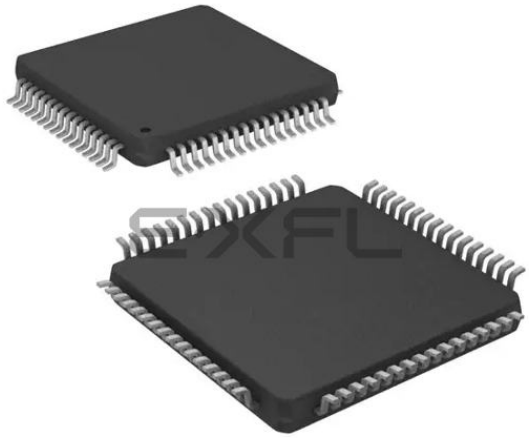


Welcome to E-XFL.COM

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.

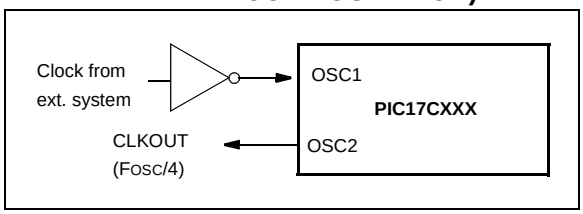
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
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	678 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°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/pic17lc752-08-pt

4.1.4 EXTERNAL CLOCK OSCILLATOR

In the EC oscillator mode, the OSC1 input can be driven by CMOS drivers. In this mode, the OSC1/CLKIN pin is hi-impedance and the OSC2/CLKOUT pin is the CLKOUT output (4 TOSC).

FIGURE 4-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



4.1.5 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used, or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 4-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 4-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

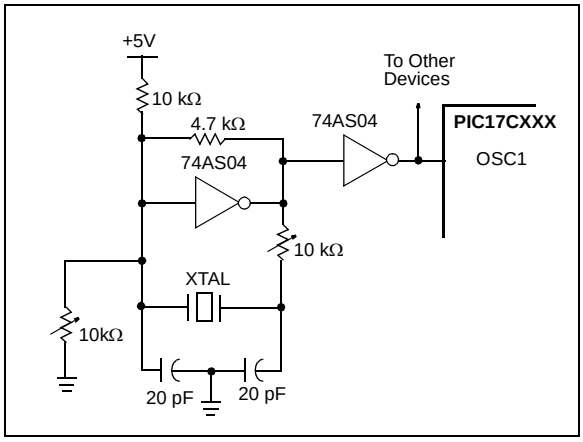
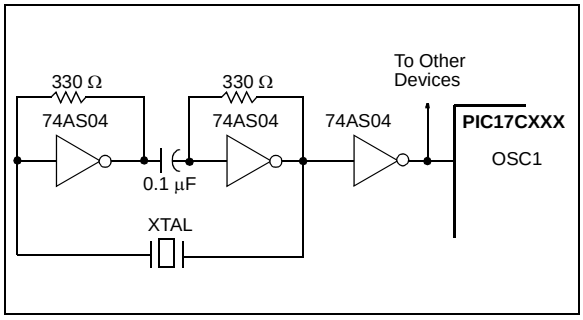


Figure 4-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 4-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



5.1 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOR)

5.1.1 POWER-ON RESET (POR)

The Power-on Reset circuit holds the device in RESET until VDD is above the trip point (in the range of 1.4V - 2.3V). The devices produce an internal RESET for both rising and falling VDD. To take advantage of the POR, just tie the MCLR/VPP pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A minimum rise time for VDD is required. See Electrical Specifications for details.

Figure 5-2 and Figure 5-3 show two possible POR circuits.

FIGURE 5-2: USING ON-CHIP POR

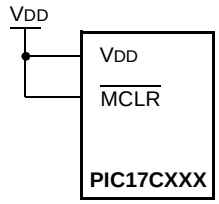
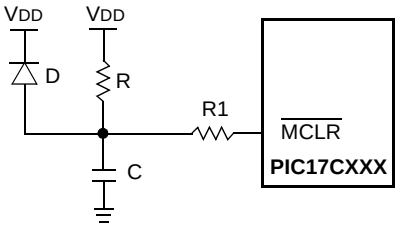


FIGURE 5-3: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: An external Power-on Reset circuit is required only if VDD power-up time is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to ensure that the voltage drop across R does not exceed 0.2V (max. leakage current spec. on the MCLR/VPP pin is 5 μA). A larger voltage drop will degrade VIH level on the MCLR/VPP pin.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

5.1.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 96 ms time-out (nominal) on power-up. This occurs from the rising edge of the internal POR signal if VDD and MCLR are tied, or after the first rising edge of MCLR (detected high). The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. In most cases, the PWRT delay allows VDD to rise to an acceptable level.

The power-up time delay will vary from chip to chip and with VDD and temperature. See DC parameters for details.

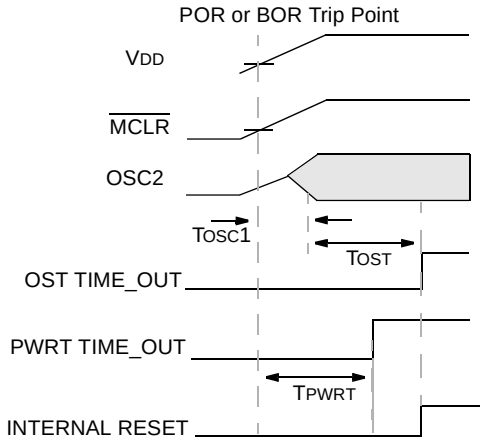
5.1.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (1024TOSC) delay whenever the PWRT is invoked, or a wake-up from SLEEP event occurs in XT or LF mode. The PWRT and OST operate in parallel.

The OST counts the oscillator pulses on the OSC1/CLKIN pin. The counter only starts incrementing after the amplitude of the signal reaches the oscillator input thresholds. This delay allows the crystal oscillator or resonator to stabilize before the device exits RESET. The length of the time-out is a function of the crystal/resonator frequency.

