize the pointers for POP and PUSH
        CLRF    BSR, F              ; Set All banks to 0
        CLRF    ALUSTA, F           ; FSR0 post increment
        BSF     ALUSTA, FS1
        CLRF    WREG, F             ; Clear WREG
        MOVLW   BSR_S1              ; Load FSR0 with 1st address to save BSR
        MOVWF   FSR0
        MOVWF   Nobank_FSR
        MOVLW   0x20
        MOVWF   Bank_FSR
        :
        :                           ; Your code
        :
        :                           ; At Interrupt Vector Address
PUSH    BSF     ALUSTA, FS0         ; FSR0 has auto-increment, does not affect status bits
        BCF     ALUSTA, FS1         ; does not affect status bits
        MOVFP   BSR, INDF0          ; No Status bits are affected
        CLRF    BSR, F              ; Peripheral and Data RAM Bank 0 No Status bits are affected
        MOVPF   ALUSTA, ALU_Temp    ;
        MOVPF   FSR0, Nobank_FSR    ; Save the FSR for BSR values
        MOVPF   WREG, WREG_TEMP     ;
        MOVFP   Bank_FSR, FSR0      ; Restore FSR value for other values
        MOVFP   ALU_Temp, INDF0     ; Push ALUSTA value
        MOVFP   WREG_TEMP, INDF0    ; Push WREG value
        MOVFP   PCLATH, INDF0       ; Push PCLATH value
        MOVPF   FSR0, Bank_FSR      ; Restore FSR value for other values
        MOVFP   Nobank_FSR, FSR0    ;
        ;
        :                           ; Interrupt Service Routine (ISR) code
        ;
POP     CLRF    ALUSTA, F           ; FSR0 has auto-decrement, does not affect status bits
        MOVFP   Bank_FSR, FSR0      ; Restore FSR value for other values
        DECF    FSR0, F             ;
        MOVFP   INDF0, PCLATH       ; Pop PCLATH value
        MOVFP   INDF0, WREG         ; Pop WREG value
        BSF     ALUSTA, FS1         ; FSR0 does not change
        MOVPF   INDF0, ALU_Temp     ; Pop ALUSTA value
        MOVPF   FSR0, Bank_FSR      ; Restore FSR value for other values
        DECF    Nobank_FSR, F       ;
        MOVFP   Nobank_FSR, FSR0    ; Save the FSR for BSR values
        MOVFP   ALU_Temp, ALUSTA    ;
        MOVFP   INDF0, BSR          ; No Status bits are affected
;
        RETFIE                      ; Return from interrupt (enable interrupts)
```

7.2.2.1 ALU Status Register (ALUSTA)

The ALUSTA register contains the status bits of the Arithmetic and Logic Unit and the mode control bits for the indirect addressing register.

As with all the other registers, the ALUSTA register can be the destination for any instruction. If the ALUSTA register is the destination for an instruction that affects the Z, DC, C, or OV bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the ALUSTA register as destination may be different than intended.

For example, the `CLRF ALUSTA, F` instruction will clear the upper four bits and set the Z bit. This leaves the ALUSTA register as `0000u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions be used to alter the ALUSTA register, because these instructions do not affect any status bits. To see how other instructions affect the status bits, see the "Instruction Set Summary."

Note 1: The C and DC bits operate as a $\overline{\text{borrow}}$ and digit $\overline{\text{borrow}}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

2: The overflow bit will be set if the 2's complement result exceeds +127, or is less than -128.

The Arithmetic and Logic Unit (ALU) is capable of carrying out arithmetic or logical operations on two operands, or a single operand. All single operand instructions operate either on the WREG register, or the given file register. For two operand instructions, one of the operands is the WREG register and the other is either a file register, or an 8-bit immediate constant.

REGISTER 7-1: ALUSTA REGISTER (ADDRESS: 04h, UNBANKED)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-x	R/W-x	R/W-x	R/W-x
FS3	FS2	FS1	FS0	OV	Z	DC	C
bit 7							bit 0

bit 7-6 **FS3:FS2**: FSR1 Mode Select bits
`00` = Post auto-decrement FSR1 value
`01` = Post auto-increment FSR1 value
`1x` = FSR1 value does not change

bit 5-4 **FS1:FS0**: FSR0 Mode Select bits
`00` = Post auto-decrement FSR0 value
`01` = Post auto-increment FSR0 value
`1x` = FSR0 value does not change

bit 3 **OV**: Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit7) to change state.
`1` = Overflow occurred for signed arithmetic (in this arithmetic operation)
`0` = No overflow occurred

bit 2 **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{\text{borrow}}$ bit
For `ADDWF` and `ADDLW` instructions.
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

Note: For borrow, the polarity is reversed.

bit 0 **C**: Carry/$\overline{\text{borrow}}$ bit

For `ADDWF` and `ADDLW` instructions. Note that a subtraction is executed by adding the two's complement of the second operand.
For rotate (`RRCF`, `RLCF`) instructions, this bit is loaded with either the high or low order bit of the source register.
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result

Note: For borrow, the polarity is reversed.