Figure 5-4 shows the operation of the OST circuit. In this figure, the oscillator is of such a low frequency that although enabled simultaneously, the OST does not time-out until after the Power-up Timer time-out.

FIGURE 5-4: OSCILLATOR START-UP TIME (LOW FREQUENCY)



This figure shows in greater detail the timings involved with the oscillator start-up timer. In this example, the low frequency crystal start-up time is larger than power-up time (TPWRT).

TOSC1 = time for the crystal oscillator to react to an oscillation level detectable by the Oscillator Start-up Timer (OST).

TOST = 1024TOSC.

7.2 Data Memory Organization

Data memory is partitioned into two areas. The first is the General Purpose Registers (GPR) area, and the second is the Special Function Registers (SFR) area. The SFRs control and provide status of device operation.

Portions of data memory are banked, this occurs in both areas. The GPR area is banked to allow greater than 232 bytes of general purpose RAM.

Banking requires the use of control bits for bank selection. These control bits are located in the Bank Select Register (BSR). If an access is made to the unbanked region, the BSR bits are ignored. Figure 7-5 shows the data memory map organization.

Instructions `MOVPF` and `MOVFP` provide the means to move values from the peripheral area (“P”) to any location in the register file (“F”), and vice-versa. The definition of the “P” range is from 0h to 1Fh, while the “F” range is 0h to FFh. The “P” range has six more locations than peripheral registers, which can be used as General Purpose Registers. This can be useful in some applications where variables need to be copied to other locations in the general purpose RAM (such as saving status information during an interrupt).

The entire data memory can be accessed either directly, or indirectly (through file select registers FSR0 and FSR1) (see Section 7.4). Indirect addressing uses the appropriate control bits of the BSR for access into the banked areas of data memory. The BSR is explained in greater detail in Section 7.8.

7.2.1 GENERAL PURPOSE REGISTER (GPR)

All devices have some amount of GPR area. The GPRs are 8-bits wide. When the GPR area is greater than 232, it must be banked to allow access to the additional memory space.

All the PIC17C7XX devices have banked memory in the GPR area. To facilitate switching between these banks, the `MOVLR bank` instruction has been added to the instruction set. GPRs are not initialized by a Power-on Reset and are unchanged on all other RESETS.

7.2.2 SPECIAL FUNCTION REGISTERS (SFR)

The SFRs are used by the CPU and peripheral functions to control the operation of the device (Figure 7-5). These registers are static RAM.

The SFRs can be classified into two sets, those associated with the “core” function and those related to the peripheral functions. Those registers related to the “core” are described here, while those related to a peripheral feature are described in the section for each peripheral feature.

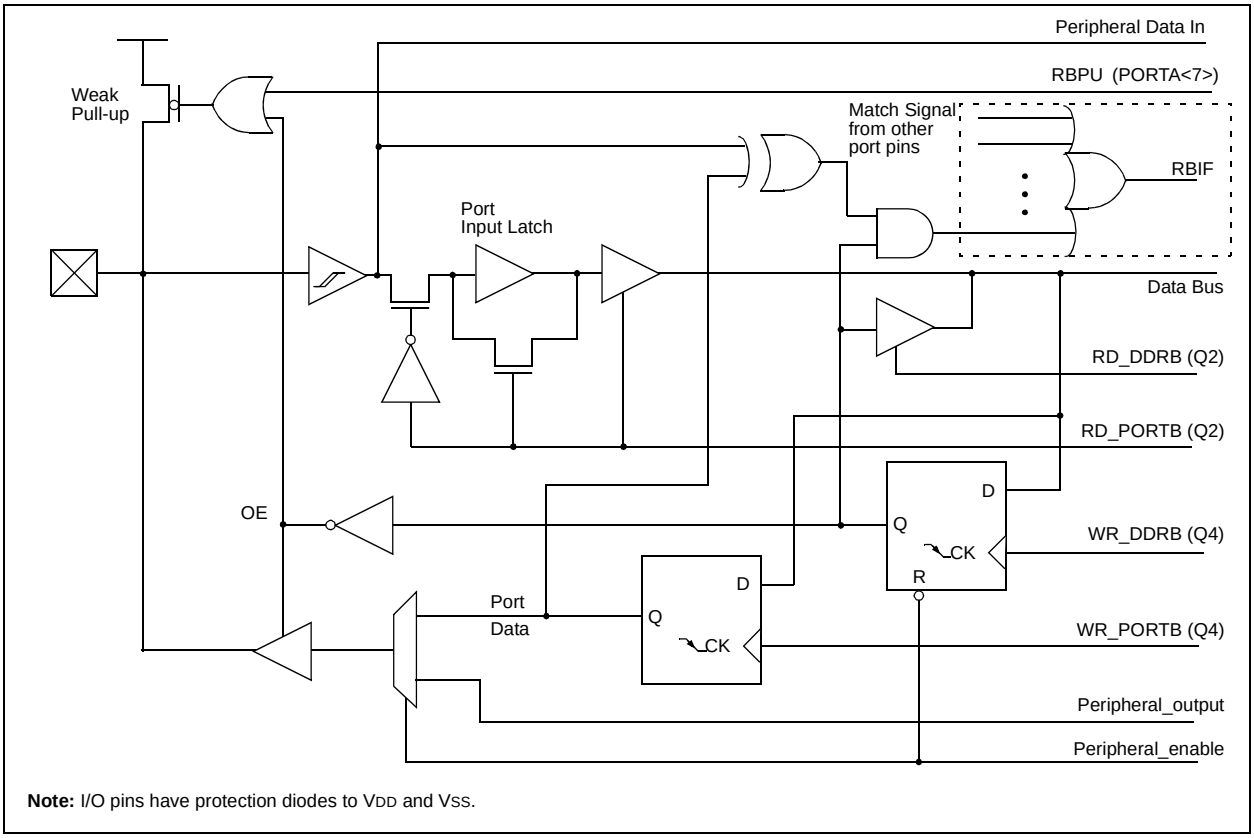
The peripheral registers are in the banked portion of memory, while the core registers are in the unbanked region. To facilitate switching between the peripheral banks, the `MOVLB bank` instruction has been provided.