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

FIGURE 8-6: CONSECUTIVE `TABLWT` WRITE TIMING (EXTERNAL MEMORY)

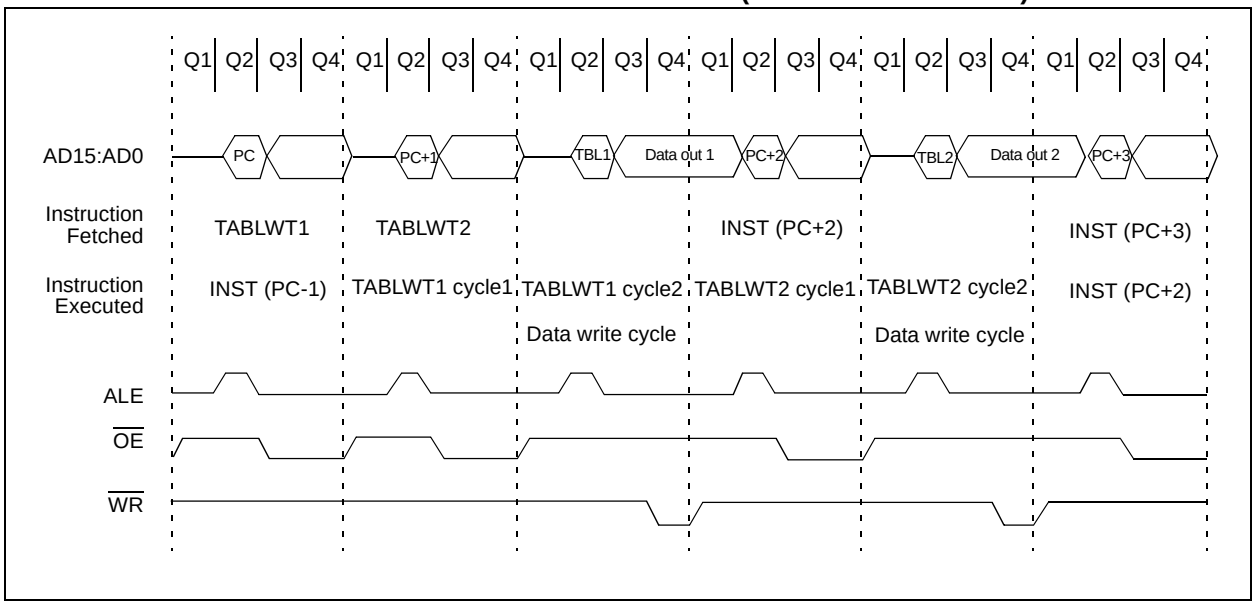
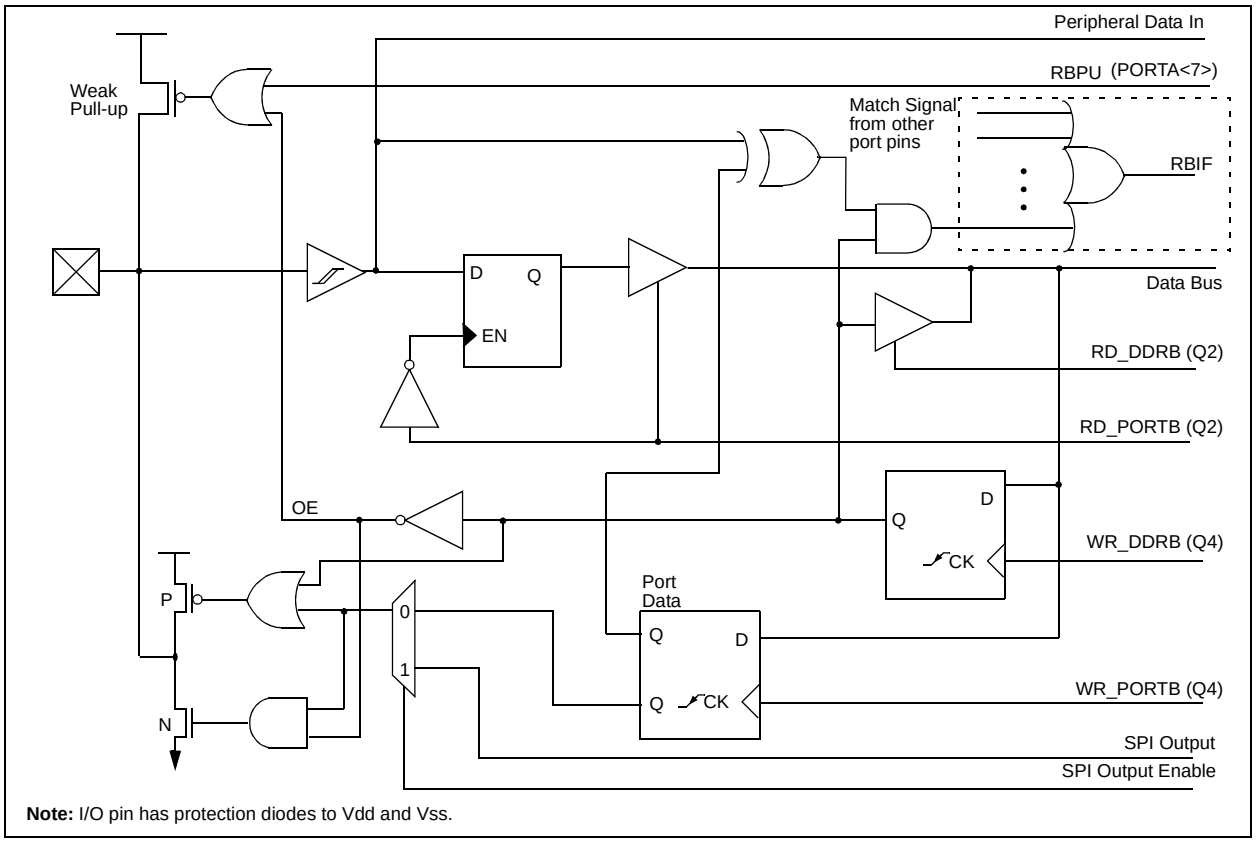
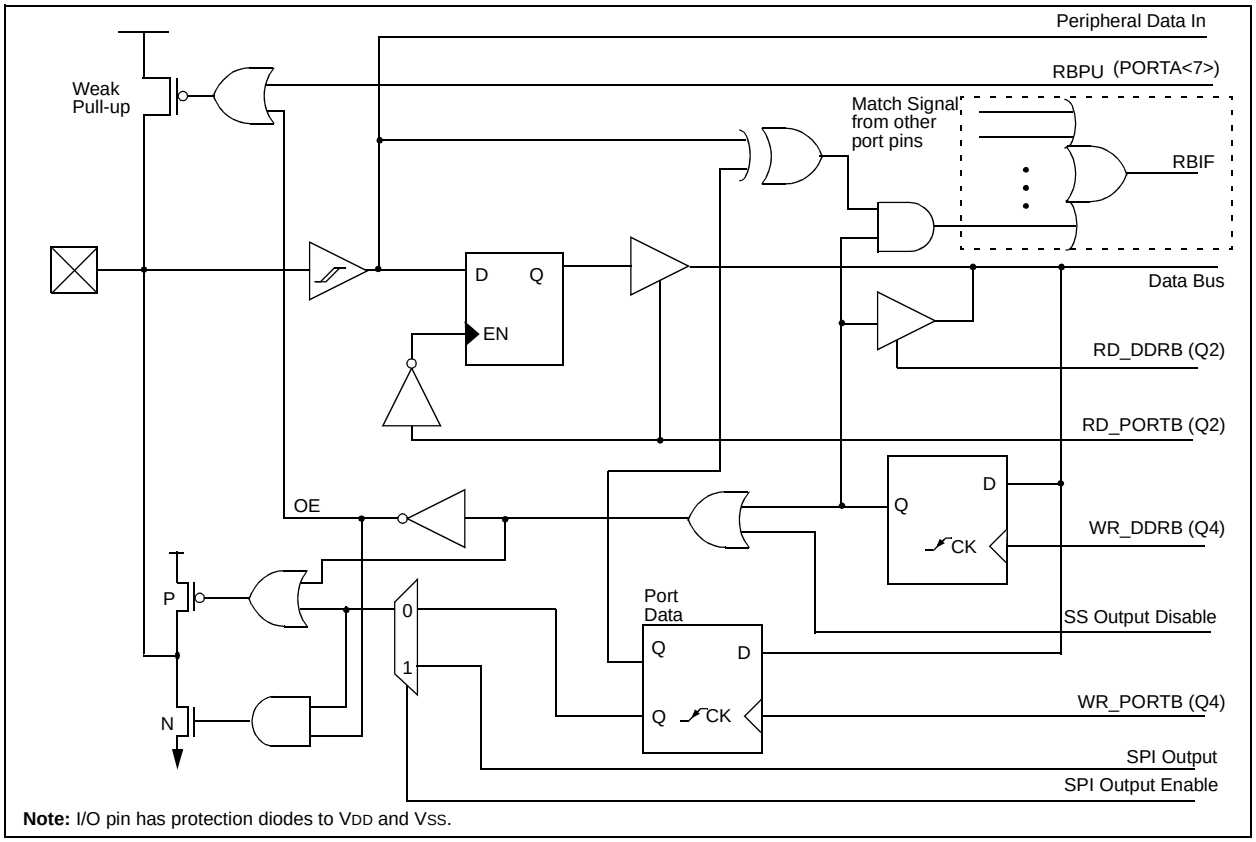


FIGURE 10-7: BLOCK DIAGRAM OF RB6 PORT PIN



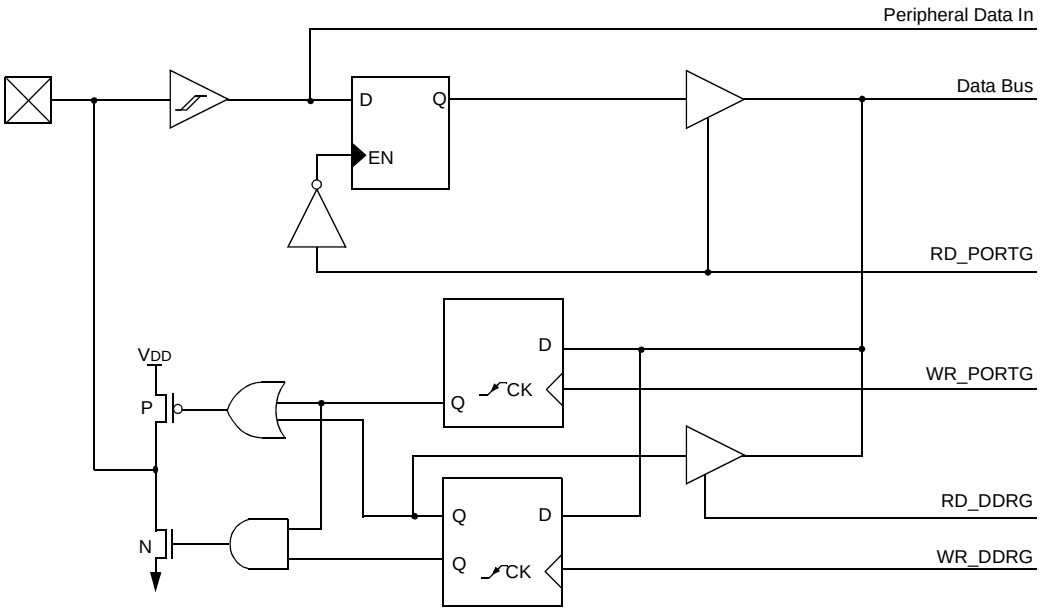
Note: I/O pin has protection diodes to Vdd and Vss.

FIGURE 10-8: BLOCK DIAGRAM OF RB7 PORT PIN



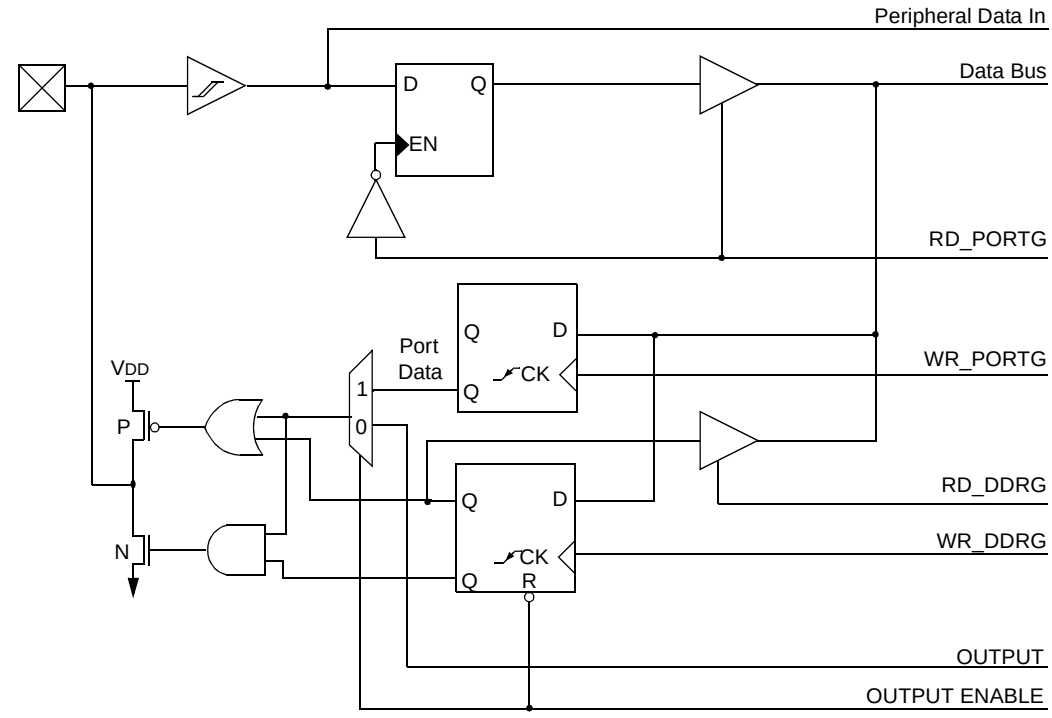
Note: I/O pin has protection diodes to VDD and VSS.

FIGURE 10-15: RG4 BLOCK DIAGRAM



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

FIGURE 10-16: RG7:RG5 BLOCK DIAGRAM



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

REGISTER 13-2: TCON2 REGISTER (ADDRESS: 17h, BANK 3)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON
bit 7							bit 0

bit 7 **CA2OVF**: Capture2 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (CA2H:CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).
1 = Overflow occurred on Capture2 register
0 = No overflow occurred on Capture2 register

bit 6 **CA1OVF**: Capture1 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (PR3H/CA1H:PR3L/CA1L), before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).
1 = Overflow occurred on Capture1 register
0 = No overflow occurred on Capture1 register

bit 5 **PWM2ON**: PWM2 On bit
1 = PWM2 is enabled
(The RB3/PWM2 pin ignores the state of the DDRB<3> bit.)
0 = PWM2 is disabled
(The RB3/PWM2 pin uses the state of the DDRB<3> bit for data direction.)

bit 4 **PWM1ON**: PWM1 On bit
1 = PWM1 is enabled
(The RB2/PWM1 pin ignores the state of the DDRB<2> bit.)
0 = PWM1 is disabled
(The RB2/PWM1 pin uses the state of the DDRB<2> bit for data direction.)

bit 3 **CA1/$\overline{PR3}$**: CA1/$\overline{PR3}$ Register Mode Select bit
1 = Enables Capture1
(PR3H/CA1H:PR3L/CA1L is the Capture1 register. Timer3 runs without a period register.)
0 = Enables the Period register
(PR3H/CA1H:PR3L/CA1L is the Period register for Timer3.)

bit 2 **TMR3ON**: Timer3 On bit
1 = Starts Timer3
0 = Stops Timer3

bit 1 **TMR2ON**: Timer2 On bit
This bit controls the incrementing of the TMR2 register. When TMR2:TMR1 form the 16-bit timer (T16 is set), TMR2ON must be set. This allows the MSB of the timer to increment.
1 = Starts Timer2 (must be enabled if the T16 bit (TCON1<3>) is set)
0 = Stops Timer2

bit 0 **TMR1ON**: Timer1 On bit

When T16 is set (in 16-bit Timer mode):
1 = Starts 16-bit TMR2:TMR1
0 = Stops 16-bit TMR2:TMR1

When T16 is clear (in 8-bit Timer mode:
1 = Starts 8-bit Timer1
0 = Stops 8-bit Timer1

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

13.1.3 USING PULSE WIDTH MODULATION (PWM) OUTPUTS WITH TIMER1 AND TIMER2

Three high speed pulse width modulation (PWM) outputs are provided. The PWM1 output uses Timer1 as its time base, while PWM2 and PWM3 may independently be software configured to use either Timer1 or Timer2 as the time base. The PWM outputs are on the RB2/PWM1, RB3/PWM2 and RG5/PWM3 pins.

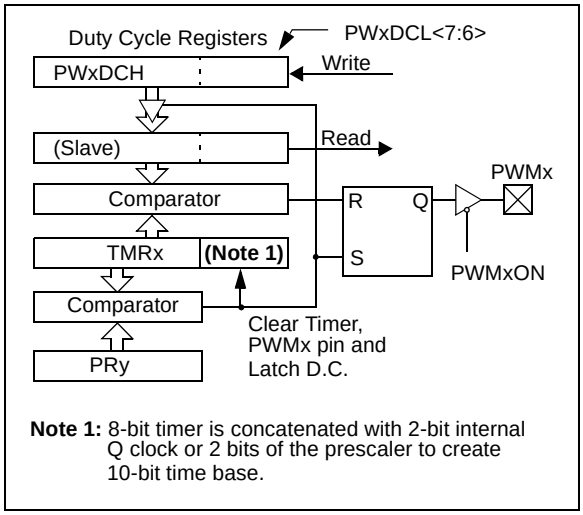
Each PWM output has a maximum resolution of 10-bits. At 10-bit resolution, the PWM output frequency is 32.2 kHz (@ 32 MHz clock) and at 8-bit resolution the PWM output frequency is 128.9 kHz. The duty cycle of the output can vary from 0% to 100%.

Figure 13-3 shows a simplified block diagram of a PWM module.

The duty cycle registers are double buffered for glitch free operation. Figure 13-4 shows how a glitch could occur if the duty cycle registers were not double buffered.

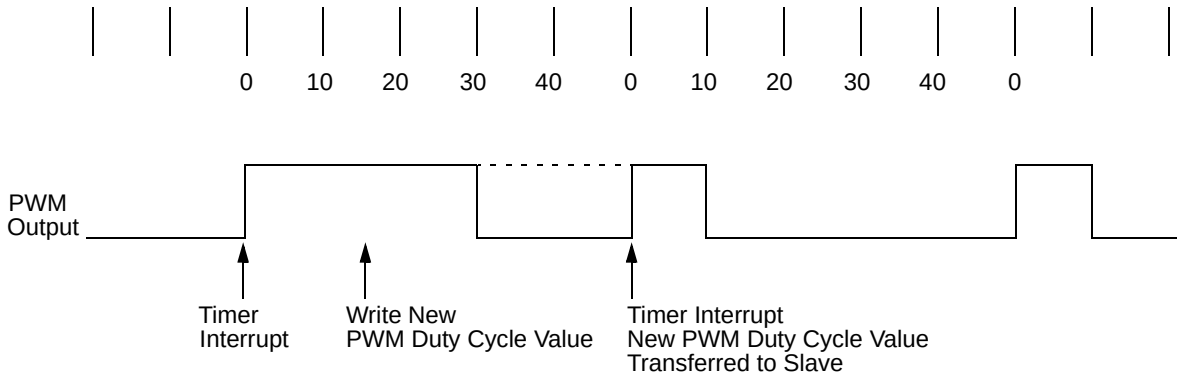
The user needs to set the PWM1ON bit (TCON2<4>) to enable the PWM1 output. When the PWM1ON bit is set, the RB2/PWM1 pin is configured as PWM1 output and forced as an output, irrespective of the data direction bit (DDRB<2>). When the PWM1ON bit is clear, the pin behaves as a port pin and its direction is controlled by its data direction bit (DDRB<2>). Similarly, the PWM2ON (TCON2<5>) bit controls the configuration of the RB3/PWM2 pin and the PWM3ON (TCON3<0>) bit controls the configuration of the RG5/PWM3 pin.

FIGURE 13-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

FIGURE 13-4: PWM OUTPUT (NOT BUFFERED)



Note: The dotted line shows PWM output if duty cycle registers were not double buffered. If the new duty cycle is written after the timer has passed that value, then the PWM does not reset at all during the current cycle, causing a "glitch".

In this example, PWM period = 50. Old duty cycle is 30. New duty cycle value is 10.

13.2.4 EXTERNAL CLOCK INPUT FOR TIMER3

When TMR3CS is set, the 16-bit TMR3 increments on the falling edge of clock input TCLK3. The input on the RB5/TCLK3 pin is sampled and synchronized by the internal phase clocks, twice every instruction cycle. This causes a delay from the time a falling edge appears on TCLK3 to the time TMR3 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section. Figure 13-7 shows the timing diagram when operating from an external clock.

13.2.5 READING/WRITING TIMER3

Since Timer3 is a 16-bit timer and only 8-bits at a time can be read or written, care should be taken when reading or writing while the timer is running. The best method is to stop the timer, perform any read or write operation and then restart Timer3 (using the TMR3ON bit). However, if it is necessary to keep Timer3 free-running, care must be taken. For writing to the 16-bit TMR3, Example 13-2 may be used. For reading the 16-bit TMR3, Example 13-3 may be used. Interrupts must be disabled during this routine.