Example 10-2 shows an instruction sequence to initialize PORTB. The Bank Select Register (BSR) must be selected to Bank 0 for the port to be initialized. The following example uses the MOVLB instruction to load the BSR register for bank selection.

EXAMPLE 10-2: INITIALIZING PORTB

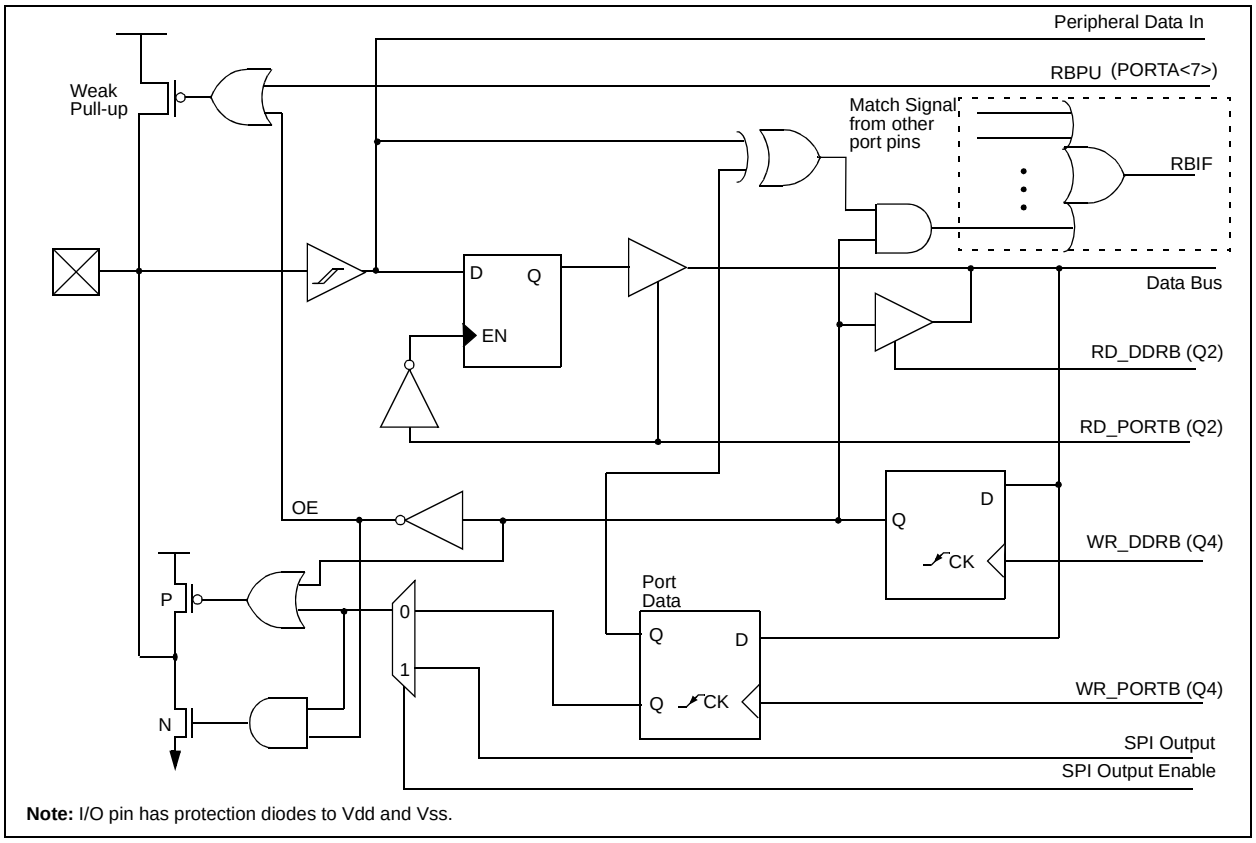
```
MOVLB   0          ; Select Bank 0
CLRF    PORTB, F   ; Init PORTB by clearing
                   ;  output data latches
MOVLW   0xCF       ; Value used to initialize
                   ;  data direction
MOVWF   DDRB       ; Set RB<3:0> as inputs
                   ;   RB<5:4> as outputs
                   ;   RB<7:6> as inputs
```

FIGURE 10-6: BLOCK DIAGRAM OF RB3:RB2 PORT PINS



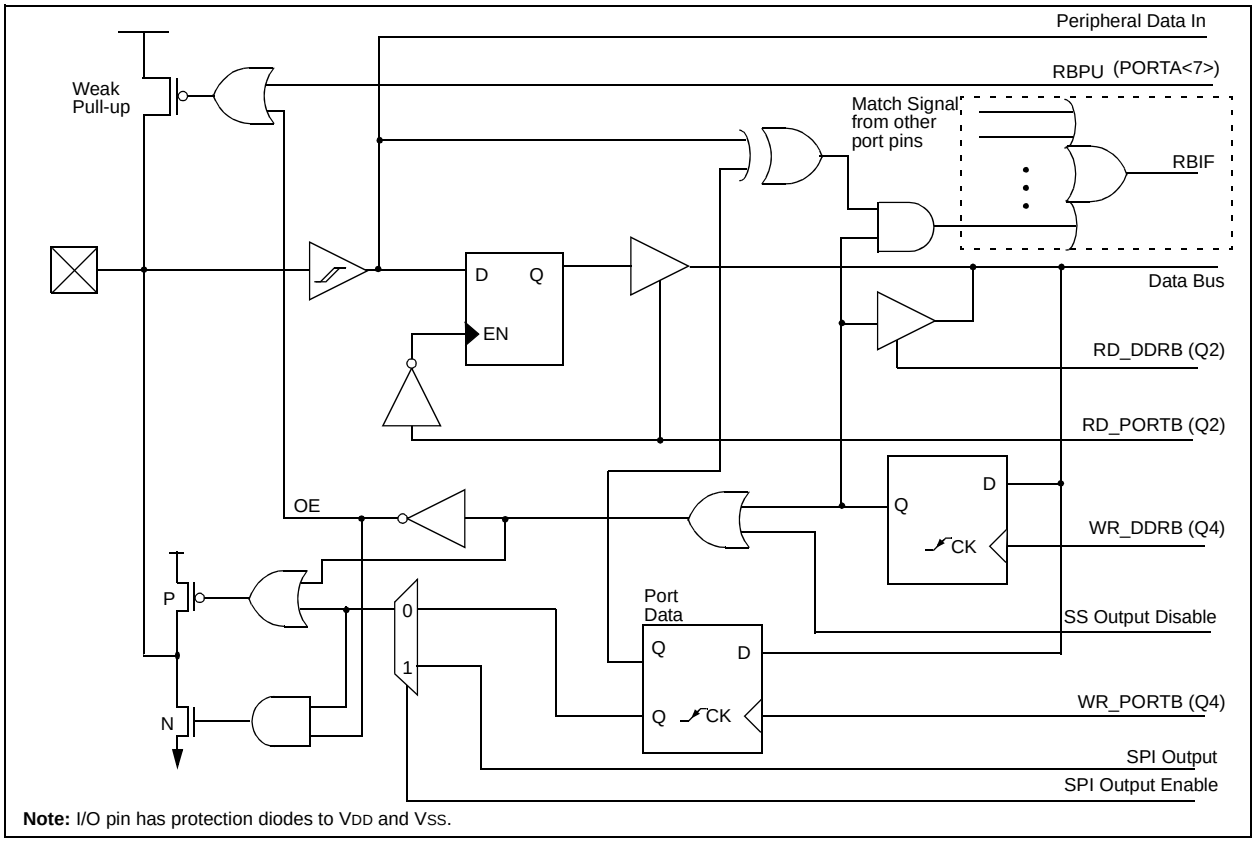
Note: I/O pins have protection diodes to VDD and VSS.

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.

10.7 PORTG and DDRG Registers

PORTG is an 8-bit wide, bi-directional port. The corresponding data direction register is DDRG. A '1' in DDRG configures the corresponding port pin as an input. A '0' in the DDRG register configures the corresponding port pin as an output. Reading PORTG reads the status of the pins, whereas writing to PORTG will write to the port latch.

The lower four bits of PORTG are multiplexed with four channels of the 10-bit A/D converter.

The remaining bits of PORTG are multiplexed with peripheral output and inputs. RG4 is multiplexed with the CAP3 input, RG5 is multiplexed with the PWM3 output, RG6 and RG7 are multiplexed with the USART2 functions.

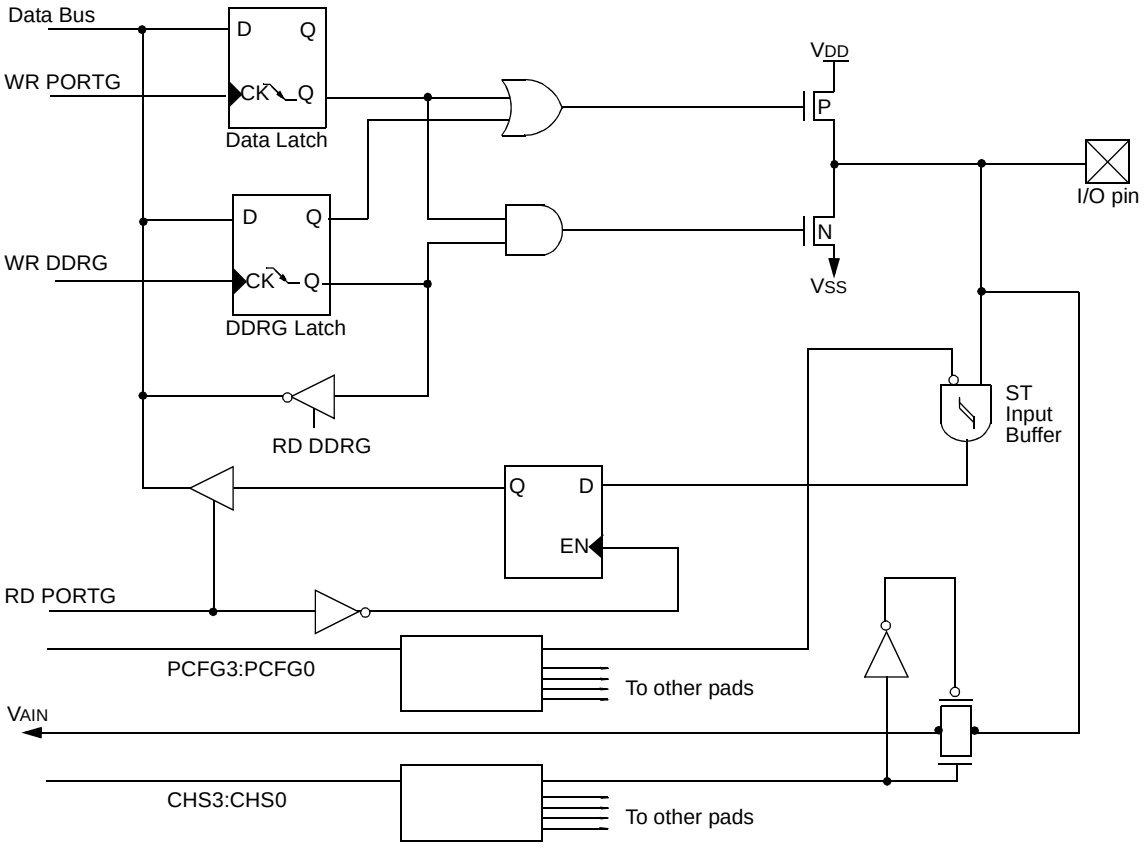
Upon RESET, RG3:RG0 is automatically configured as analog inputs and must be configured in software to be a digital I/O.