EXAMPLE 13-2: WRITING TO TMR3

```
    BSF     CPUSTA, GLINTD      ; Disable interrupts
    MOVFP   RAM_L,  TMR3L       ;
    MOVFP   RAM_H,  TMR3H       ;
    BCF     CPUSTA, GLINTD      ; Done, enable interrupts
```

EXAMPLE 13-3: READING FROM TMR3

```
    MOVPF   TMR3L, TMPLO        ; read low TMR3
    MOVPF   TMR3H, TMPHI        ; read high TMR3
    MOVFP   TMPLO, WREG         ; tmplo –> wreg
    CPFSLT  TMR3L               ; TMR3L < wreg?
    RETURN                      ; no then return
    MOVPF   TMR3L, TMPLO        ; read low TMR3
    MOVPF   TMR3H, TMPHI        ; read high TMR3
    RETURN                      ; return
```

FIGURE 13-7: TIMER1, TIMER2 AND TIMER3 OPERATION (IN COUNTER MODE)

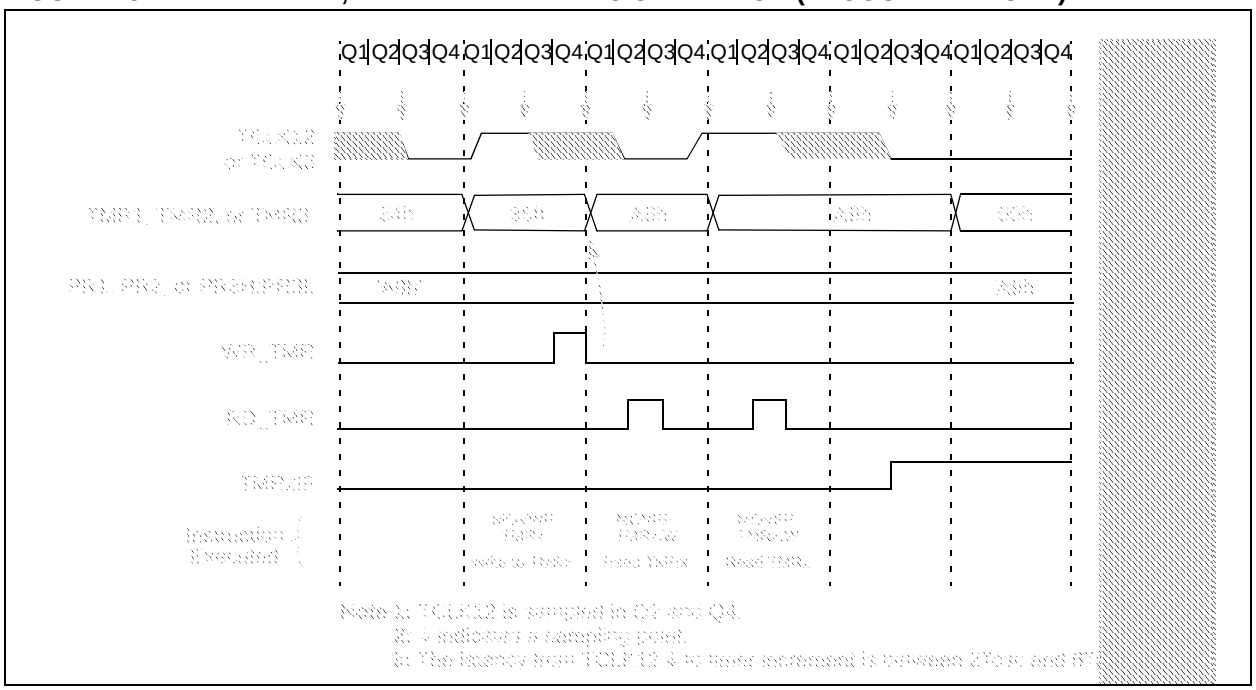


TABLE 14-5: BAUD RATES FOR ASYNCHRONOUS MODE

BAUD RATE (K)	FOSC = 33 MHz			FOSC = 25 MHz			FOSC = 20 MHz			FOSC = 16 MHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	NA	—	—	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	1.221	+1.73	255	1.202	+0.16	207
2.4	2.398	-0.07	214	2.396	0.14	162	2.404	+0.16	129	2.404	+0.16	103
9.6	9.548	-0.54	53	9.53	-0.76	40	9.469	-1.36	32	9.615	+0.16	25
19.2	19.09	-0.54	26	19.53	+1.73	19	19.53	+1.73	15	19.23	+0.16	12
76.8	73.66	-4.09	6	78.13	+1.73	4	78.13	+1.73	3	83.33	+8.51	2
96	103.12	+7.42	4	97.65	+1.73	3	104.2	+8.51	2	NA	—	—
300	257.81	-14.06	1	390.63	+30.21	0	312.5	+4.17	0	NA	—	—
500	515.62	+3.13	0	NA	—	—	NA	—	—	NA	—	—
HIGH	515.62	—	0	—	—	0	312.5	—	0	250	—	0
LOW	2.014	—	255	1.53	—	255	1.221	—	255	0.977	—	255

BAUD RATE (K)	FOSC = 10 MHz			FOSC = 7.159 MHz			FOSC = 5.068 MHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	NA	—	—	NA	—	—	0.31	+3.13	255
1.2	1.202	+0.16	129	1.203	_0.23	92	1.2	0	65
2.4	2.404	+0.16	64	2.380	-0.83	46	2.4	0	32
9.6	9.766	+1.73	15	9.322	-2.90	11	9.9	-3.13	7
19.2	19.53	+1.73	7	18.64	-2.90	5	19.8	+3.13	3
76.8	78.13	+1.73	1	NA	—	—	79.2	+3.13	0
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	156.3	—	0	111.9	—	0	79.2	—	0
LOW	0.610	—	255	0.437	—	255	0.309	—	255

BAUD RATE (K)	FOSC = 3.579 MHz			FOSC = 1 MHz			FOSC = 32.768 kHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.190	-0.83	46	1.202	+0.16	12	NA	—	—
2.4	2.432	+1.32	22	2.232	-6.99	6	NA	—	—
9.6	9.322	-2.90	5	NA	—	—	NA	—	—
19.2	18.64	-2.90	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	55.93	—	0	15.63	—	0	0.512	—	0
LOW	0.218	—	255	0.061	—	255	0.002	—	255

REGISTER 15-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 13h, BANK 6)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High Speed mode (400 kHz)

bit 6 **CKE**: SPI Clock Edge Select (Figure 15-6, Figure 15-8 and Figure 15-9)

CKP = 0:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

CKP = 1:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{A}$**: Data/$\overline{Address}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P**: STOP bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)
0 = STOP bit was not detected last

bit 3 **S**: START bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)
0 = START bit was not detected last

bit 2 **R/$\overline{W}$**: Read/Write bit Information (I^2C mode only)
This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit, or not ACK bit.