Example 10-7 shows the instruction sequence to initialize PORTG. The Bank Select Register (BSR) must be selected to Bank 5 for the port to be initialized. The following example uses the `MOVLB` instruction to load the BSR register for bank selection.

EXAMPLE 10-7: INITIALIZING PORTG

```
MOVLB  5              ; Select Bank 5
MOVLW  0x0E           ; Configure PORTG as
MOVPF  WREG, ADCON1   ; digital
CLRF   PORTG, F      ; Initialize PORTG data
                      ;   latches before
                      ;   the data direction
                      ;   register
MOVLW   0x03          ; Value used to init
                      ;   data direction
MOVWF   DDRG         ; Set RG<1:0> as inputs
                      ;   RG<7:2> as outputs
```

FIGURE 10-14: BLOCK DIAGRAM OF RG3:RG0



Note: I/O pins have protection diodes to V_{DD} and V_{SS}.

TABLE 14-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
13h, Bank 0	RCSTA1	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
15h, Bank 0	TXSTA1	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
16h, Bank 0	TXREG1	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
17h, Bank 0	SPBRG1	Baud Rate Generator Register								0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
13h, Bank 4	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
16h, Bank 4	TXREG2	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
15h, Bank 4	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 4	SPBRG2	Baud Rate Generator Register								0000 0000	0000 0000

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

TABLE 14-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
13h, Bank0	RCSTA1	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank0	RCREG1	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
15h, Bank 0	TXSTA1	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG1	Baud Rate Generator Register								0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
13h, Bank 4	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 4	RCREG2	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
15h, Bank 4	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 4	SPBRG2	Baud Rate Generator Register								0000 0000	0000 0000

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

REGISTER 15-3: SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 12h, BANK 6)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

bit 7 **GCEN**: General Call Enable bit (in I^2C Slave mode only)
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT**: Acknowledge Status bit (in I^2C Master mode only)

In Master Transmit mode:
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT**: Acknowledge Data bit (in I^2C Master mode only)

In Master Receive mode:
Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN**: Acknowledge Sequence Enable bit (in I^2C Master mode only)

In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit AKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

Note: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

bit 3 **RCEN**: Receive Enable bit (in I^2C Master mode only)
1 = Enables Receive mode for I^2C
0 = Receive idle

Note: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

bit 2 **PEN**: STOP Condition Enable bit (in I^2C Master mode only)

SCK Release Control:
1 = Initiate STOP condition on SDA and SCL pins. Automatically cleared by hardware.
0 = STOP condition idle

Note: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

bit 1 **RSEN**: Repeated Start Condition Enabled bit (in I^2C Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition idle

Note: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

bit 0 **SEN**: START Condition Enabled bit (In I^2C Master mode only)
1 = Initiate START condition on SDA and SCL pins. Automatically cleared by hardware.
0 = START condition idle.

Note: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

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

15.1 SPI Mode

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$)

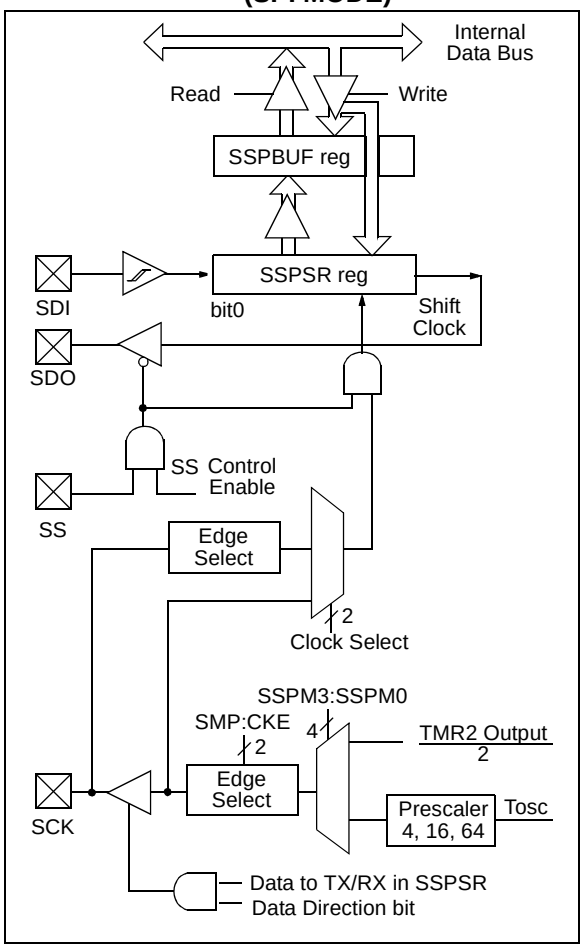
15.1.1 OPERATION

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

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

Figure 15-4 shows the block diagram of the MSSP module when in SPI mode.

FIGURE 15-4: MSSP BLOCK DIAGRAM (SPI MODE)



The MSSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit BF (SSPSTAT<0>) and the interrupt flag bit SSPIF (PIR2<7>) are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL (SSPCON1<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

15.2.1.1 Addressing

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register on the falling edge of the 8th SCL pulse.
b) The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
c) An $\overline{ACK}$ pulse is generated.
d) SSP interrupt flag bit, SSPIF (PIR2<7>), is set (interrupt is generated if enabled) - on the falling edge of the 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{W}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address. The sequence of events for a 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of Address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of Address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of Address. This will clear bit UA and release the SCL line.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of Address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

Note:	Following the Repeated Start condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

15.2.1.2 Slave Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR2<7>) must be cleared in software. The SSPSTAT register is used to determine the status of the received byte.

Note:	The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the $\overline{ACK}$ is not sent and the SSPBUF is updated.

TABLE 15-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	Yes	No	Yes

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

17.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC17CXXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection (Section 4.0)
- RESET (Section 5.0)
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts (Section 6.0)
- Watchdog Timer (WDT)
- SLEEP mode
- Code protection

The PIC17CXXX has a Watchdog Timer which can be shut-off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on POR and BOR. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external RESET, Watchdog Timer Reset, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 17-1.