In I^2C Slave mode:
1 = Read
0 = Write

In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
Or'ing this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.

bit 1 **UA**: Update Address (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF**: Buffer Full Status bit

Receive (SPI and I^2C modes)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

Transmit (I^2C mode only)
1 = Data transmit in progress (does not include the $\overline{ACK}$ and STOP bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and STOP bits), SSPBUF is empty

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

15.2.9 I²C MASTER MODE START CONDITION TIMING

To initiate a START condition, the user sets the START condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the baud rate generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the START condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the baud rate generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the baud rate generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the baud rate generator is suspended, leaving the SDA line held low and the START condition is complete.

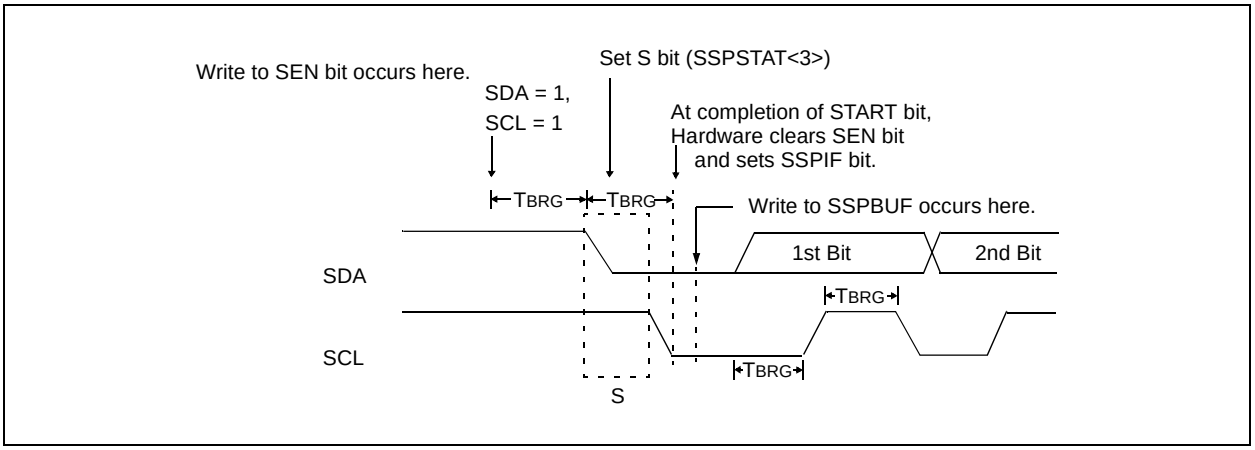
Note:	If at the beginning of START condition, the SDA and SCL pins are already sampled low, or if during the START condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted and the I²C module is reset into its IDLE state.

15.2.9.1 WCOL Status Flag

If the user writes the SSPBUF when a START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the START condition is complete.

FIGURE 15-20: FIRST START BIT TIMING



15.2.12 I^2C MASTER MODE RECEPTION

Master mode reception is enabled by programming the receive enable bit, RCEN (SSPCON2<3>).

Note:	The SSP Module must be in an IDLE STATE before the RCEN bit is set, or the RCEN bit will be disregarded.

The baud rate generator begins counting and on each rollover, the state of the SCL pin changes (high to low/ low to high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an acknowledge bit at the end of reception, by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

15.2.12.1 BF Status Flag

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

15.2.12.2 SSPOV Status Flag

In receive operation, SSPOV is set when 8 bits are received into the SSPSR, and the BF flag is already set from a previous reception.

15.2.12.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 15-36: BUS COLLISION DURING START CONDITION (SCL = 0)

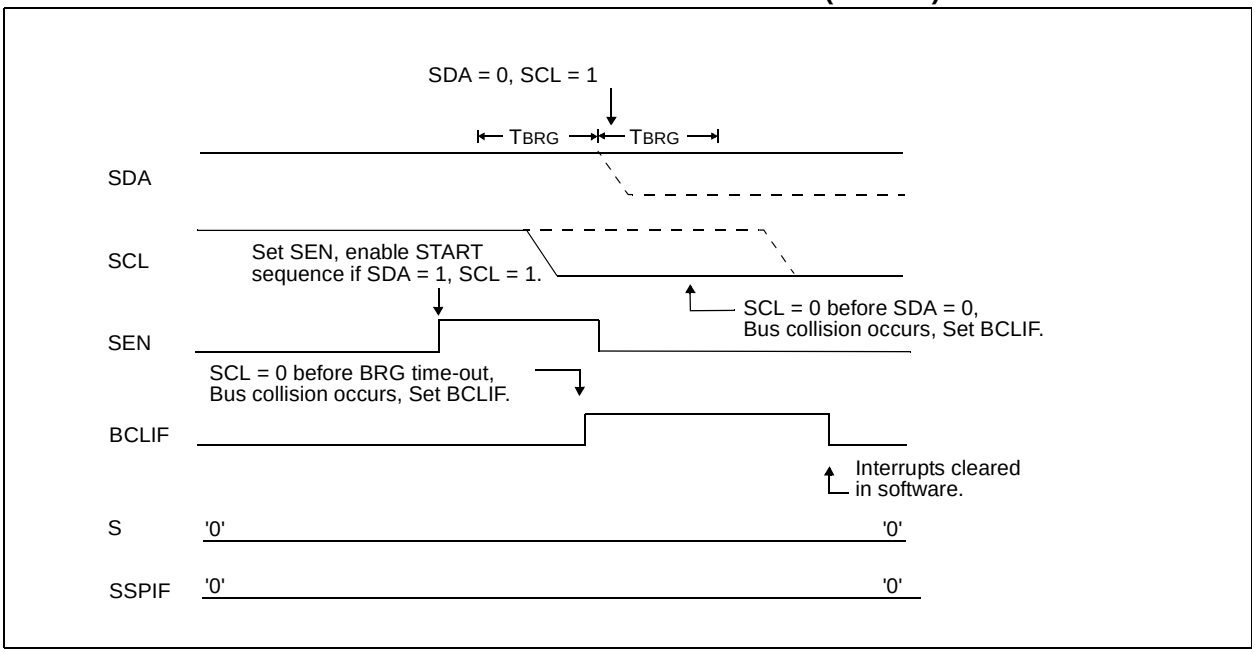
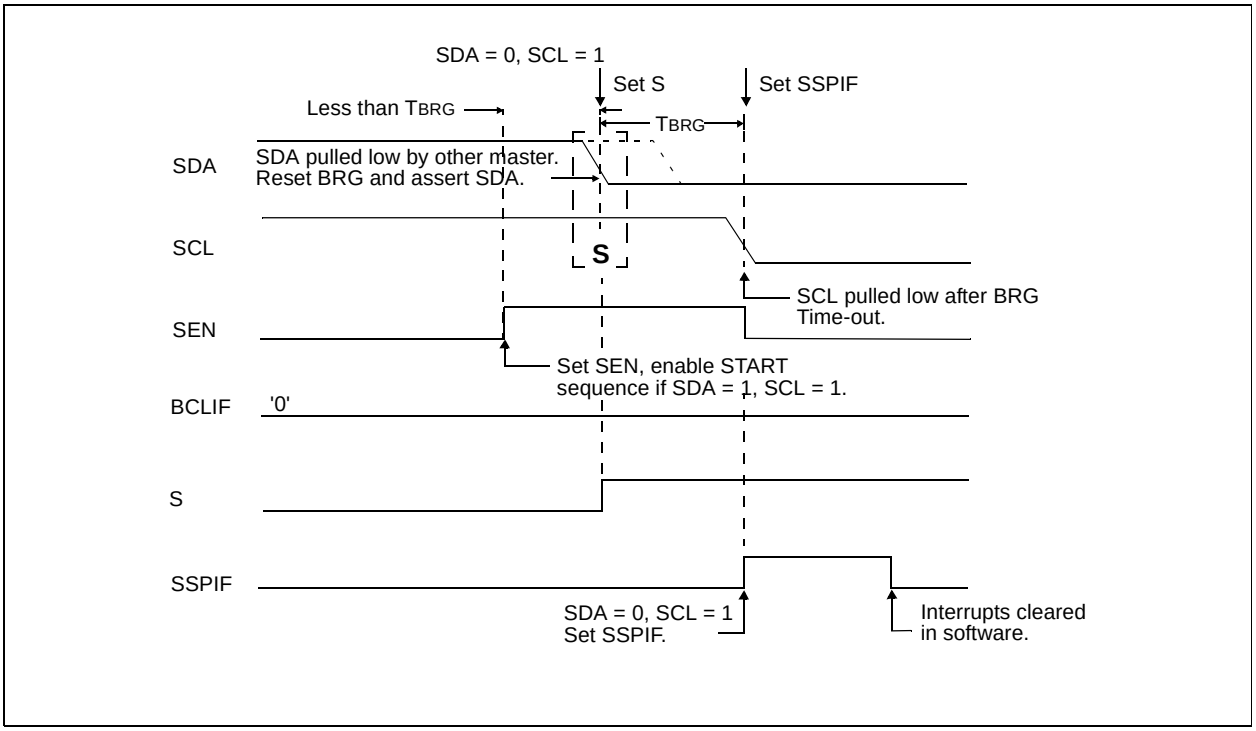


FIGURE 15-37: BRG RESET DUE TO SDA COLLISION DURING START CONDITION



16.7 A/D Accuracy/Error

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator.

The absolute accuracy specified for the A/D converter includes the sum of all contributions for quantization error, integral error, differential error, full scale error, offset error, and monotonicity. It is defined as the maximum deviation from an actual transition versus an ideal transition for any code. The absolute error of the A/D converter is specified at < ±1 LSb for VDD = VREF (over the device's specified operating range). However, the accuracy of the A/D converter will degrade as VREF diverges from VDD.

For a given range of analog inputs, the output digital code will be the same. This is due to the quantization of the analog input to a digital code. Quantization error is typically ± 1/2 LSb and is inherent in the analog to digital conversion process. The only way to reduce quantization error is to increase the resolution of the A/D converter or oversample.

Offset error measures the first actual transition of a code versus the first ideal transition of a code. Offset error shifts the entire transfer function. Offset error can be calibrated out of a system or introduced into a system through the interaction of the total leakage current and source impedance at the analog input.

Gain error measures the maximum deviation of the last actual transition and the last ideal transition adjusted for offset error. This error appears as a change in slope of the transfer function. The difference in gain error to full scale error is that full scale does not take offset error into account. Gain error can be calibrated out in software.

Linearity error refers to the uniformity of the code changes. Linearity errors cannot be calibrated out of the system. Integral non-linearity error measures the actual code transition versus the ideal code transition, adjusted by the gain error for each code.

Differential non-linearity measures the maximum actual code width versus the ideal code width. This measure is unadjusted.

The maximum pin leakage current is specified in the Device Data Sheet electrical specification (Table 20-2, parameter #D060).

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator. TAD must not violate the minimum and should be minimized to reduce inaccuracies due to noise and sampling capacitor bleed off.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

16.8 Connection Considerations

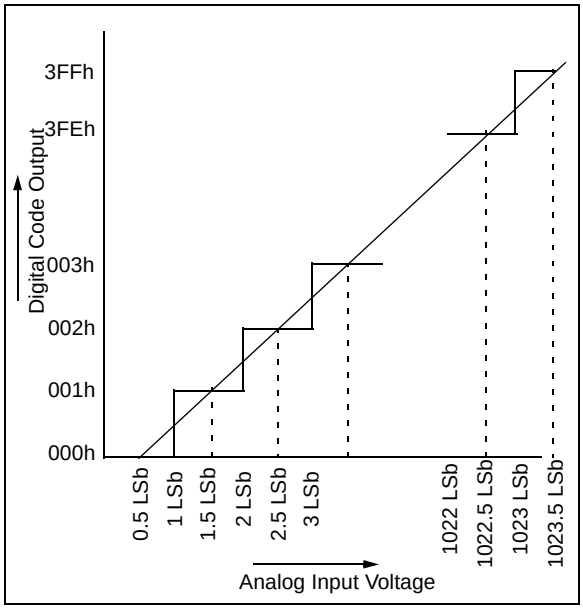
If the input voltage exceeds the rail values (VSS or VDD) by greater than 0.3V, then the accuracy of the conversion is out of specification.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 kΩ recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

16.9 Transfer Function

The transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (VAIN) equals Analog VREF / 1024 (Figure 16-7).

FIGURE 16-7: A/D TRANSFER FUNCTION



17.4 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction. This clears the Watchdog Timer and postscaler (if enabled). The $\overline{PD}$ bit is cleared and the $\overline{TO}$ bit is set (in the CPUSTA register). In SLEEP mode, the oscillator driver is turned off. The I/O ports maintain their status (driving high,low, or hi-impedance input).

The $\overline{MCLR}/V_{PP}$ pin must be at a logic high level (V_{IHMC}). A WDT time-out RESET does not drive the $\overline{MCLR}/V_{PP}$ pin low.