REGISTER 17-1: CONFIGURATION WORDS

High (H) Table Read Addr.	U-x	R/P-1	R/P-1	U-x	U-x	U-x	U-x	U-x	U-x
FE0Fh - FE08h	—	PM2	BODEN	—	—	—	—	—	—
	bit 15 bit 8	bit 7							bit 0

Low (L) Table Read Addr.	U-x	U-x	R/P-1	U-x	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FE07h - FE00h	—	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0
	bit 15 bit 8	bit 7							bit 0

bits 7H, 6L, 4L	**PM2, PM1, PM0:** Processor Mode Select bits 111 = Microprocessor mode 110 = Microcontroller mode 101 = Extended Microcontroller mode 000 = Code Protected Microcontroller mode
bit 6H	**BODEN:** Brown-out Detect Enable 1 = Brown-out Detect circuitry is enabled 0 = Brown-out Detect circuitry is disabled
bits 3L:2L	**WDTPS1:WDTPS0**: WDT Postscaler Select bits 11 = WDT enabled, postscaler = 1 10 = WDT enabled, postscaler = 256 01 = WDT enabled, postscaler = 64 00 = WDT disabled, 16-bit overflow timer
bits 1L:0L	**FOSC1:FOSC0**: Oscillator Select bits 11 = EC oscillator 10 = XT oscillator 01 = RC oscillator 00 = LF oscillator
Shaded bits (—)	**Reserved**

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: $1 \rightarrow (f<b>)$

Status Affected: None

Encoding:

1000	0bbb	ffff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `BSF FLAG_REG, 7`

Before Instruction
FLAG_REG = 0x0A

After Instruction
FLAG_REG = 0x8A

BTFSC **Bit Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1001	1bbb	ffff	ffff

Description: If bit 'b' in register 'f' is 0, then the next instruction is skipped.

If bit 'b' is 0, then the next instruction fetched during the current instruction execution is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

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

Example:
```
HERE    BTFSC   FLAG,1
FALSE   :
TRUE    :
```

Before Instruction
PC = address `(HERE)`

After Instruction
If FLAG<1> = 0;
PC = address `(TRUE)`
If FLAG<1> = 1;
PC = address `(FALSE)`

INCF **Increment f**

Syntax: [*label*] INCF f,d

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

Operation: (f) + 1 → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0001	010d	ffff	ffff

Description: The contents of register 'f' are incremented. 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: `INCF CNT, 1`

Before Instruction
CNT = 0xFF
Z = 0
C = ?

After Instruction
CNT = 0x00
Z = 1
C = 1

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

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

Operation: (f) + 1 → (dest)
skip if result = 0

Status Affected: None

Encoding:

0001	111d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If skip:

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

Example:
```
HERE    INCFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT + 1
If CNT = 0;
PC = Address (ZERO)
If CNT ≠ 0;
PC = Address (NZERO)

INFSNZ **Increment f, skip if not 0**

Syntax: [*label*] INFSNZ f,d

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

Operation: (f) + 1 → (dest),
skip if not 0

Status Affected: None

Encoding:

0010	010d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.
If the result is not 0, the next instruction, which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If skip:

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

Example:
```
HERE    INFSNZ  REG, 1
ZERO
NZERO
```

Before Instruction
REG = REG

After Instruction
REG = REG + 1
If REG = 1;
PC = Address (ZERO)
If REG = 0;
PC = Address (NZERO)

IORLW **Inclusive OR Literal with WREG**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (WREG) .OR. (k) → (WREG)

Status Affected: Z

Encoding:

1011	0011	kkkk	kkkk

Description: The contents of WREG are OR'ed with the eight-bit literal 'k'. The result is placed in WREG.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to WREG

Example:
```
IORLW   0x35
```

Before Instruction
WREG = 0x9A

After Instruction
WREG = 0xBF

MOVFP **Move f to p**

Syntax: [*label*] MOVFP f,p

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (f) → (p)

Status Affected: None

Encoding:

011p	pppp	ffff	ffff

Description: Move data from data memory location 'f' to data memory location 'p'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh), while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful, special situation).

MOVFP is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: MOVFP REG1, REG2

Before Instruction
REG1 = 0x33,
REG2 = 0x11

After Instruction
REG1 = 0x33,
REG2 = 0x33

MOVLB **Move Literal to low nibble in BSR**

Syntax: [*label*] MOVLB k

Operands: $0 \le k \le 15$

Operation: k → (BSR<3:0>)

Status Affected: None

Encoding:

1011	1000	uuuu	kkkk

Description: The four-bit literal 'k' is loaded in the Bank Select Register (BSR). Only the low 4-bits of the Bank Select Register are affected. The upper half of the BSR is unchanged. The assembler will encode the “u” fields as '0'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR<3:0>

Example: MOVLB 5

Before Instruction
BSR register = 0x22

After Instruction
BSR register = 0x25 (Bank 5)

NOTES:

20.2 DC Characteristics: PIC17C7XX-16 (Commercial, Industrial, Extended) PIC17C7XX-33 (Commercial, Industrial, Extended) PIC17LC7XX-08 (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C for extended -40°C ≤ TA ≤ +85°C for industrial 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in Section 20.1				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Input Low Voltage					
	VIL	I/O ports					
D030		with TTL buffer **(Note 6)**	Vss	–	0.8	V	4.5V ≤ VDD ≤ 5.5V
			Vss	–	0.2VDD	V	3.0V ≤ VDD ≤ 4.5V
D031		with Schmitt Trigger buffer					
		RA2, RA3	Vss	–	0.3VDD	V	I^2C compliant
		All others	Vss	–	0.2VDD	V	
D032		$\overline{MCLR}$, OSC1 (in EC and RC mode)	Vss	–	0.2VDD	V	**(Note 1)**
D033		OSC1 (in XT, and LF mode)	–	0.5VDD	–	V	
		Input High Voltage					
	VIH	I/O ports					
D040		with TTL buffer **(Note 6)**	2.0	–	VDD	V	4.5V ≤ VDD ≤ 5.5V