17.4.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

- Power-on Reset
- Brown-out Reset
- External RESET input on $\overline{MCLR}/V_{PP}$ pin
- WDT Reset (if WDT was enabled)
- Interrupt from RA0/INT pin, RB port change, T0CKI interrupt, or some peripheral interrupts

The following peripheral interrupts can wake the device from SLEEP:

- Capture interrupts
- USART synchronous slave transmit interrupts
- USART synchronous slave receive interrupts
- A/D conversion complete
- SPI slave transmit/receive complete
- I^2C slave receive

Other peripherals cannot generate interrupts since during SLEEP, no on-chip Q clocks are present.

Any RESET event will cause a device RESET. Any interrupt event is considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the CPUSTA register can be used to determine the cause of a device RESET. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused a RESET).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GLINTD bit. If the GLINTD bit is set (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GLINTD bit is clear (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt vector address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

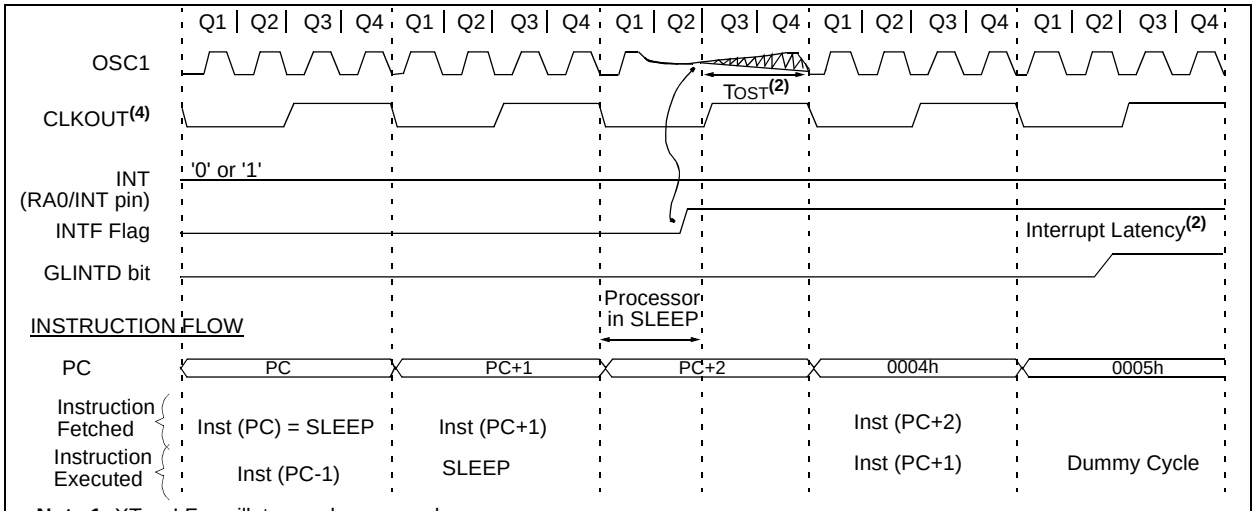
Note:	If the global interrupt is disabled (GLINTD is set), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bit set, the device will immediately wake-up from SLEEP. The $\overline{TO}$ bit is set and the $\overline{PD}$ bit is cleared.

The WDT is cleared when the device wakes from SLEEP, regardless of the source of wake-up.

17.4.1.1 Wake-up Delay

When the oscillator type is configured in XT or LF mode, the Oscillator Start-up Timer (OST) is activated on wake-up. The OST will keep the device in RESET for 1024TOSC. This needs to be taken into account when considering the interrupt response time when coming out of SLEEP.

FIGURE 17-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT or LF oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale). This delay will not be there for RC osc mode.
3: When GLINTD = 0, processor jumps to interrupt routine after wake-up. If GLINTD = 1, execution will continue in line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

0000	0000	0000	0010

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: `RETURN`

After Interrupt
PC = TOS

RLCF **Rotate Left f through Carry**

Syntax: [*label*] RLCF f,d

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

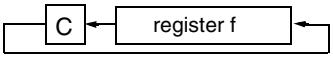
Operation: f<n> → d<n+1>;
f<7> → C;
C → d<0>

Status Affected: C

Encoding:

0001	101d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is stored back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `RLCF REG,0`

Before Instruction
REG = 1110 0110
C = 0

After Instruction
REG = 1110 0110
WREG = 1100 1100
C = 1

RLNCF **Rotate Left f (no carry)**

Syntax: [*label*] RLNCF f,d

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

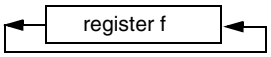
Operation: f<n> → d<n+1>;
f<7> → d<0>

Status Affected: None

Encoding:

0010	001d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is stored back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: RLNCF REG, 1

Before Instruction
C = 0
REG = 1110 1011

After Instruction
C =
REG = 1101 0111

RRCF **Rotate Right f through Carry**

Syntax: [*label*] RRCF f,d

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

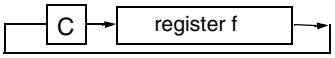
Operation: f<n> → d<n-1>;
f<0> → C;
C → d<7>

Status Affected: C

Encoding:

0001	100d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: RRCF REG1,0

Before Instruction
REG1 = 1110 0110
C = 0

After Instruction
REG1 = 1110 0110
WREG = 0111 0011
C = 0

TABLE 20-18: A/D CONVERTER CHARACTERISTICS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution		—	—	10	bit	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	10	bit	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A02	EABS	Absolute error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A03	EIL	Integral linearity error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A04	EDL	Differential linearity error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A05	EFS	Full scale error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A06	EOFF	Offset error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A10	—	Monotonicity		—	guaranteed(3)	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage (VREF+ — VREF-)		0V	—	—	V	VREF delta when changing voltage levels on VREF inputs
A20A				3V	—	—	V	Absolute minimum electrical spec. to ensure 10-bit accuracy
A21	VREF+	Reference voltage high		AVSS + 3.0V	—	AVDD + 0.3V	V	
A22	VREF-	Reference voltage low		Avss - 0.3V	—	AVDD - 3.0V	V	
A25	VAIN	Analog input voltage		AVSS - 0.3V	—	Vref + 0.3V	V	
A30	ZAIN	Recommended impedance of analog voltage source		—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	PIC17**C**XXX	—	180	—	μA	Average current consumption when A/D is on **(Note 1)**
			PIC17**LC**XXX	—	90	—	μA	
A50	IREF	VREF input current **(Note 2)**		10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN
				—	—	10	μA	During A/D conversion cycle

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RG0 and RG1 pins or AVDD and AVSS pins, whichever is selected as reference input.

3: The A/D conversion result never decreases with an increase in the Input Voltage and has no missing codes.

FIGURE 21-23: MAXIMUM AND MINIMUM VIN vs. VDD (I^2C Input, -40°C to +125°C)