			1 + 0.2VDD	–	VDD	V	3.0V ≤ VDD ≤ 4.5V
D041		with Schmitt Trigger buffer					
		RA2, RA3	0.7VDD	–	VDD	V	I^2C compliant
		All others	0.8VDD	–	VDD	V	
D042		$\overline{MCLR}$	0.8VDD	–	VDD	V	**(Note 1)**
D043		OSC1 (XT, and LF mode)	–	0.5VDD	–	V	
D050	VHYS	**Hysteresis of Schmitt Trigger Inputs**	0.15VDD	–	–	V	

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

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17C7XX Programming Specifications (Literature number DS TBD).

5: The $\overline{MCLR}$/VPP pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

FIGURE 20-7: CLKOUT AND I/O TIMING

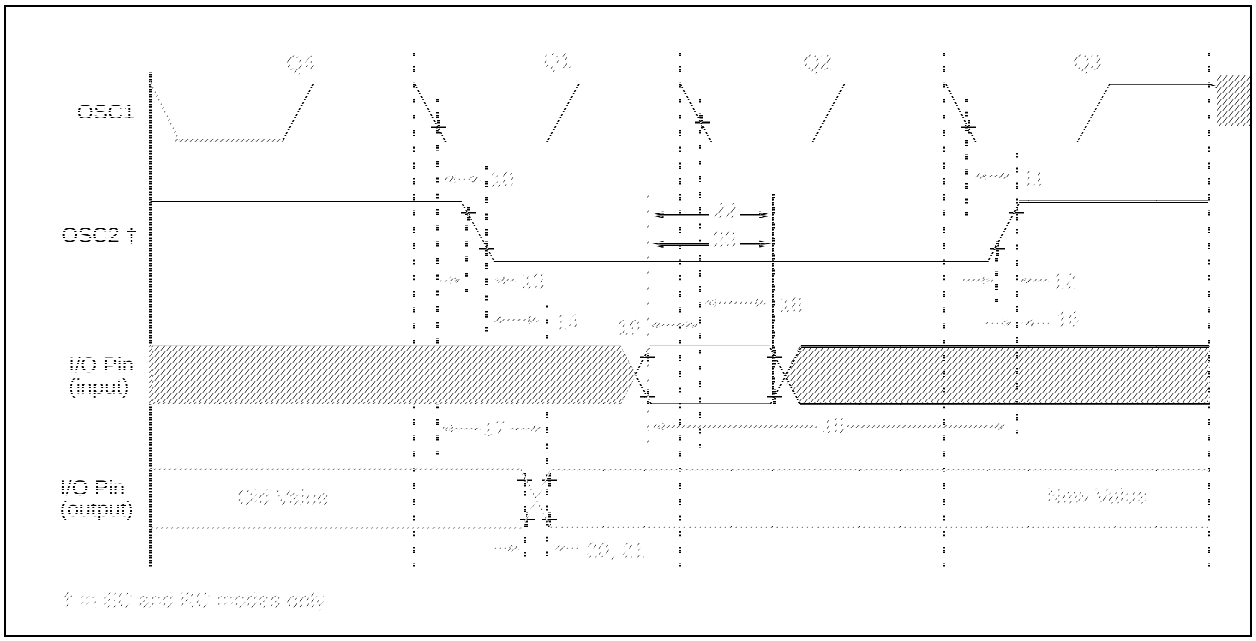


TABLE 20-2: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosL2ckL	OSC1↓ to CLKOUT↓	—	15	30	ns	**(Note 1)**
11	TosL2ckH	OSC1↓ to CLKOUT↑	—	15	30	ns	**(Note 1)**
12	TckR	CLKOUT rise time	—	5	15	ns	**(Note 1)**
13	TckF	CLKOUT fall time	—	5	15	ns	**(Note 1)**
14	TckH2ioV	CLKOUT ↑ to Port out valid	—	—	0.5TCY + 20	ns	**(Note 1)**
15	TioV2ckH	Port in valid before CLKOUT↑	0.25TCY + 25	—	—	ns	**(Note 1)**
16	TckH2ioI	Port in hold after CLKOUT↑	0	—	—	ns	**(Note 1)**
17	TosL2ioV	OSC1↓ (Q1 cycle) to Port out valid	—	—	100	ns	
18	TosL2ioI	OSC1↓ (Q2 cycle) to Port input invalid (I/O in hold time)	0	—	—	ns	
19	TioV2osL	Port input valid to OSC1↓ (I/O in setup time)	30	—	—	ns	
20	TioR	Port output rise time	—	10	35	ns	
21	TioF	Port output fall time	—	10	35	ns	
22	TinHL	INT pin high or low time	25	—	—	ns	
23	TrbHL	RB7:RB0 change INT high or low time	25	—	—	ns	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in EC mode, where CLKOUT output is 4 x TOSC.

FIGURE 20-25: MEMORY INTERFACE READ TIMING

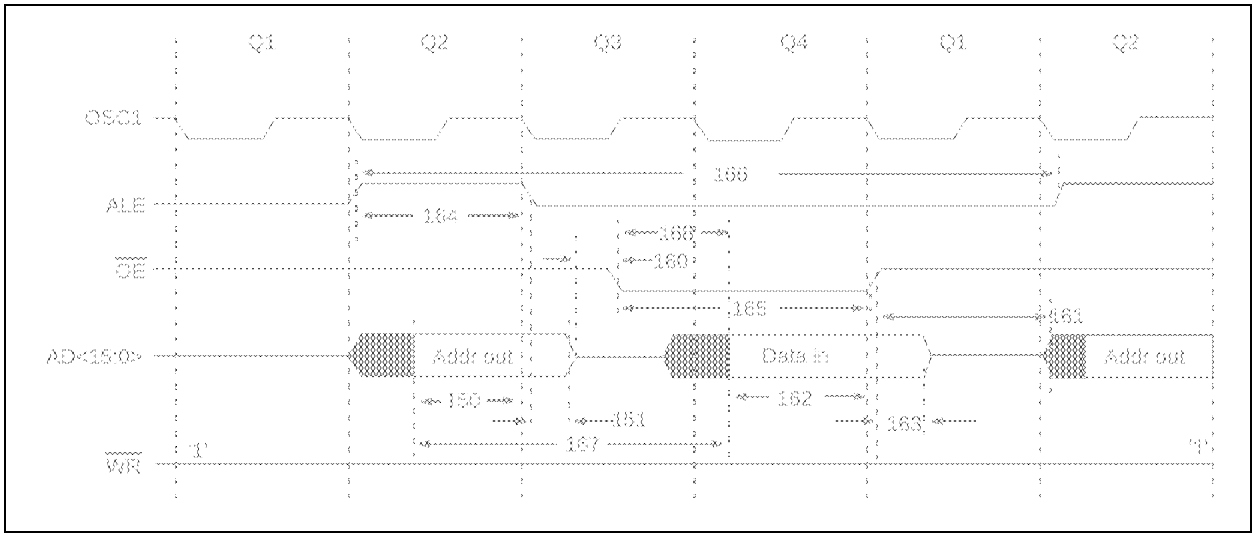


TABLE 20-21: MEMORY INTERFACE READ REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD15:AD0 (address) valid to ALE↓ (address setup time)	PIC17**C**XXX	0.25TCY - 10	—	—	ns	
			PIC17**LC**XXX	0.25TCY - 10	—	—		
151	TalL2adI	ALE↓ to address out invalid (address hold time)	PIC17**C**XXX	5	—	—	ns	
			PIC17**LC**XXX	5	—	—		
160	TadZ2oeL	AD15:AD0 hi-impedance to $\overline{OE}$↓	PIC17**C**XXX	0	—	—	ns	
			PIC17**LC**XXX	0	—	—		
161	ToeH2adD	$\overline{OE}$↑ to AD15:AD0 driven	PIC17**C**XXX	0.25TCY - 15	—	—	ns	
			PIC17**LC**XXX	0.25TCY - 15	—	—		
162	TadV2oeH	Data in valid before $\overline{OE}$↑ (data setup time)	PIC17**C**XXX	35	—	—	ns	
			PIC17**LC**XXX	45	—	—		
163	ToeH2adI	$\overline{OE}$↑to data in invalid (data hold time)	PIC17**C**XXX	0	—	—	ns	
			PIC17**LC**XXX	0	—	—		
164	TalH	ALE pulse width	PIC17**C**XXX	—	0.25TCY	—	ns	
			PIC17**LC**XXX	—	0.25TCY	—		
165	ToeL	$\overline{OE}$ pulse width	PIC17**C**XXX	0.5TCY - 35	—	—	ns	
			PIC17**LC**XXX	0.5TCY - 35	—	—		
166	TalH2alH	ALE↑ to ALE↑(cycle time)	PIC17**C**XXX	—	TCY	—	ns	
			PIC17**LC**XXX	—	TCY	—		
167	Tacc	Address access time	PIC17**C**XXX	—	—	0.75TCY - 30	ns	
			PIC17**LC**XXX	—	—	0.75TCY - 45		
168	Toe	Output enable access time ($\overline{OE}$ low to data valid)	PIC17**C**XXX	—	—	0.5TCY - 45	ns	
			PIC17**LC**XXX	—	—	0.5TCY - 75		

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

FIGURE 21-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

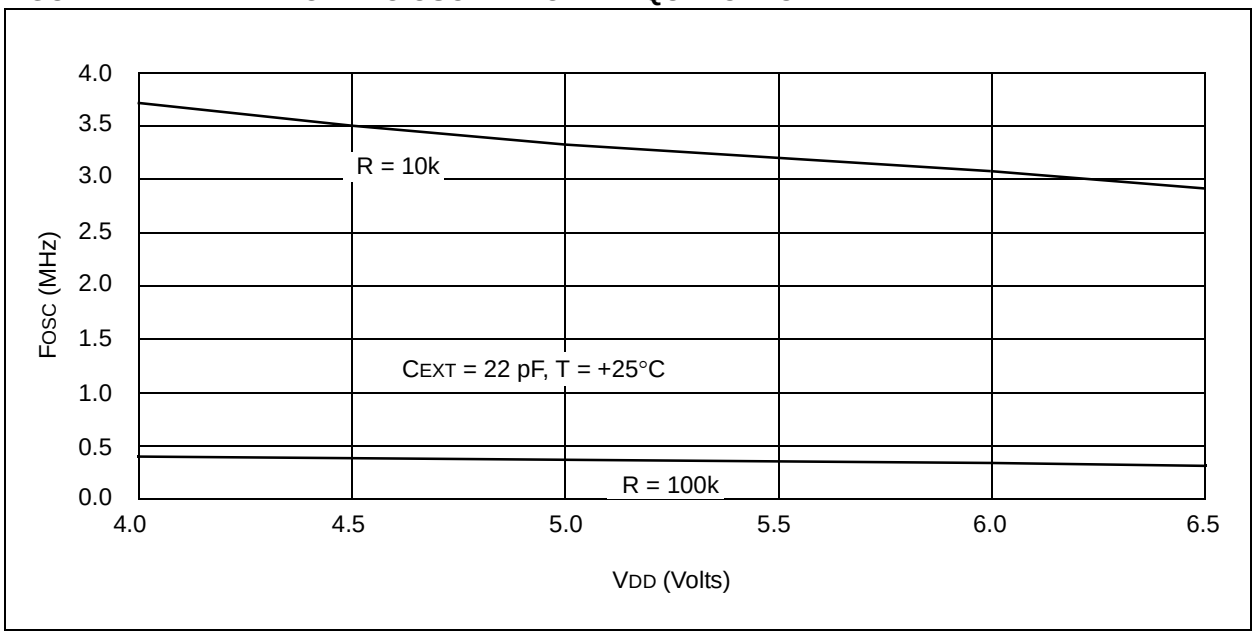


FIGURE 21-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD