MICROCHIP

PIC16F91X Data Sheet

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

PIC16F91X

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Program Memory Read (PMR) capability
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 32 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced Low-Current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High-Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years

Low-Power Features

- Standby Current:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5 μA @ 32 kHz, 2.0V, typical
  - 100 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical

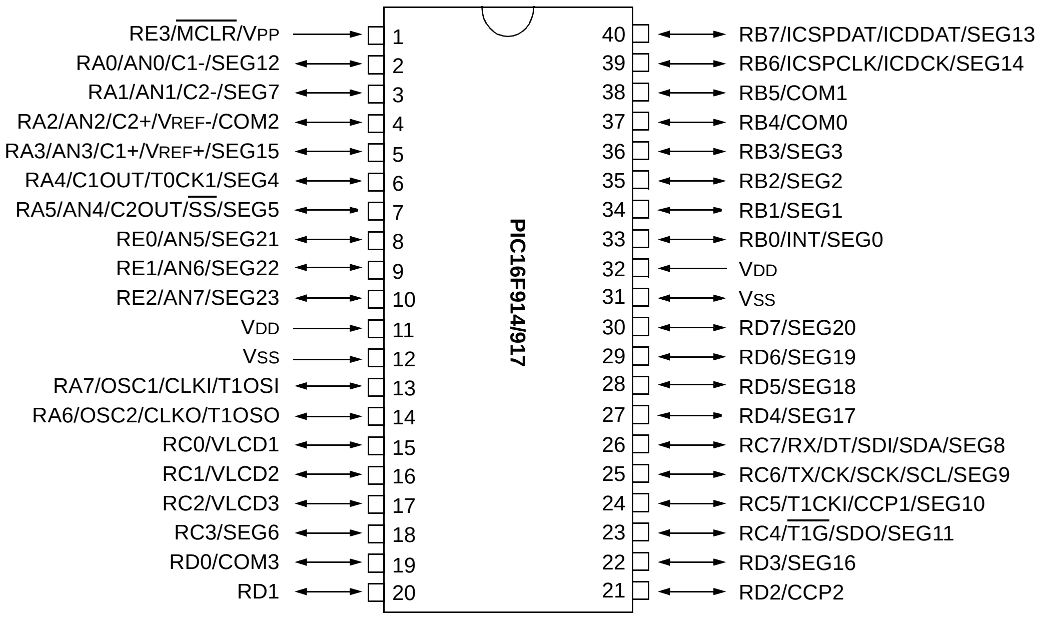
Peripheral Features

- Liquid Crystal Display module:
  - Up to 24 segment drivers
  - Four commons
- Up to 35 I/O pins and 1 input-only pin:
  - High-current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - Comparator inputs and outputs externally accessible
- A/D Converter:
  - 10-bit resolution and up to 8 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 as Timer1 oscillator if INTOSCIO or LP mode is selected
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Up to 2 Capture, Compare, PWM modules:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Synchronous Serial Port (SSP) with I^2C™

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	LCD (segment drivers)	CCP	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F913	4096	256	256	25	5	16	1	2/1
PIC16F914	4096	256	256	35	8	24	2	2/1
PIC16F916	8192	352	256	25	5	16	1	2/1
PIC16F917	8192	352	256	35	8	24	2	2/1

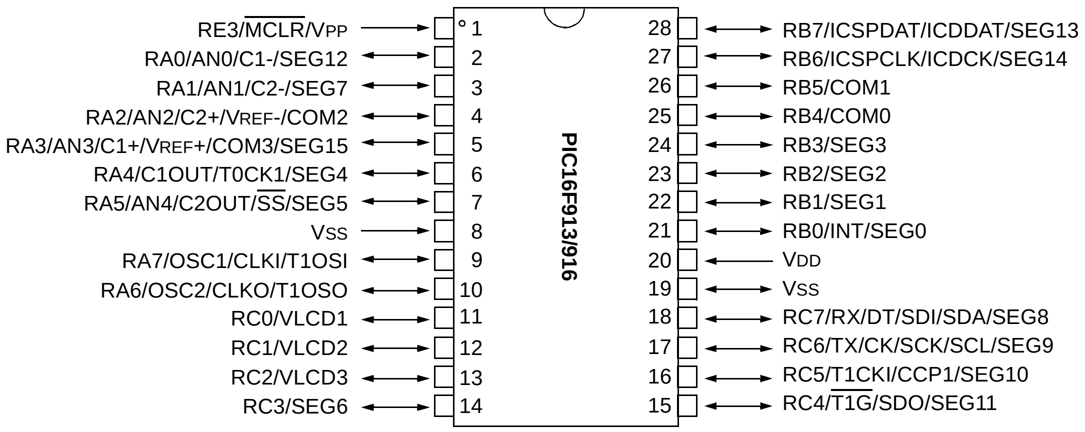
Pin Diagrams – PIC16F914/917, 40-Pin

40-pin PDIP

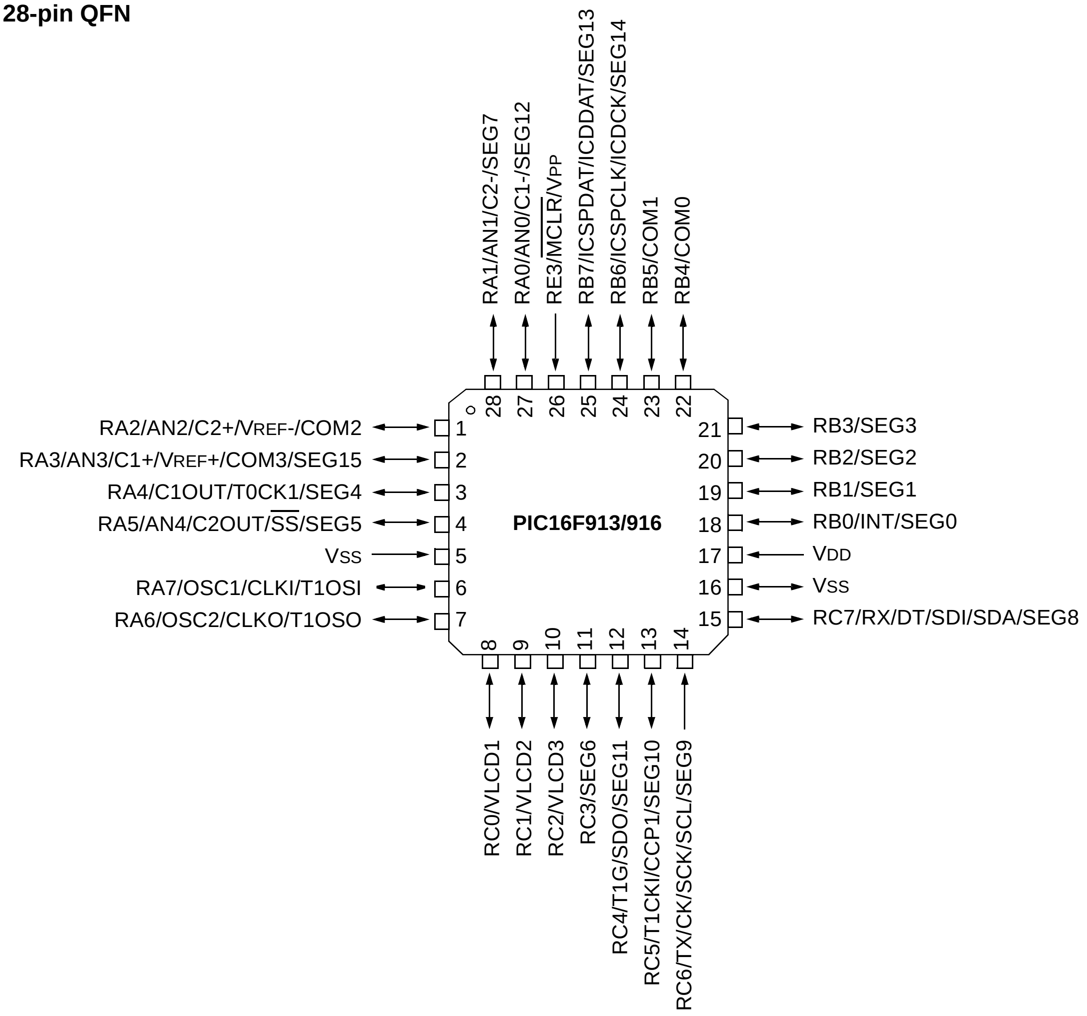


Pin Diagrams – PIC16F913/916, 28-Pin

28-pin PDIP, SOIC, SSOP



28-pin QFN



Pin Diagrams – PIC16F914/917, 44-Pin

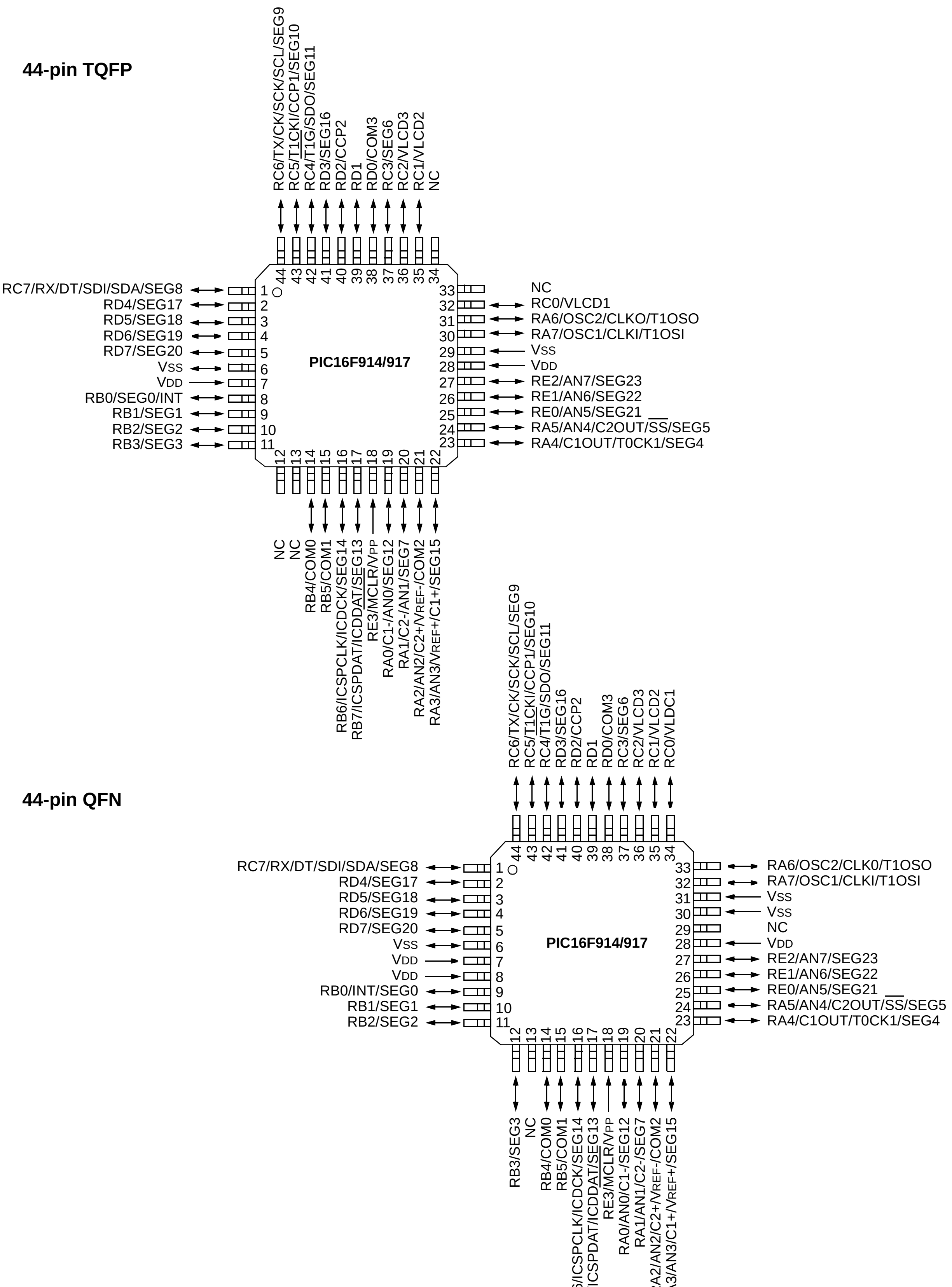


Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

NOTES:

1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC16F91X. Additional information may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023), downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F91X devices are covered by this data sheet. It is available in 28/40/44-pin packages. Figure 1-1 shows a block diagram of the PIC16F913/916 device and Table 1-1 shows the pinout description. Figure 1-2 shows a block diagram of the PIC16F914/917 device and Table 1-1 shows the pinout description.

FIGURE 1-1: PIC16F913/916 BLOCK DIAGRAM

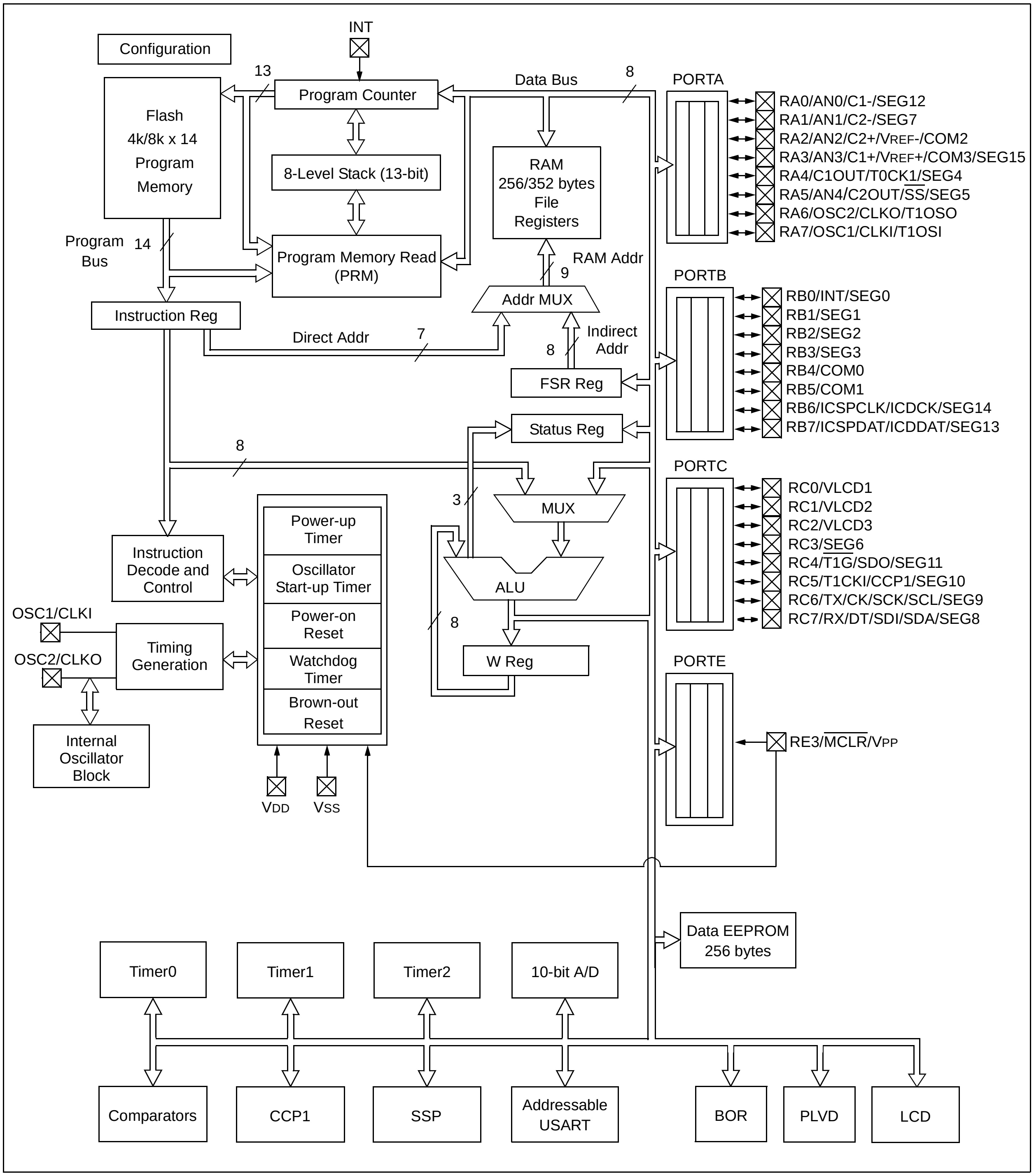


FIGURE 1-2: PIC16F914/917 BLOCK DIAGRAM

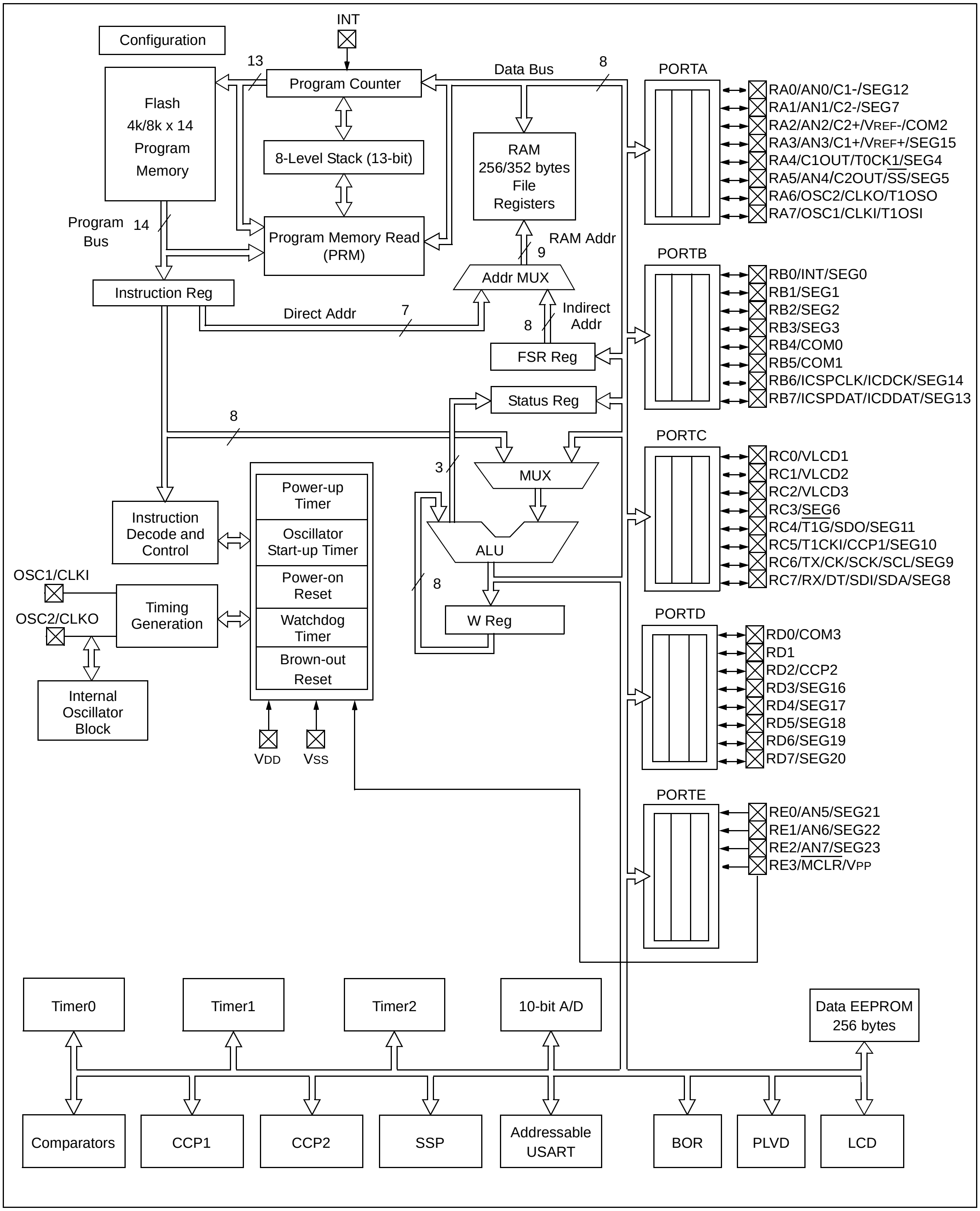


TABLE 1-1: PIC16F91X PINOUT DESCRIPTIONS

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1-/SEG12	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	Analog input Channel 0/Comparator 1 input – negative.
	C1-	—	AN	Comparator 1 negative input.
	SEG12	—	AN	LCD analog output.
RA1/AN1/C2-/SEG7	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	—	Analog input Channel 1/Comparator 2 input – negative.
	C2-	—	AN	Comparator 2 negative input.
	SEG7	—	AN	LCD analog output.
RA2/AN2/C2+/VREF-/COM2	RA2	TTL	CMOS	General purpose I/O.
	AN2	AN	—	Analog input Channel 2/Comparator 2 input – positive.
	C2+	—	AN	Comparator 2 positive input.
	VREF-	AN	—	External Voltage Reference – negative.
	COM2	—	AN	LCD analog output.
RA3/AN3/C1+/VREF+/COM3(1)/ SEG15	RA3	TTL	CMOS	General purpose I/O.
	AN3	AN	—	Analog input Channel 3/Comparator 1 input – positive.
	C1+	—	AN	Comparator 1 positive input.
	VREF+	AN	—	External Voltage Reference – positive.
	COM3(1)	—	AN	LCD analog output.
	SEG15	—	AN	LCD analog output.
RA4/C1OUT/T0CK1/SEG4	RA4	TTL	CMOS	General purpose I/O.
	C1OUT	—	CMOS	Comparator 1 output.
	T0CKI	ST	—	Timer0 clock input.
	SEG4	—	AN	LCD analog output.
RA5/AN4/C2OUT/$\overline{\text{SS}}$/SEG5	RA5	TTL	CMOS	General purpose I/O.
	AN4	AN	—	Analog input Channel 4.
	C2OUT	—	CMOS	Comparator 2 output.
	$\overline{\text{SS}}$	TTL	—	Slave select input.
	SEG5	—	AN	LCD analog output.
RA6/OSC2/CLKO/T1OSO	RA6	TTL	CMOS	General purpose I/O.
	OSC2	—	XTAL	Crystal/Resonator.
	CLKO	—	CMOS	TOSC/4 reference clock.
	T1OSO	—	XTAL	Timer1 oscillator output.
RA7/OSC1/CLKI/T1OSI	RA7	TTL	CMOS	General purpose I/O.
	OSC1	XTAL	—	Crystal/Resonator.
	CLKI	ST	—	Clock input.
	T1OSI	XTAL	—	Timer1 oscillator input.
RB0/INT/SEG0	RB0	TTL	CMOS	General purpose I/O. Individually enabled pull-up.
	INT	ST	—	External interrupt pin.
	SEG0	—	AN	LCD analog output.
RB1/SEG1	RB1	TTL	CMOS	General purpose I/O. Individually enabled pull-up.
	SEG1	—	AN	LCD analog output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output D = Direct
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

Note 1: COM3 is available on RA3 for the PIC16F913/916 and on RD0 for the PIC16F914/917.
2: Pins available on PIC16F914/917 only.

TABLE 1-1: PIC16F91X PINOUT DESCRIPTIONS (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB2/SEG2	RB2	TTL	CMOS	General purpose I/O. Individually enabled pull-up.
	SEG2	—	AN	LCD analog output.
RB3/SEG3	RB3	TTL	CMOS	General purpose I/O. Individually enabled pull-up.
	SEG3	—	AN	LCD analog output.
RB4/COM0	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	COM0	—	AN	LCD analog output.
RB5/COM1	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	COM1	—	AN	LCD analog output.
RB6/ICSPCLK/ICDCK/SEG14	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPCLK	ST	—	ICSP™ clock.
	ICDCK	ST	—	ICD clock I/O.
	SEG14	—	AN	LCD analog output.
RB7/ICSPDAT/ICDDAT/SEG13	RB7	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPDAT	ST	CMOS	ICSP Data I/O.
	ICDDAT	ST	CMOS	ICD Data I/O.
	SEG13	—	AN	LCD analog output.
RC0/VLCD1	RC0	TTL	CMOS	General purpose I/O.
	VLCD1	AN	—	LCD analog input.
RC1/VLCD2	RC1	ST	CMOS	General purpose I/O.
	VLCD2	AN	—	LCD analog input.
RC2/VLCD3	RC2	ST	CMOS	General purpose I/O.
	VLCD3	AN	—	LCD analog input.
RC3/SEG6	RC3	ST	CMOS	General purpose I/O.
	SEG6	—	AN	LCD analog output.
RC4/$\overline{\text{T1G}}$/SDO/SEG11	RC4	ST	CMOS	General purpose I/O.
	$\overline{\text{T1G}}$	ST	—	Timer1 gate input.
	SDO	—	CMOS	Serial data output.
	SEG11	—	AN	LCD analog output.
RC5/T1CKI/CCP1/SEG10	RC5	ST	CMOS	General purpose I/O.
	T1CKI	ST	—	Timer1 clock input.
	CCP1	ST	CMOS	Capture 1 input/Compare 1 output/PWM 1 output.
	SEG10	—	AN	LCD analog output.
RC6/TX/CK/SCK/SCL/SEG9	RC6	ST	CMOS	General purpose I/O.
	TX	—	CMOS	USART asynchronous serial transmit.
	CK	ST	CMOS	USART synchronous serial clock.
	SCK	ST	CMOS	SPI™ clock.
	SCL	ST	CMOS	I^{2}C™ clock.
	SEG9	—	AN	LCD analog output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output D = Direct
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

Note 1: COM3 is available on RA3 for the PIC16F913/916 and on RD0 for the PIC16F914/917.
2: Pins available on PIC16F914/917 only.

TABLE 1-1: PIC16F91X PINOUT DESCRIPTIONS (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC7/RX/DT/SDI/SDA/SEG8	RC7	ST	CMOS	General purpose I/O.
	RX	ST	—	USART asynchronous serial receive.
	DT	ST	CMOS	USART synchronous serial data.
	SDI	ST	CMOS	SPI™ data input.
	SDA	ST	CMOS	I^2C™ data.
	SEG8	—	AN	LCD analog output.
RD0/COM3(1, 2)	RD0	ST	CMOS	General purpose I/O.
	COM3	—	AN	LCD analog output.
RD1(2)	RD1	ST	CMOS	General purpose I/O.
RD2/CCP2(2)	RD2	ST	CMOS	General purpose I/O.
	CCP2	ST	CMOS	Capture 2 input/Compare 2 output/PWM 2 output.
RD3/SEG16(2)	RD3	ST	CMOS	General purpose I/O.
	SEG16	—	AN	LCD analog output.
RD4/SEG17(2)	RD4	ST	CMOS	General purpose I/O.
	SEG17	—	AN	LCD analog output.
RD5/SEG18(2)	RD5	ST	CMOS	General purpose I/O.
	SEG18	—	AN	LCD analog output.
RD6/SEG19(2)	RD6	ST	CMOS	General purpose I/O.
	SEG19	—	AN	LCD analog output.
RD7/SEG20(2)	RD7	ST	CMOS	General purpose I/O.
	SEG20	—	AN	LCD analog output.
RE0/AN5/SEG21(2)	RE0	ST	CMOS	General purpose I/O.
	AN5	AN	—	Analog input Channel 5.
	SEG21	—	AN	LCD analog output.
RE1/AN6/SEG22(2)	RE1	ST	CMOS	General purpose I/O.
	AN6	AN	—	Analog input Channel 6.
	SEG22	—	AN	LCD analog output.
RE2/AN7/SEG23(2)	RE2	ST	CMOS	General purpose I/O.
	AN7	AN	—	Analog input Channel 7.
	SEG23	—	AN	LCD analog output.
RE3/$\overline{\text{MCLR}}$/VPP	RE3	ST	—	Digital input only.
	$\overline{\text{MCLR}}$	ST	—	Master Clear w/internal pull-up.
	VPP	HV	—	Programming voltage.
VDD	VDD	D	—	Power supply for microcontroller.
VSS	VSS	D	—	Ground reference for microcontroller.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output D = Direct
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

Note 1: COM3 is available on RA3 for the PIC16F913/916 and on RD0 for the PIC16F914/917.
2: Pins available on PIC16F914/917 only.

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16F91X has a 13-bit program counter capable of addressing a 4k x 14 program memory space for the PIC16F913/914 (0000h-0FFFh) and an 8k x 14 program memory space for the PIC16F916/917 (0000h-1FFFh). Accessing a location above the memory boundaries for the PIC16F913 and PIC16F914 will cause a wrap around within the first 4k x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F913/914

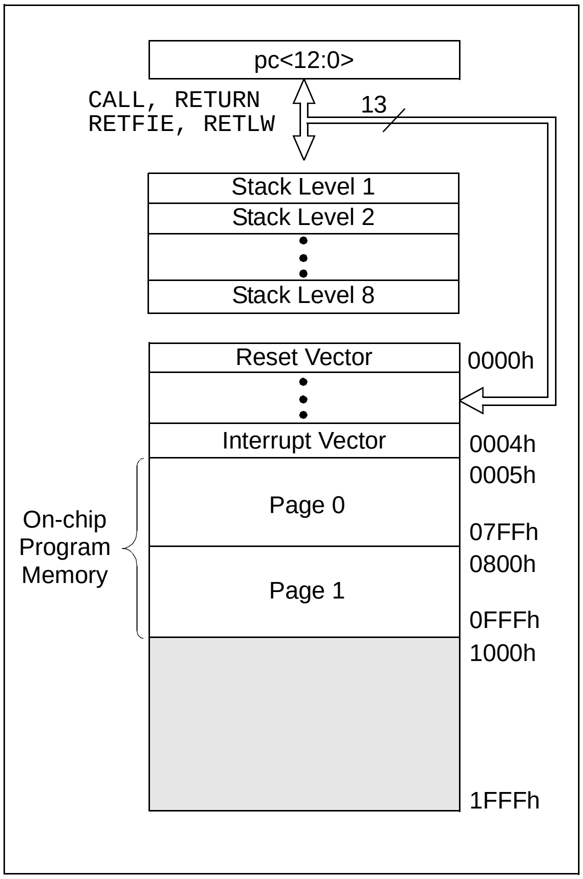
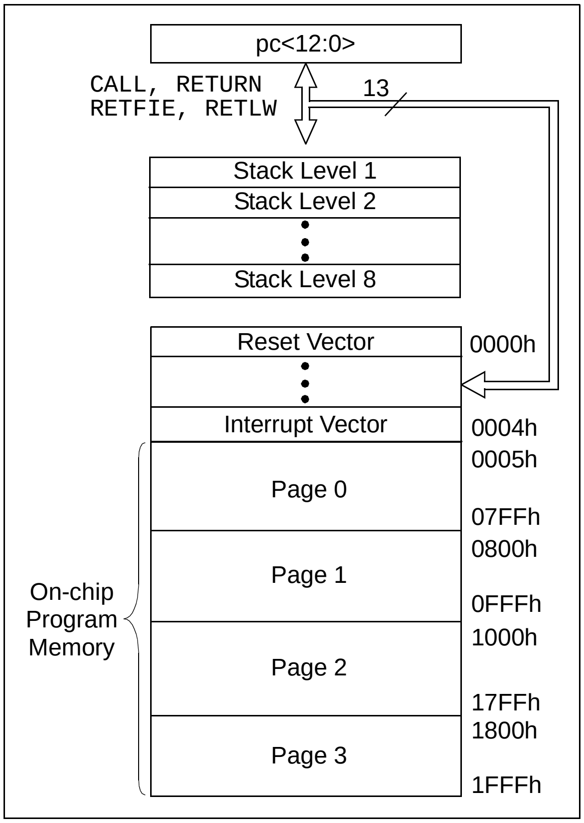


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F916/917



2.2 Data Memory Organization

The data memory is partitioned into multiple banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bits RP0 and RP1 are bank select bits.

RP0	RP1	(Status<6:5>)

= 00: → Bank 0

= 01: → Bank 1

= 10: → Bank 2

= 11: → Bank 3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank are mirrored in another bank for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 256 x 8 in the PIC16F913/914 and 352 x 8 in the PIC16F916/917. Each register is accessed either directly or indirectly through the File Select Register (FSR) (see **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Tables 2-1, 2-2, 2-3 and 2-4). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

FIGURE 2-3: PIC16F913/916 SPECIAL FUNCTION REGISTERS

	File Address		File Address		File Address		File Address
Indirect addr.[1]	00h	Indirect addr.[1]	80h	Indirect addr.[1]	100h	Indirect addr.[1]	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h	▨	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	LCDCON	107h	▨	187h
▨	08h	▨	88h	LCDPS	108h	▨	188h
PORTE	09h	TRISE	89h	LVDCON	109h	▨	189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATL	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADRL	10Dh	EECON2[1]	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	▨	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	▨	18Fh
T1CON	10h	OSCTUNE	90h	LCDDATA0	110h	General Purpose Register[2] 96 Bytes	190h
TMR2	11h	ANSEL	91h	LCDDATA1	111h		
T2CON	12h	PR2	92h	▨	112h		
SSPBUF	13h	SSPADD	93h	LCDDATA3	113h		
SSPCON	14h	SSPSTAT	94h	LCDDATA4	114h		
CCPR1L	15h	WPUB	95h	▨	115h		
CCPR1H	16h	IOCB	96h	LCDDATA6	116h		
CCP1CON	17h	CMCON1	97h	LCDDATA7	117h		
RCSTA	18h	TXSTA	98h	▨	118h		
TXREG	19h	SPBRG	99h	LCDDATA9	119h		
RCREG	1Ah	▨	9Ah	LCDDATA10	11Ah		
▨	1Bh	▨	9Bh	▨	11Bh		
▨	1Ch	CMCON0	9Ch	LCDSE0	11Ch		
▨	1Dh	VRCON	9Dh	LCDSE1	11Dh		
ADRESH	1Eh	ADRESL	9Eh	▨	11Eh		
ADCON0	1Fh	ADCON1	9Fh	▨	11Fh		
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h		
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank1		Bank2		Bank3	

▨ Unimplemented data memory locations, read as ‘0’.

Note 1: Not a physical register.

2: On the PIC16F913, unimplemented data memory locations, read as ‘0’.

FIGURE 2-4: PIC16F914/917 SPECIAL FUNCTION REGISTERS

	File Address		File Address		File Address		File Address
Indirect addr. [1]	00h	Indirect addr. [1]	80h	Indirect addr. [1]	100h	Indirect addr. [1]	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h	▨	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	LCDCON	107h	▨	187h
PORTD	08h	TRISD	88h	LCDPS	108h	▨	188h
PORTE	09h	TRISE	89h	LVDCON	109h	▨	189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATL	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADRL	10Dh	EECON2[1]	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	▨	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	▨	18Fh
T1CON	10h	OSCTUNE	90h	LCDDATA0	110h	General Purpose Register[2] 96 Bytes	190h
TMR2	11h	ANSEL	91h	LCDDATA1	111h		
T2CON	12h	PR2	92h	LCDDATA2	112h		
SSPBUF	13h	SSPADD	93h	LCDDATA3	113h		
SSPCON	14h	SSPSTAT	94h	LCDDATA4	114h		
CCPR2L	15h	WPUB	95h	LCDDATA5	115h		
CCPR2H	16h	IOCB	96h	LCDDATA6	116h		
CCP2CON	17h	CMCON1	97h	LCDDATA7	117h		
RCSTA	18h	TXSTA	98h	LCDDATA8	118h		
TXREG	19h	SPBRG	99h	LCDDATA9	119h		
RCREG	1Ah	▨	9Ah	LCDDATA10	11Ah		
CCPR2L	1Bh	▨	9Bh	LCDDATA11	11Bh		
CCPR2H	1Ch	CMCON0	9Ch	LCDSE0	11Ch		
CCPR2CON	1Dh	VRCON	9Dh	LCDSE1	11Dh		
ADRESH	1Eh	ADRESL	9Eh	LCDSE2	11Eh		
ADCON0	1Fh	ADCON1	9Fh	▨	11Fh		
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h		
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank1		Bank2		Bank3	

▨ Unimplemented data memory locations, read as ‘0’.

Note 1: Not a physical register.

2: On the PIC16F914, unimplemented data memory locations, read as ‘0’.

TABLE 2-1: PIC16F91X SPECIAL REGISTERS SUMMARY BANK 0

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets[(1)]
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								`xxxx xxxx`	`xxxx xxxx`
01h	TMR0	Timer0 Module's register								`xxxx xxxx`	`uuuu uuuu`
02h	PCL	Program Counter's (PC) Least Significant Byte								`0000 0000`	`0000 0000`
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	`0001 1xxx`	`000q quuu`
04h	FSR	Indirect Data Memory Address Pointer								`xxxx xxxx`	`uuuu uuuu`
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	`xxxx xxxx`	`uuuu uuuu`
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	`uuuu uuuu`
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	`xxxx xxxx`	`uuuu uuuu`
08h	PORTD[(2)]	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	`xxxx xxxx`	`uuuu uuuu`
09h	PORTE	—	—	—	—	RE3	RE2[(2)]	RE1[(2)]	RE0[(2)]	`---- xxxx`	`---- uuuu`
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					`---0 0000`	`---0 0000`
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000x`
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	`0000 -0-0`	`0000 -0-0`
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								`xxxx xxxx`	`uuuu uuuu`
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								`xxxx xxxx`	`uuuu uuuu`
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	`0000 0000`	`uuuu uuuu`
11h	TMR2	Timer2 Module Register								`0000 0000`	`0000 0000`
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	`-000 0000`	`-000 0000`
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								`xxxx xxxx`	`uuuu uuuu`
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	`0000 0000`	`0000 0000`
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								`xxxx xxxx`	`uuuu uuuu`
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								`xxxx xxxx`	`uuuu uuuu`
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	`--00 0000`	`--00 0000`
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 000x`	`0000 000x`
19h	TXREG	USART Transmit Data Register								`0000 0000`	`0000 0000`
1Ah	RCREG	USART Receive Data Register								`0000 0000`	`0000 0000`
1Bh[(2)]	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								`xxxx xxxx`	`uuuu uuuu`
1Ch[(2)]	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								`xxxx xxxx`	`uuuu uuuu`
1Dh[(2)]	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	`--00 0000`	`--00 0000`
1Eh	ADRESH	A/D Result Register High Byte								`xxxx xxxx`	`uuuu uuuu`
1Fh	ADCON0	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	`0000 0000`	`0000 0000`

Legend: — = Unimplemented locations read as '`0`', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: PIC16F914/917 only.

TABLE 2-2: PIC16F91X SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets[(1)]
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
88h	TRISD[(2)]	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111
89h	TRISE	—	—	—	—	TRISE3[(5)]	TRISE2[(2)]	TRISE1[(2)]	TRISE0[(2)]	---- 1111	---- 1111
8Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
8Eh	PCON	—	—	—	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	---1 --qq	---u --uu
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS[(4)]	HTS	LTS	SCS	-110 q000	-110 x000
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
91h	ANSEL	ANS7[(3)]	ANS6[(3)]	ANS5[(3)]	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	0000 0000
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111
96h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
97h	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	SPBRG7	SPBRG6	SPBRG5	SPBRG4	SPBRG3	SPBRG2	SPBRG1	SPBRG0	0000 0000	0000 0000
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
9Dh	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ---

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

2: PIC16F914/917 only.

3: PIC16F914/917 only, forced '0' on PIC16F913/916.

4: The value of the OSTS bit is dependent on the value of the Configuration Word (CONFIG) of the device. See **Section 3.0 "Clock Sources"**.

5: Bit is read-only; TRISE = 1 always.

TABLE 2-3: PIC16F91X SPECIAL REGISTERS SUMMARY BANK 2

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets[1]
Bank 2											
100h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 Module’s register								xxxx xxxx	uuuu uuuu
102h	PCL	Program Counter’s (PC) Least Significant Byte								0000 0000	0000 0000
103h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
104h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000
106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	0001 0011	0001 0011
108h	LCDPS	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0	0000 0000	0000 0000
109h	LVDCON	—	—	IRVST	LVDEN	—	LVDL2	LVDL1	LVDL0	--00 -100	--00 -100
10Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
10Ch	EEDATL	EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0	0000 0000	0000 0000
10Dh	EEADRL	EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0	0000 0000	0000 0000
10Eh	EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
10Fh	EEADRH	—	—	—	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---0 0000	---0 0000
110h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	xxxx xxxx	uuuu uuuu
111h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	xxxx xxxx	uuuu uuuu
112h	LCDDATA2[2]	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	xxxx xxxx	uuuu uuuu
113h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	xxxx xxxx	uuuu uuuu
114h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	xxxx xxxx	uuuu uuuu
115h	LCDDATA5[2]	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	xxxx xxxx	uuuu uuuu
116h	LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	xxxx xxxx	uuuu uuuu
117h	LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	xxxx xxxx	uuuu uuuu
118h	LCDDATA8[2]	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	xxxx xxxx	uuuu uuuu
119h	LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	xxxx xxxx	uuuu uuuu
11Ah	LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	xxxx xxxx	uuuu uuuu
11Bh	LCDDATA11[2]	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	xxxx xxxx	uuuu uuuu
11Ch	LCDSE0[3]	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu
11Dh	LCDSE1[3]	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu
11Eh	LCDSE2[2,3]	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	0000 0000	uuuu uuuu
11Fh	—	Unimplemented								—	—

Legend: — = Unimplemented locations read as ‘0’, u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

2: PIC16F914/917 only.

3: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

TABLE 2-4: PIC16F91X SPECIAL FUNCTION REGISTERS SUMMARY BANK 3

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets[1]
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
183h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
184h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	---- ----

Legend: — = Unimplemented locations read as ‘0’, u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

2.2.2.1 Status Register

The Status register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The Status register can be the destination for any instruction, like any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the Status register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (see **Section 17.0 "Instruction Set Summary"**).

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

REGISTER 2-1: STATUS – STATUS REGISTER (ADDRESS: 03h, 83h, 103h, OR 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h-7Fh)
01 = Bank 1 (80h-FFh)
10 = Bank 2 (100h-17Fh)
11 = Bank 3 (180h-1FFh)

bit 4 **$\overline{\text{TO}}$:** Time-out bit
1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$**: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

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 (`ADDWF`, `ADDLW`,`SUBLW`,`SUBWF` instructions)[(1)]
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

bit 0 **C**: Carry/$\overline{\text{Borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)[(1)]
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 occurred

Note 1: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

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

2.2.2.2 Option Register

The Option register is a readable and writable register, which contains various control bits to configure:

- TMR0/WDT prescaler
- External RB0/INT interrupt
- TMR0
- Weak pull-ups on PORTB

Note: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT by setting PSA bit to ‘1’ (OPTION_REG<3>). See **Section 5.4 “Prescaler”**.

REGISTER 2-2: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7 bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit

1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG:** Interrupt Edge Select bit

1 = Interrupt on rising edge of RB0/INT/SEG0 pin
0 = Interrupt on falling edge of RB0/INT/SEG0 pin

bit 5 **T0CS:** TMR0 Clock Source Select bit

1 = Transition on RA4/C1OUT/T0CK1/SEG4 pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE:** TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on RA4/C1OUT/T0CK1/SEG4 pin
0 = Increment on low-to-high transition on RA4/C1OUT/T0CK1/SEG4 pin

bit 3 **PSA:** Prescaler Assignment bit

1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

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

2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTB change and external RB0/INT/SEG0 pin interrupts.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON – INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, or 18Bh)

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	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE:** RB0/INT/SEG0 External Interrupt Enable bit
1 = Enables the RB0/INT/SEG0 external interrupt
0 = Disables the RB0/INT/SEF0 external interrupt

bit 3 **RBIE:** PORTB Change Interrupt Enable bit**(1)**
1 = Enables the PORTB change interrupt
0 = Disables the PORTB change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit**(2)**
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT/SEG0 External Interrupt Flag bit
1 = The RB0/INT/SEG0 external interrupt occurred (must be cleared in software)
0 = The RB0/INT/SEG0 external interrupt did not occur

bit 0 **RBIF:** PORTB Change Interrupt Flag bit
1 = When at least one of the PORTB <5:0> pins changed state (must be cleared in software)
0 = None of the PORTB <7:4> pins have changed state

Note 1: IOCB register must also be enabled.

2: T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

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

2.2.2.4 PIE1 Register

The PIE1 register contains the interrupt enable bits, as shown in Register 2-1.

> **Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1 – PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **EEIE:** EE Write Complete Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **SSPIE:** Synchronous Serial Port (SSP) Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled

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

2.2.2.5 PIE2 Register

The PIE2 register contains the interrupt enable bits, as shown in Register 2-5.

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-5: PIE2 – PERIPHERAL INTERRUPT ENABLE REGISTER 2 (ADDRESS: 8Dh)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0
OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE
bit 7							bit 0

bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **C2IE:** Comparator 2 Interrupt Enable bit
1 = Enables Comparator 2 interrupt
0 = Disables Comparator 2 interrupt

bit 5 **C1IE:** Comparator 1 Interrupt Enable bit
1 = Enables Comparator 1 interrupt
0 = Disables Comparator 1 interrupt

bit 4 **LCDIE:** LCD Module Interrupt Enable bit
1 = LCD interrupt is enabled
0 = LCD interrupt is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **LVDIE:** Low Voltage Detect Interrupt Enable bit
1 = Enables LVD Interrupt
0 = Disables LVD Interrupt

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit (only available in 16F914/917)
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

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

2.2.2.6 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-6.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-6: PIR1 – PERIPHERAL INTERRUPT REQUEST REGISTER 1 (ADDRESS: 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **EEIF:** EE Write Operation Interrupt Flag bit

1 = The write operation completed (must be cleared in software)
0 = The write operation has not completed or has not started

bit 6 **ADIF:** A/D Converter Interrupt Flag bit

1 = The A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RCIF:** USART Receive Interrupt Flag bit

1 = The USART receive buffer is full (cleared by reading RCREG)
0 = The USART receive buffer is not full

bit 4 **TXIF:** USART Transmit Interrupt Flag bit

1 = The USART transmit buffer is empty (cleared by writing to TXREG)
0 = The USART transmit buffer is full

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit

1 = The Transmission/Reception is complete (must be cleared in software)
0 = Waiting to Transmit/Receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit

Capture Mode

1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare Mode

1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM mode

Unused in this mode

bit 1 **TMR2IF:** TMR2 to PR2 Interrupt Flag bit

1 = A TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit

1 = The TMR1 register overflowed (must be cleared in software)
0 = The TMR1 register did not overflow

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

2.2.2.7 PIR2 Register

The PIR2 register contains the interrupt flag bits, as shown in Register 2-7.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-7: PIR2 – PERIPHERAL INTERRUPT REQUEST REGISTER 2 (ADDRESS: 0Dh)

R/W-0	R/W-0	R-0	R-0	U-0	R/W-0	U-0	R/W-0
OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF
bit 7							bit 0

bit 7 **OSFIF:** Oscillator Fail Interrupt Flag bit

1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = System clock operating

bit 6 **C2IF:** Comparator 2 Interrupt Flag bit

1 = Comparator output (C2OUT bit) has changed (must be cleared in software)
0 = Comparator output (C2OUT bit) has not changed

bit 5 **C1IF:** Comparator 1 Interrupt Flag bit

1 = Comparator output (C1OUT bit) has changed (must be cleared in software)
0 = Comparator output (C1OUT bit) has not changed

bit 4 **LCDIF:** LCD Module Interrupt bit

1 = LCD has generated an interrupt
0 = LCD has not generated an interrupt

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **LVDIF:** Low Voltage Detect Interrupt Flag bit

1 = LVD has generated an interrupt
0 = LVD has not generated an interrupt

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit (only available in 16F914/917)

Capture Mode

1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare Mode

1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM mode

Unused in this mode

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

2.2.2.8 PCON Register

The Power Control (PCON) register (See Table 17-2) contains flag bits to differentiate between a:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Reset ($\overline{\text{BOR}}$)
- Watchdog Timer Reset (WDT)
- External $\overline{\text{MCLR}}$ Reset

The PCON register also controls the software enable of the BOR.

The PCON register bits are shown in Register 2-8.

REGISTER 2-8: PCON – POWER CONTROL REGISTER (ADDRESS: 8Eh)

U-0	U-0	U-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	—	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **SBOREN**: Software BOR Enable bit[1]

1 = BOR enabled
0 = BOR disabled

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit

1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

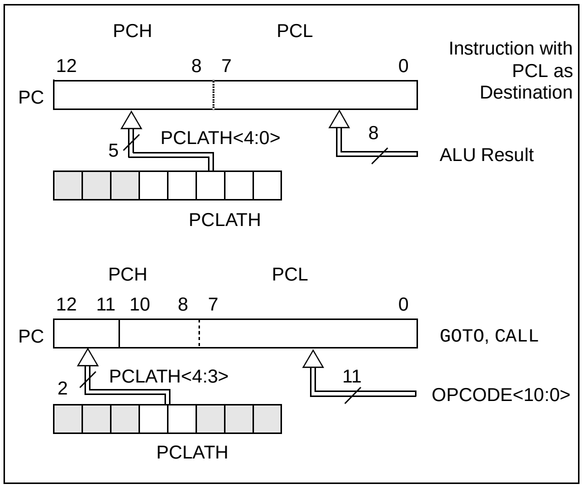
Note 1: BOREN<1:0> = 01 in the Configuration Word register for this bit to control the $\overline{\text{BOR}}$.

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

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-5 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, *"Implementing a Table Read"* (DS00556).

2.3.2 STACK

The PIC16F91X family has an 8-level x 13-bit wide hardware stack (see Figures 2-1and 2-2). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1:	There are no Status bits to indicate stack overflow or stack underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Program Memory Paging

All PIC16F91X devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note:	The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
        ORG 0x500
        BCF PCLATH,4
        BSF PCLATH,3  ;Select page 1
                      ;(800h-FFFh)
        CALL SUB1_P1  ;Call subroutine in
        :             ;page 1 (800h-FFFh)
        :
        ORG 0x900     ;page 1 (800h-FFFh)
SUB1_P1
        :             ;called subroutine
                      ;page 1 (800h-FFFh)
        :
        RETURN        ;return to
                      ;Call subroutine
                      ;in page 0
                      ;(000h-7FFh)
```

2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

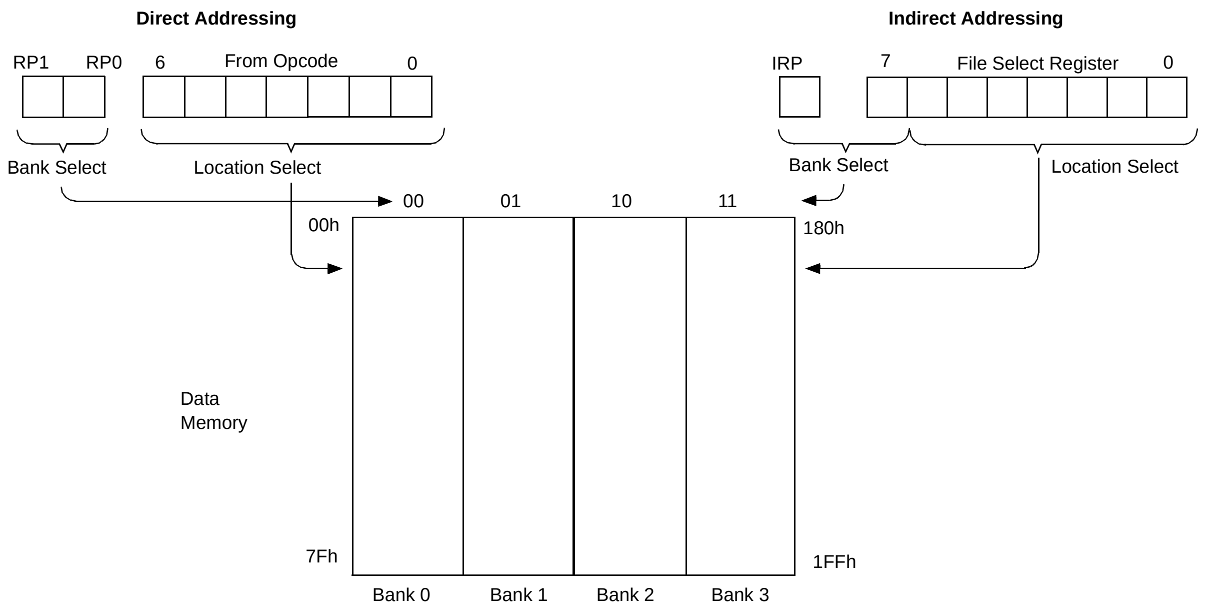
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>), as shown in Figure 2-6.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
    MOVLW   0x20    ;initialize pointer
    MOVWF   FSR     ;to RAM
NEXTCLRF    INDF    ;clear INDF register
    INCF    FSR     ;inc pointer
    BTFSS   FSR,4   ;all done?
    GOTO    NEXT    ;no clear next
CONTINUE            ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING PIC16F91X



Note: For memory map detail, see Figures 2-3 and 2-4.

3.0 CLOCK SOURCES

3.1 Overview

The PIC16F91X has a wide variety of clock sources and selection features to allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 3-1 illustrates a block diagram of the PIC16F91X clock sources.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators, and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

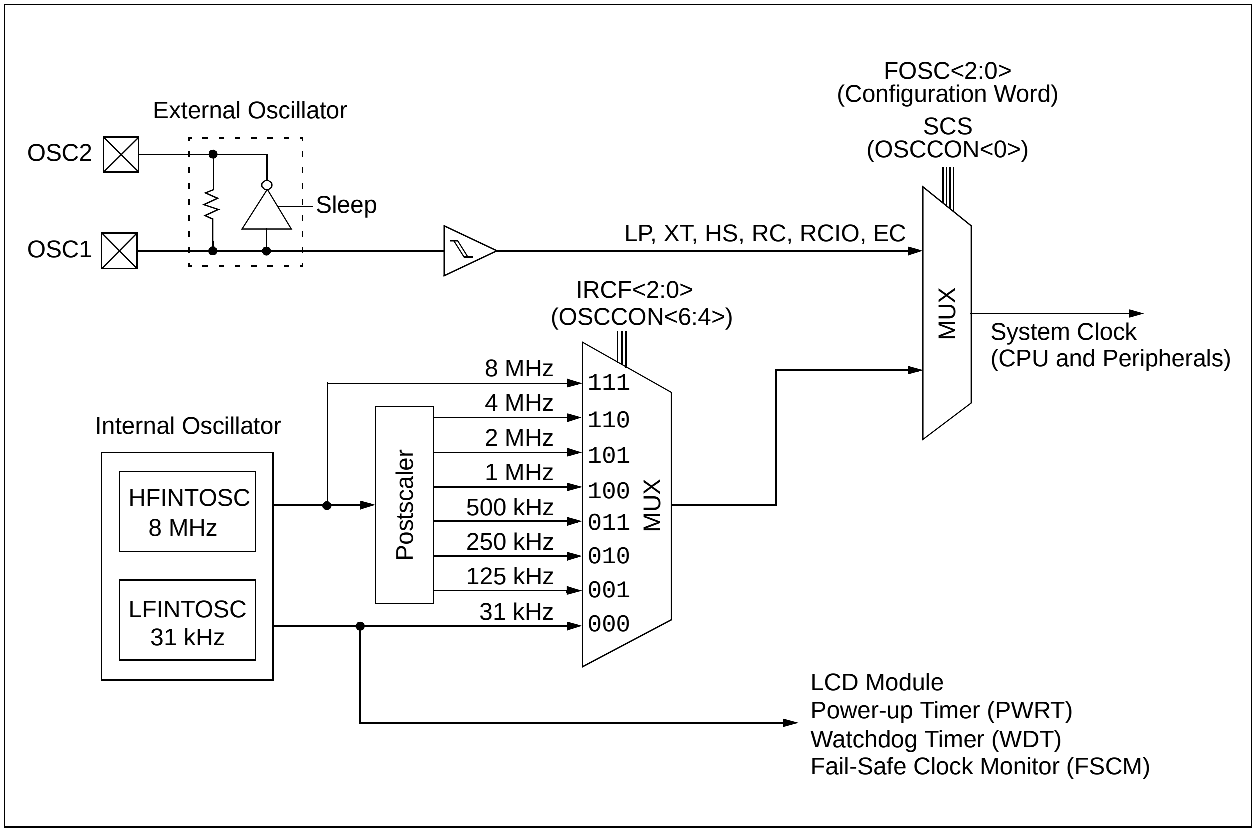
- Selectable system clock source between external or internal via software.
- Two-Speed Clock Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch to the Internal Oscillator.

The PIC16F91X can be configured in one of eight clock modes.

1. EC – External clock with I/O on RA6.
2. LP – Low-gain Crystal or Ceramic Resonator Oscillator mode.
3. XT – Medium-gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High-gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with FOSC/4 output on RA6.
6. RCIO – External Resistor-Capacitor with I/O on RA6.
7. INTOSC – Internal oscillator with FOSC/4 output on RA6 and I/O on RA7.
8. INTRCIO – Internal oscillator with I/O on RA6 and RA7.

Clock source modes are configured by the FOSC<2:0> bits in the Configuration Word register (see **Section 16.0 “Special Features of the CPU”**). The internal clock can be generated by two oscillators. The HFINTOSC is a high-frequency calibrated oscillator. The LFINTOSC is a low-frequency uncalibrated oscillator.

FIGURE 3-1: PIC16F91X SYSTEM CLOCK BLOCK DIAGRAM



REGISTER 3-1: OSCCON – OSCILLATOR CONTROL REGISTER (ADDRESS: 8Fh)

U-0	R/W-1	R/W-1	R/W-0	R-q	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS[1]	HTS	LTS	SCS
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
000 = 31 kHz
001 = 125 kHz
010 = 250 kHz
011 = 500 kHz
100 = 1 MHz
101 = 2 MHz
110 = 4 MHz
111 = 8 MHz

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit
1 = Device is running from the external system clock defined by FOSC<2:0>
0 = Device is running from the internal system clock (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC (High Frequency – 8 MHz to 125 kHz) Status bit
1 = HFINTOSC is stable
0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC (Low Frequency – 31 kHz) Stable bit
1 = LFINTOSC is stable
0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit
1 = Internal oscillator is used for system clock
0 = Clock source defined by FOSC<2:0>

Note 1: The value of the OSTS bit on device power-up is dependent on the value of the Configuration Word (CONFIG) of the device. The value of the OSTS bit will be ‘0’ on a device Power-on Reset (POR) or any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, if the following conditions are true:
OSTS = 0 if:
FOSC<2:0> = 000 (LP) or 001 (XT) or 010 (HS)
and
IESO = 1 or FSCM = 1
(IESO will be enabled automatically if FSCM is enabled)
If any of the above conditions are not met, the value of the OSTS bit will be ‘1’ on a device POR. See **Section 3.6 “Two-Speed Clock Start-up Mode”** and **Section 3.7 “Fail-Safe Clock Monitor”** for more details.

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
q = value depends on condition			

3.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- External clock modes rely on external circuitry for the clock source. Examples are oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes), and Resistor-Capacitor (RC mode) circuits.
- Internal clock sources are contained internally within the PIC16F91X. The PIC16F91X has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

3.3 External Clock Modes

3.3.1 OSCILLATOR START-UP TIMER (OST)

If the PIC16F91X is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from the OSC1 pin, following a Power-on Reset (POR), and the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the PIC16F91X. When switching between clock sources a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

3.3.1.1 Special Case

An exception to this is when the device is put to Sleep while the following conditions are true:

- LP is the selected primary oscillator mode.
- T1OSCEN = `1` (Timer1 oscillator is enabled).
- SCS = `0` (oscillator mode is defined by FOSC<2:0>).
- OSTS = `1` (device is running from primary system clock).

For this case, the OST is not necessary after a wake-up from Sleep, since Timer1 continues to run during Sleep and uses the same LP oscillator circuit as its clock source. For these devices, this case is typically seen when the LCD module is running during Sleep.

In applications where the OSCTUNE register is used to shift the FINTOSC frequency, the application should not expect the FINTOSC frequency to stabilize immediately. In this case, the frequency may shift gradually toward the new value. The time for this frequency shift is less than eight cycles of the base frequency.

Note:	When the OST is invoked, the WDOG is held in Reset, because the WDOG ripple counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDOG will begin counting (if enabled).

Table 3-1 shows examples where the oscillator delay is invoked.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 3.6 "Two-Speed Clock Start-up Mode"**).

TABLE 3-1: OSCILLATOR DELAY EXAMPLES

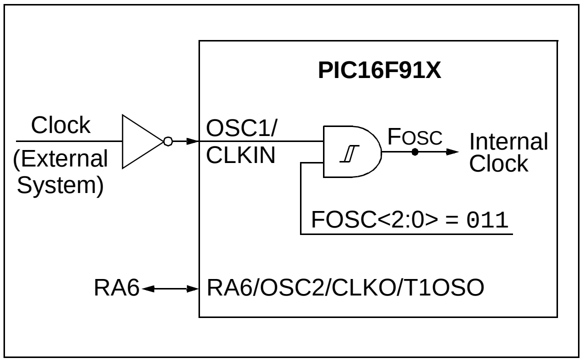
System Clock Source	Frequency	Switching From	Oscillator Delay (TOST)	Comments
LFIOSC	31 kHz	Sleep	10 µs internal delay	Following a wake-up from Sleep mode or POR, an internal delay is invoked to allow the memory bias to stabilize before program execution can begin.
HFIOSC	125 kHz-8 MHz	Sleep	10 µs internal delay	Following a wake-up from Sleep mode or POR, an internal delay is invoked to allow the memory bias to stabilize before program execution can begin.
XT or HS	4-20 MHz	INTOSC or Sleep	1024 clock cycles	Following a change from INTOSC, an OST of 1024 cycles must occur.
LP	32 kHz	INTOSC or Sleep	1024 clock cycles	Following a change from INTOSC, an OST of 1024 cycles must occur. See **Section 3.3.1.1 “Special Case”** for special case conditions.
LP with T1OSC enabled	32 kHz	Sleep	10 µs internal delay	Following a wake-up from Sleep mode, an internal delay is invoked to allow the memory bias to stabilize before program execution can begin. See **Section 3.3.1.1 “Special Case”** for details about this special case.
EC, RC	0-20 MHz	Sleep	10 µs internal delay	Following a wake-up from Sleep mode or POR, an internal delay is invoked to allow the memory bias to stabilize before program execution can begin.
EC, RC	0-20 MHz	LFIOSC	10 µs internal delay	Following a switch from a LFIOSC or POR, an internal delay is invoked to allow the memory bias to stabilize before program execution can begin.

3.3.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 pin and the RA6 pin is available for general purpose I/O. Figure 3-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC16F91X design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 3-2: EXTERNAL CLOCK (EC) MODE OPERATION



3.3.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to the OSC1 and OSC2 pins (Figures 3-3 and 3-4). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

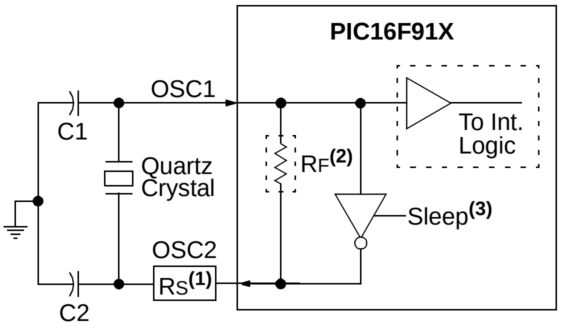
> **Note:** In the past, the sources for the LP oscillator and Timer1 oscillator have been separate circuits. In this family of devices, the LP oscillator and Timer1 oscillator use the same oscillator circuitry. When using a device configured for the LP oscillator and with T1OSCEN = `1`, the source of the clock for each function comes from the same oscillator block.

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification, for example, low-frequency/AT-cut quartz crystal resonators.

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting, for example, high-frequency/AT-cut quartz crystal resonators or ceramic resonators.

Figures 3-3 and 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

FIGURE 3-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

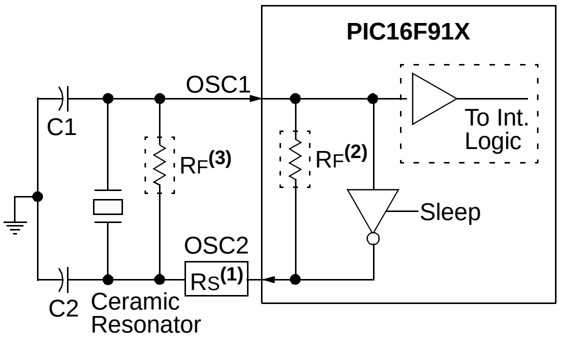
2: The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

3: If using LP mode and T1OSC in enable, the LP oscillator will continue to run during Sleep.

Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

FIGURE 3-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)



Note 1: A series resistor (RS) may be required for ceramic resonators with low drive level.

2: The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

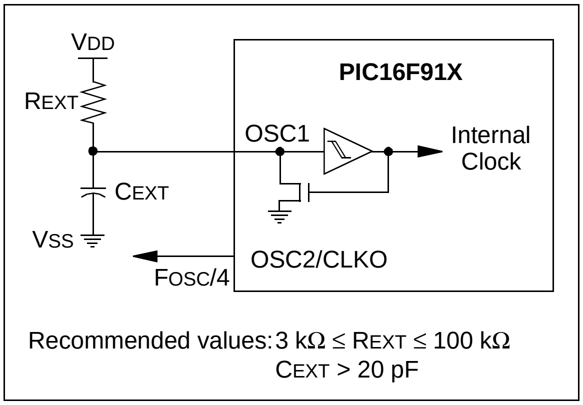
3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation (typical value 1 MΩ).

3.3.4 EXTERNAL RC MODES

The External Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes, RC and RCIO.

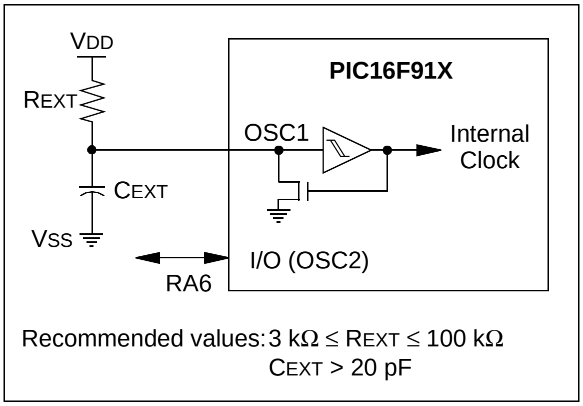
In RC mode, the RC circuit connects to the OSC1 pin. The OSC2/CLKO pin outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the RC mode connections.

FIGURE 3-5: RC MODE



In RCIO mode, the RC circuit is connected to the OSC1 pin. The OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 4 of PORTA (RA4). Figure 3-6 shows the RCIO mode connections.

FIGURE 3-6: RCIO MODE



The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal threshold voltage. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency or for low CEXT values. The user also needs to take into account variation due to tolerance of external RC components used.

3.4 Internal Clock Modes

The PIC16F91X has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user adjusted ±12% via software using the OSCTUNE register (Register 3-2).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at approximately 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select (IRCF) bits.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 “Clock Switching”**).

3.4.1 INTOSC AND INTRCIO MODES

The INTOSC and INTRCIO modes configure the internal oscillators as the system clock source when the device is programmed using the Oscillator Selection (FOSC) bits in the Configuration Word register (Register 16-1).

In **INTOSC** mode, the OSC1 pin is available for general purpose I/O. The OSC2/CLKO pin outputs the selected internal oscillator frequency divided by 4. The CLKO signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In **INTRCIO** mode, the OSC1 and OSC2 pins are available for general purpose I/O.

3.4.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered approximately ±12% via software using the OSCTUNE register (Register 3-2).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the IRCF bits (see **Section 3.4.4 “Frequency Select Bits (IRCF)”**).

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz (IRCF ≠ `000`) as the system clock source (SCS = `1`), or when Two-Speed Start-up is enabled (IESO = `1` and IRCF ≠ `000`).

The HF Internal Oscillator (HTS) bit (OSCCON<2>) indicates whether the HFINTOSC is stable or not.

3.4.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-2).

The OSCTUNE register has a tuning range of ±12%. The default value of the OSCTUNE register is ‘0’. The value is a 5-bit two’s complement number. Due to process variation, the monotonicity and frequency step cannot be specified.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. The HFINTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

REGISTER 3-2: OSCTUNE – OSCILLATOR TUNING RESISTOR (ADDRESS: 90h)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency
01110 =
•
•
•
00001 =
00000 = Center frequency. Oscillator module is running at the calibrated frequency.
11111 =
•
•
•
10000 = Minimum frequency

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

3.4.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated (approximate) 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). 31 kHz can be selected via software using the IRCF bits (see **Section 3.4.4 “Frequency Select Bits (IRCF)”**). The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF = `000`) as the system clock source (SCS = `1`), or when any of the following are enabled:

- Two-Speed Start-up (IESO = `1` and IRCF = `000`)
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)
- Selected as LCD module clock source

The LF Internal Oscillator (LTS) bit (OSCCON<1>) indicates whether the LFINTOSC is stable or not.

3.4.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connect to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency select bits, IRCF<2:0> (OSCCON<6:4>), select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz

Note: Following any Reset, the IRCF bits are set to ‘`110`’ and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency.

3.4.5 HF AND LF INTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power. If this is the case, there is a 10 μs delay after the IRCF bits are modified before the frequency selection takes place. The LTS/HTS bits will reflect the current active status of the LFINTOSC and the HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF bits are modified.
2. If the new clock is shut down, a 10 μs clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKO is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKO is now connected with the new clock. HTS/LTS bits are updated as required.
6. Clock switch is complete.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and the new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

3.5 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit.

3.5.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit (OSCCON<0>) selects the system clock source that is used for the CPU and peripherals.

- When SCS = `0`, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When SCS = `1`, the system clock source is chosen by the internal oscillator frequency selected by the IRCF bits. After a Reset, SCS is always cleared.

Note: Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit. The user can monitor the OSTS (OSCCON<3>) to determine the current system clock source.

3.5.2 OSCILLATOR START-UP TIME-OUT STATUS BIT

The Oscillator Start-up Time-out Status (OSTS) bit (OSCCON<3>) indicates whether the system clock is running from the external clock source, as defined by the FOSC bits, or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

3.6 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

> **Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit (OSCCON<3>) to remain clear.

When the PIC16F91X is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.3.1 "Oscillator Start-up Timer (OST)"**). The OST timer will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit (OSCCON<3>) is set, program execution switches to the external oscillator.

3.6.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO = 1 (CONFIG<10>) Internal/External Switch Over bit.
- SCS = 0.
- FOSC configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after PWRT has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

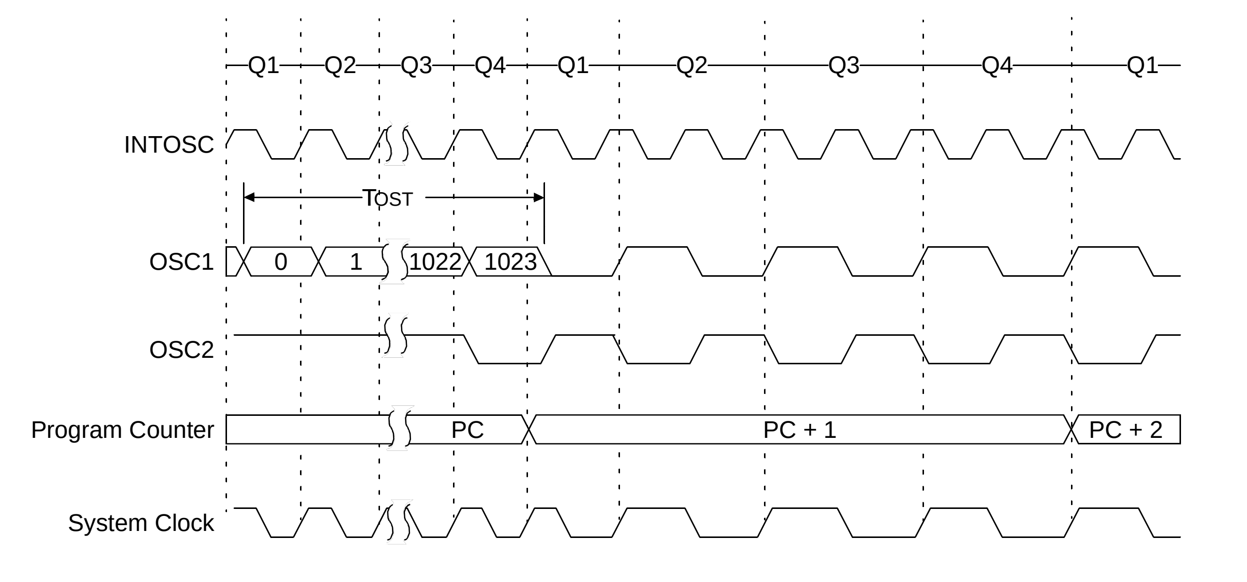
3.6.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF bits (OSCCON<6:4>).
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

3.6.3 CHECKING EXTERNAL/INTERNAL CLOCK STATUS

Checking the state of the OSTS bit (OSCCON<3>) will confirm if the PIC16F91X is running from the external clock source as defined by the FOSC bits in the Configuration Word (CONFIG) or the internal oscillator.

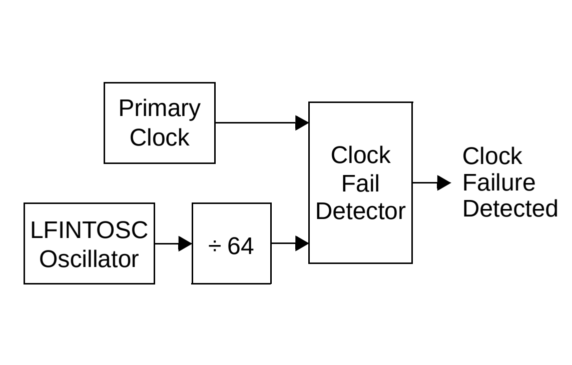
FIGURE 3-7: TWO-SPEED START-UP



3.7 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate in the event of an oscillator failure. The FSCM can detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired.

FIGURE 3-8: FSCM BLOCK DIAGRAM



The FSCM function is enabled by setting the FCMEN bit in the Configuration Word (CONFIG). It is applicable to all external clock options (LP, XT, HS, EC or RC modes).

In the event of an external clock failure, the FSCM will set the OSFIF bit (PIR2<7>) and generate an oscillator fail interrupt if the OSFIE bit (PIE2<7>) is set. The device will then switch the system clock to the internal oscillator. The system clock will continue to come from the internal oscillator unless the external clock recovers and the Fail-Safe condition is exited.

The frequency of the internal oscillator will depend upon the value contained in the IRCF bits (OSCCON<6:4>). Upon entering the Fail-Safe condition, the OSTS bit (OSCCON<3>) is automatically cleared to reflect that the internal oscillator is active and the WDT is cleared. The SCS bit (OSCCON<0>) is not updated. Enabling FSCM does not affect the LTS bit.

The FSCM sample clock is generated by dividing the INTOSC clock by 64. This will allow enough time between FSCM sample clocks for a system clock edge to occur. Figure 3-8 shows the FSCM block diagram.

On the rising edge of the sample clock, a monitoring latch (CM = 0) will be cleared. On a falling edge of the primary system clock, the monitoring latch will be set (CM = 1). In the event that a falling edge of the sample clock occurs, and the monitoring latch is not set, a clock failure has been detected. The assigned internal oscillator is enabled when FSCM is enabled as reflected by the IRCF.

Note 1: Two-Speed Start-up is automatically enabled when the Fail-Safe Clock Monitor mode is enabled.

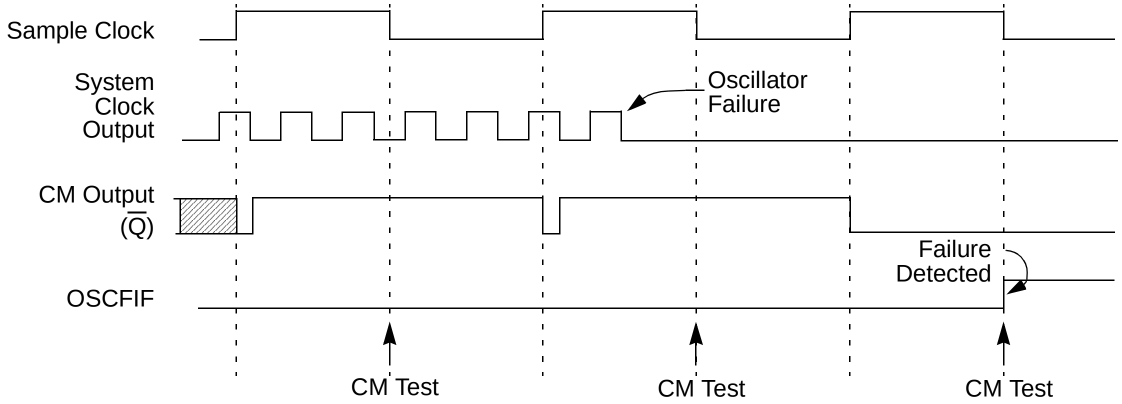
2: Primary clocks with a frequency ≤ ~488 Hz will be considered failed by the FSCM. A slow starting oscillator can cause an FSCM interrupt.

3.7.1 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, the execution of a SLEEP instruction, or a modification of the SCS bit. While in Fail-Safe condition, the PIC16F91X uses the internal oscillator as the system without exiting the Fail-Safe condition.

The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

FIGURE 3-9: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

3.7.2 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired. If the external clock is EC or RC mode, monitoring will begin immediately following these events.

For LP, XT or HS mode the external oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected (see Figure 3-9). To prevent this, the internal oscillator is automatically configured as the system clock and functions until the external clock is stable (the OST has timed out). This is identical to Two-Speed Start-up mode. Once the external oscillator is stable, the LFINTOSC returns to its role as the FSCM source.

.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit (OSCCON<3>) to verify the oscillator start-up and system clock switchover has successfully completed.

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Addr	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
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS[2]	HTS	LTS	SCS	`-110 q000`	`-110 x000`
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	`---0 0000`	`---u uuuu`
2007h[1]	CONFIG	$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0	—	—

Legend: x = unknown, u = unchanged, — = unimplemented locations read as ‘0’. Shaded cells are not used by oscillators.

Note 1: See Register 16-1 for operation of all Configuration Word bits.

2: See Register 3-1 for details.

4.0 I/O PORTS

This device includes four 8-bit port registers along with their corresponding TRIS registers and one four bit port:

- PORTA and TRISA
- PORTB and TRISB
- PORTC and TRISC
- PORTD and TRISD
- PORTE and TRISE

PORTA, PORTB, PORTC and RE3/$\overline{\text{MCLR}}$/V_{PP} are implemented on all devices. PORTD and RE<2:0> are implemented only on the PIC16F914 and PIC16F917.

4.1 PORTA and TRISA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 4-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). Example 4-1 shows how to initialize PORTA.

Five of the pins of PORTA can be configured as analog inputs. These pins, RA5 and RA<3:0>, are configured as analog inputs on device power-up and must be reconfigured by the user to be used as I/O's. This is done by writing the appropriate values to the CMCON0 and ANSEL registers (see Example 4-1).

Reading the PORTA register (Register 4-1) reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the port data latch.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

Note 1: The CMCON0 (9Ch) register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

2: Analog lines that carry LCD signals (i.e., SEGx, COMy, where x and y are segment and common identifiers) are shown as direct connections to the device pins. The signals are outputs from the LCD module and may be tri-stated, depending on the configuration of the LCD module.

EXAMPLE 4-1: INITIALIZING PORTA

```
BCF     STATUS,RP0      ;Bank 0
BCF     STATUS,RP1      ;
CLRF    PORTA           ;Init PORTA
BSF     STATUS,RP0      ;Bank 1
BCF     STATUS,RP1      ;
MOVLW   07h             ;Set RA<2:0> to
MOVWF   CMCON0          ;digital I/O
CLF     ANSEL           ;Make all PORTA I/O
MOVLW   F0h             ;Set RA<7:4> as inputs
MOVWF   TRISA           ;and set RA<3:0>
                        ; as outputs
BCF     STATUS,RP0      ;Bank 0
BCF     STATUS,RP1      ;
```

REGISTER 4-1: PORTA – PORTA REGISTER (ADDRESS: 05h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

bit 7-0 **RA<7:0>**: PORTA I/O Pin bits
1 = Port pin is >V_{IH}
0 = Port pin is <V_{IL}

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

REGISTER 4-2: TRISA – PORTA TRI-STATE REGISTER (ADDRESS: 85h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

bit 7-0 **TRISA<7:0>:** PORTA Tri-State Control bits
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

Note: TRISA<7:6> always reads ‘1’ in XT, HS, and LP OSC modes.

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

4.1.1 PIN DESCRIPTIONS AND DIAGRAMS

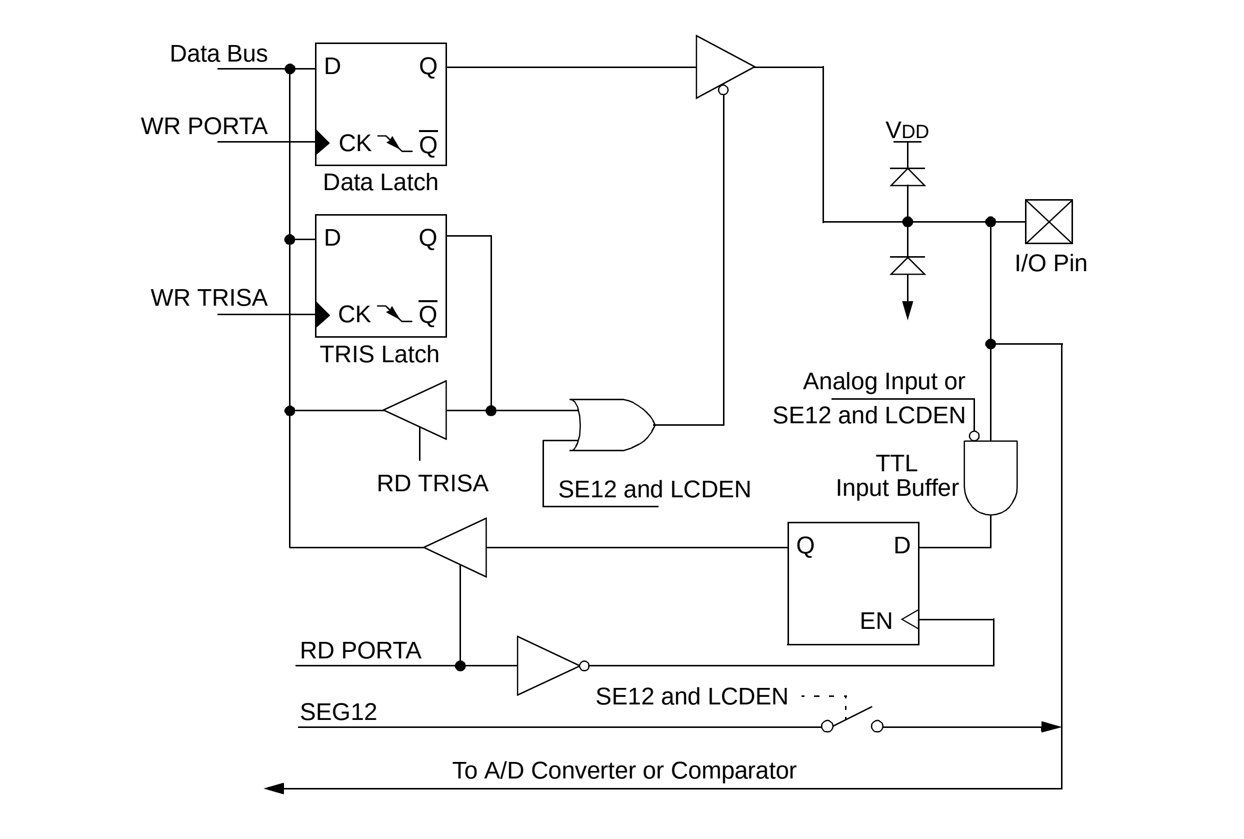
Each PORTA pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions, refer to the appropriate section in this data sheet.

4.1.1.1 RA0/AN0/C1-/SEG12

Figure 4-1 shows the diagram for this pin. The RA0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input for Comparator 1
- an analog output for the LCD

FIGURE 4-1: BLOCK DIAGRAM OF RA0

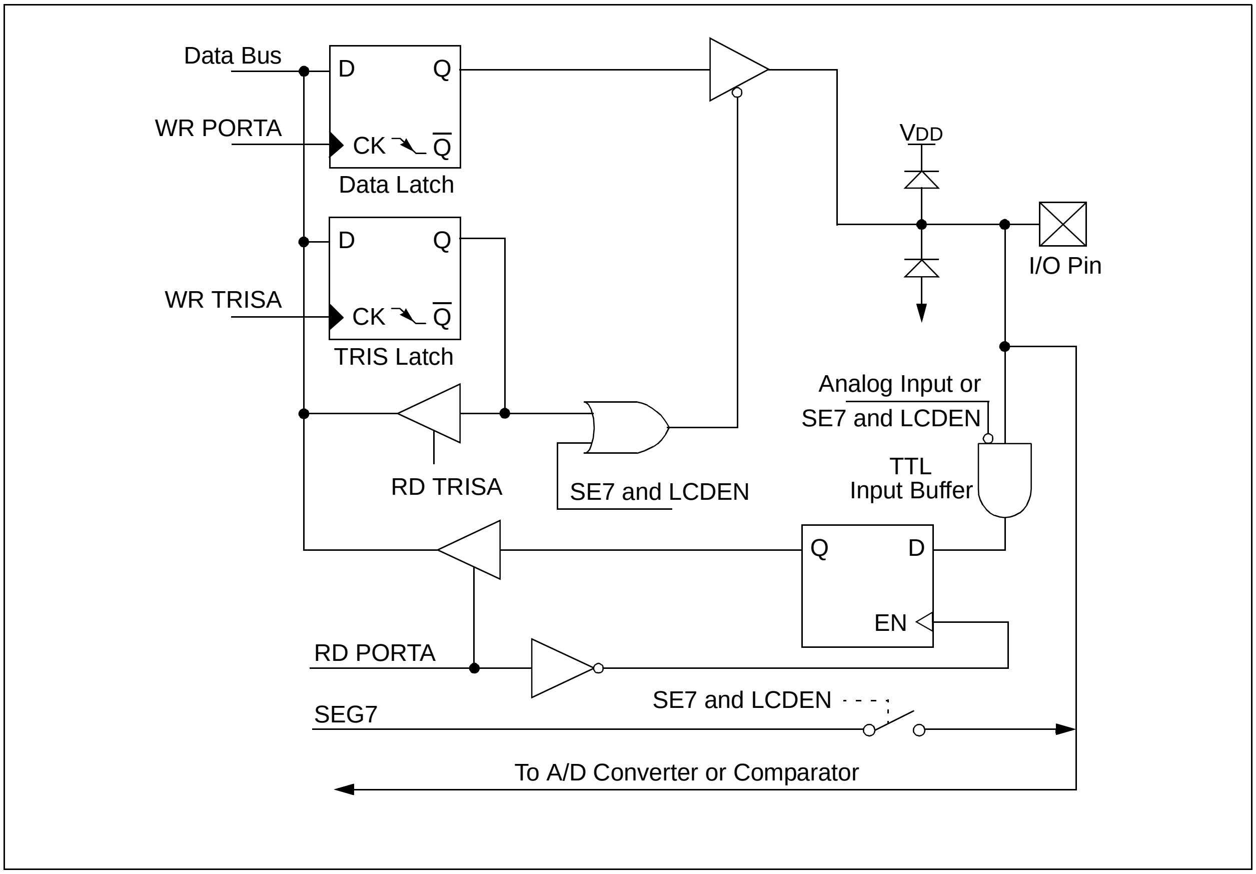


4.1.1.2 RA1/AN1/C2-/SEG7

Figure 4-2 shows the diagram for this pin. The RA1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input for Comparator 2
- an analog output for the LCD

FIGURE 4-2: BLOCK DIAGRAM OF RA1

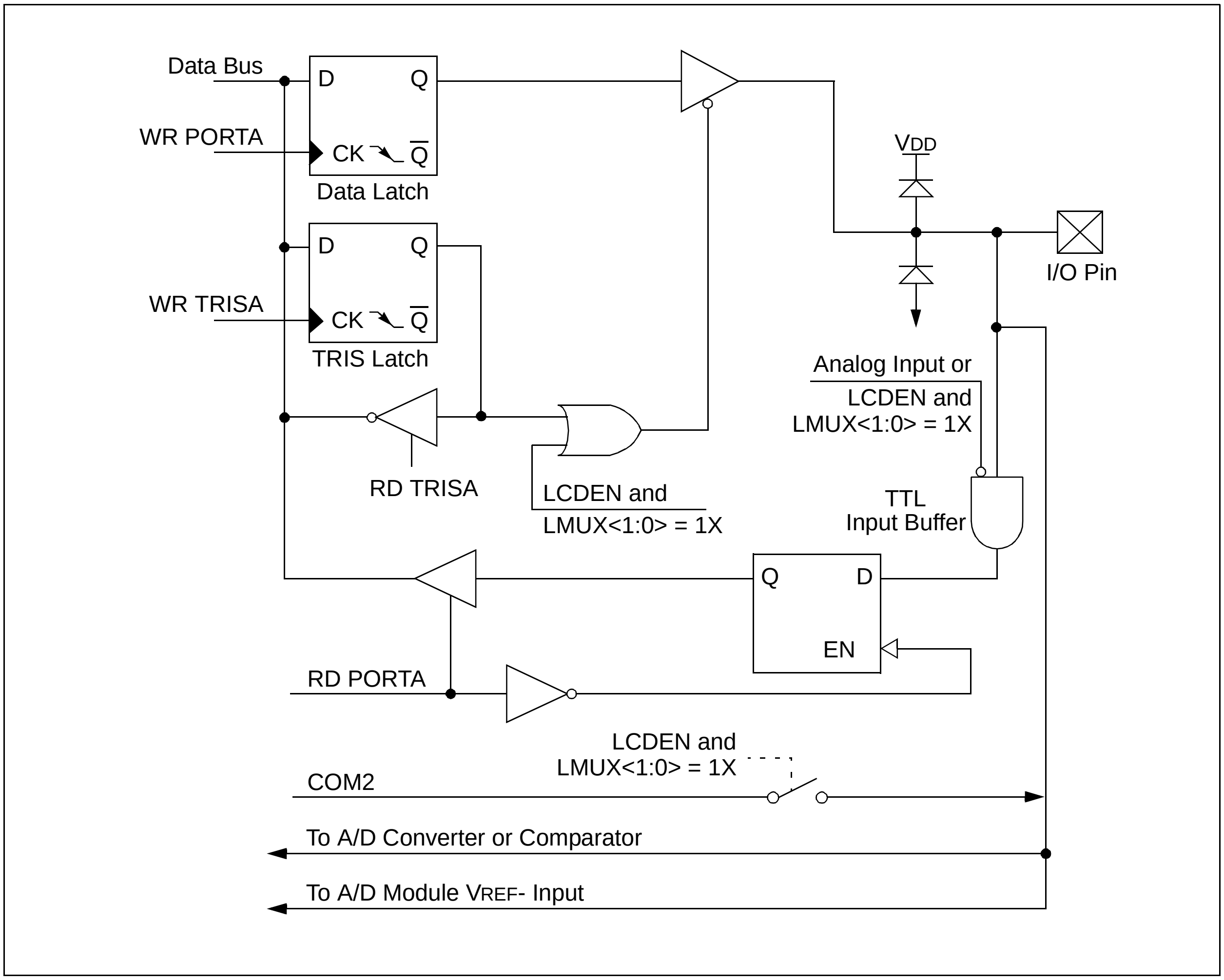


4.1.1.3 RA2/AN2/C2+/VREF-/COM2

Figure 4-3 shows the diagram for this pin. The RA2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input for Comparator 2
- a voltage reference input for the A/D
- an analog output for the LCD

FIGURE 4-3: BLOCK DIAGRAM OF RA2

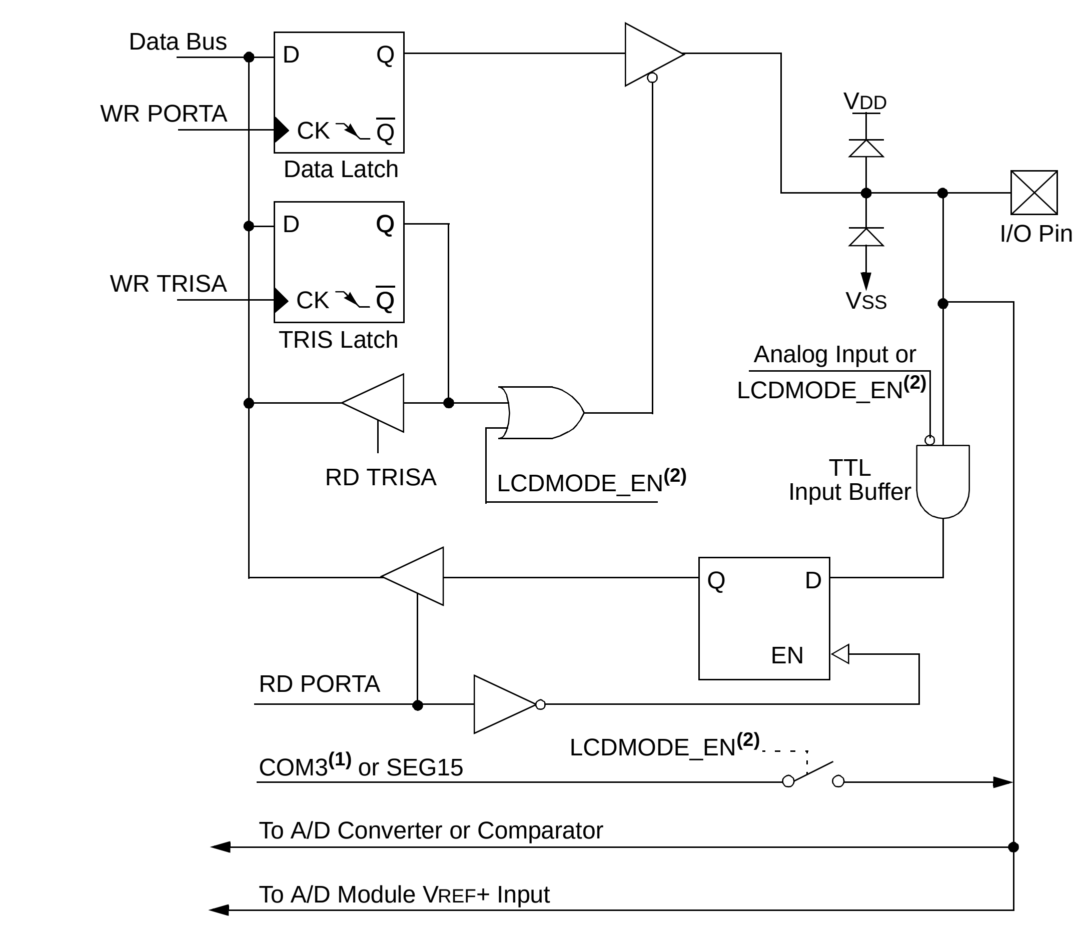


4.1.1.4 RA3/AN3/C1+/VREF+/COM3/SEG15

Figure 4-4 shows the diagram for this pin. The RA3 pin is configurable to function as one of the following:

- a general purpose input
- an analog input for the A/D
- an analog input from Comparator 1
- a voltage reference input for the A/D
- analog outputs for the LCD

FIGURE 4-4: BLOCK DIAGRAM OF RA3



Note 1: PIC16F913/916 only.

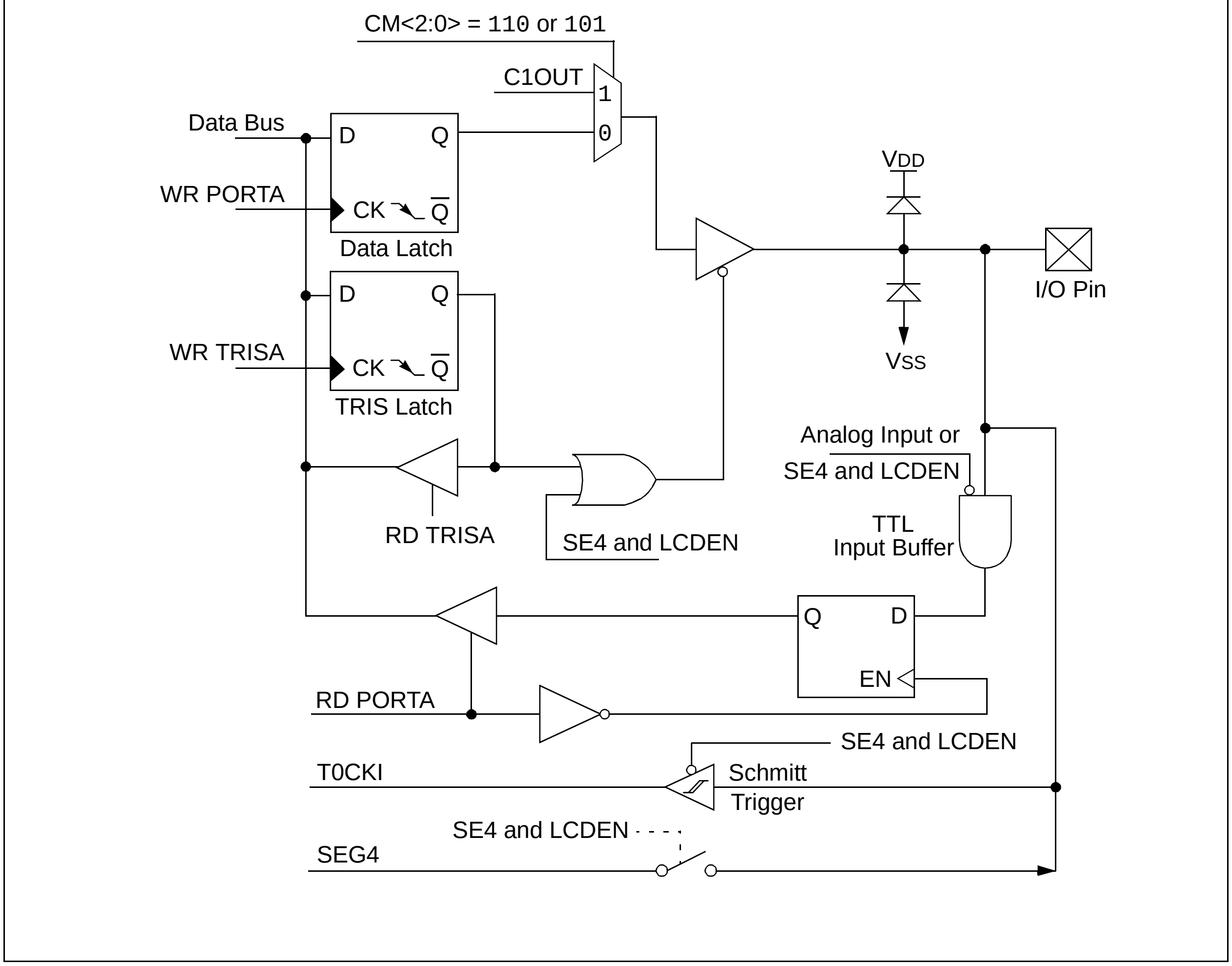
2: For the PIC16F913/916, the LCDMODE_EN = LCDEN and (SE15 or LMUX<1:0> = 11). For the PIC16F914/917, the LCDMODE_EN = LCDEN and SE15.

4.1.1.5 RA4/C1OUT/T0CK1/SEG4

Figure 4-5 shows the diagram for this pin. The RA4 pin is configurable to function as one of the following:

- a general purpose I/O
- a digital output from Comparator 1
- a clock input for TMR0
- an analog output for the LCD

FIGURE 4-5: BLOCK DIAGRAM OF RA4

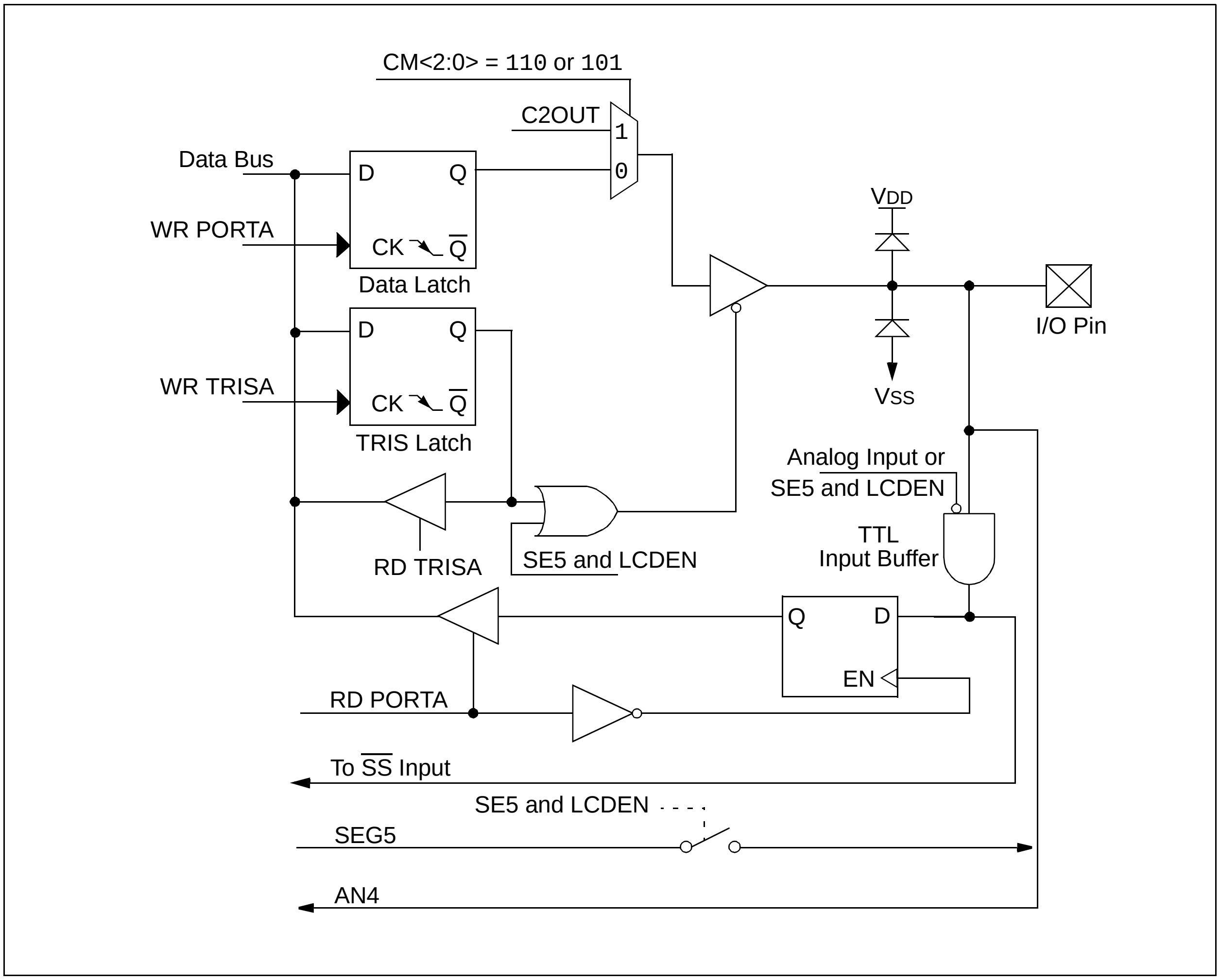


4.1.1.6 RA5/AN4/C2OUT/$\overline{\text{SS}}$/SEG5

Figure 4-6 shows the diagram for this pin. The RA5 pin is configurable to function as one of the following:

- a general purpose I/O
- a digital output from Comparator 2
- a slave select input
- an analog output for the LCD
- an analog input for the A/D

FIGURE 4-6: BLOCK DIAGRAM OF RA5

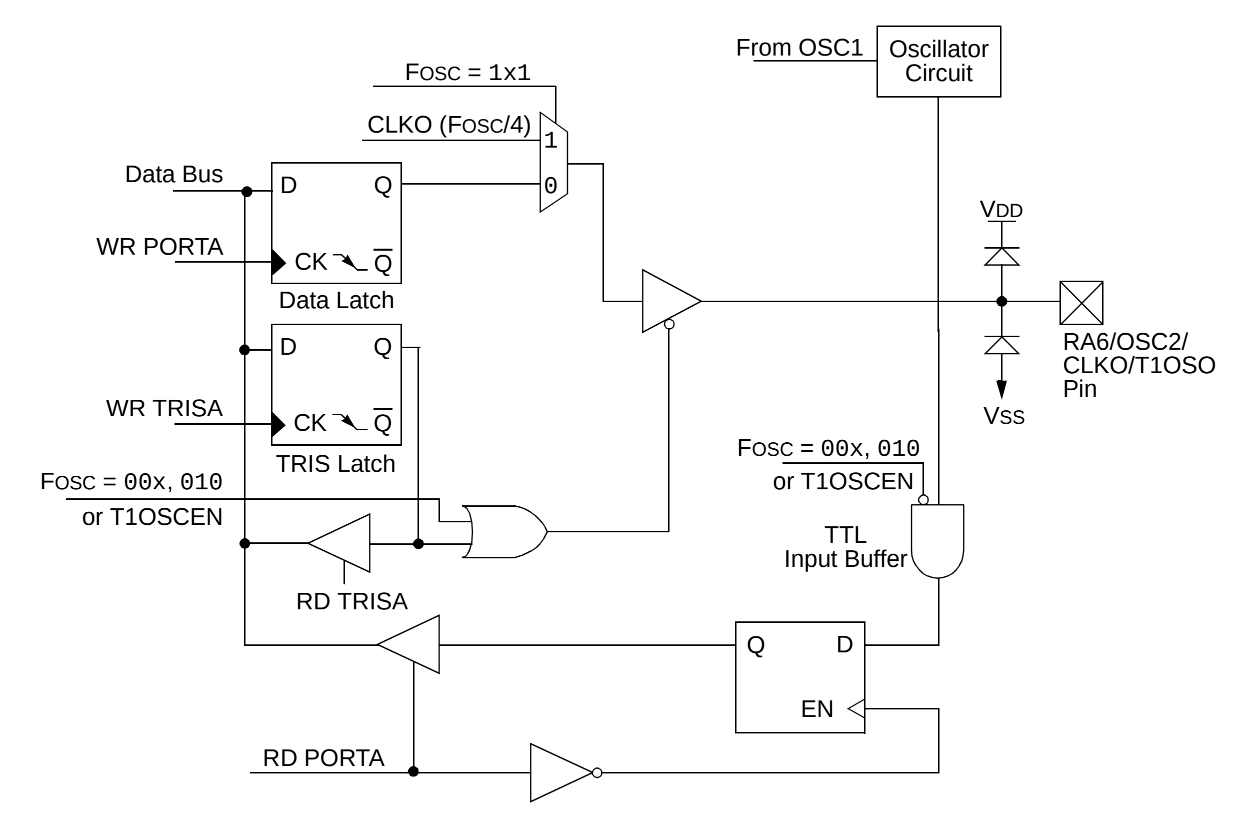


4.1.1.7 RA6/OSC2/CLKO/T1OSO

Figure 4-7 shows the diagram for this pin. The RA6 pin is configurable to function as one of the following:

- a general purpose I/O
- a crystal/resonator connection
- a clock output
- a TMR1 oscillator connection

FIGURE 4-7: BLOCK DIAGRAM OF RA6



4.1.1.8 RA7/OSC1/CLKI/T1OSI

Figure 4-8 shows the diagram for this pin. The RA7 pin is configurable to function as one of the following:

- a general purpose I/O
- a crystal/resonator connection
- a clock input
- a TMR1 oscillator connection

FIGURE 4-8: BLOCK DIAGRAM OF RA7

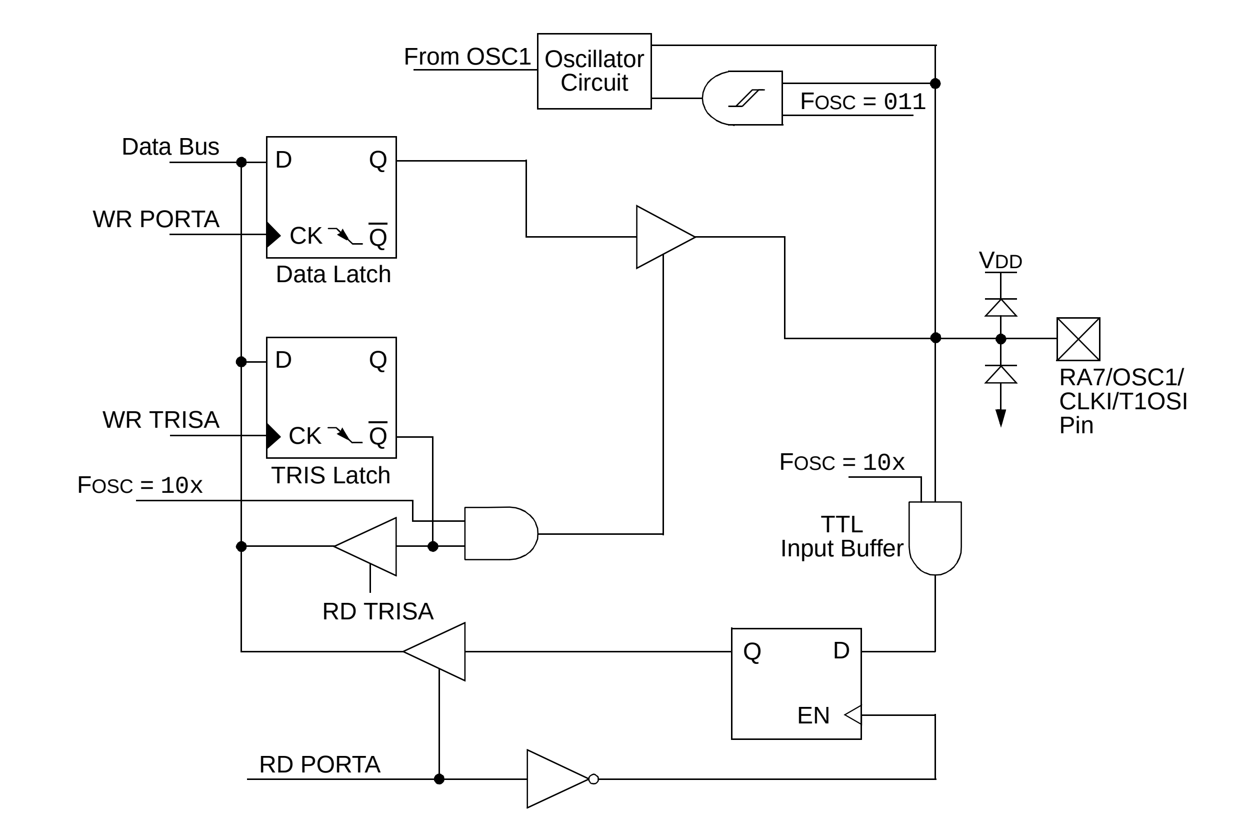


TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	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
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
1Fh	ADCON0	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	0000 0000
81h/181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
9Ch	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	0001 0011	0001 0011
11Ch	LCDSE0(1)	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu
11Dh	LCDSE1(1)	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTA.
Note 1: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

4.2 PORTB and TRISB Registers

PORTB is a general purpose I/O port with similar functionality as the PIC16F77. All PORTB pins can have a weak pull-up feature, and PORTB<7:4> implements an interrupt-on-input change function.

PORTB is also used for the Serial Flash programming interface.

> **Note:** Analog lines that carry LCD signals (i.e., SEGx, COMy, where x and y are segment and common identifiers) are shown as direct connections to the device pins. The signals are outputs from the LCD module and may be tri-stated, depending on the configuration of the LCD module.

EXAMPLE 4-2: INITIALIZING PORTB

```
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
CLRF    PORTB      ;Init PORTB
BSF     STATUS,RP0 ;Bank 1
BCF     STATUS,RP1 ;
MOVLW   FFh        ;Set RB<7:0> as inputs
MOVWF   TRISB      ;
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
```

4.3 Additional PORTB Pin Functions

RB<7:6> are used as data and clock signals, respectively, for both serial programming and the in-circuit debugger features on the device. Also, RB0 can be configured as an external interrupt input.

4.3.1 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:0> enable or disable each pull-up. Refer to Register 4-6. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

4.3.2 INTERRUPT-ON-CHANGE

Four of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:4> enable or disable the interrupt function for each pin. Refer to Register 4-5. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTB. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RBIF) in the INTCON register (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear the flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading or writing PORTB will end the mismatch condition and allow flag bit RBIF to be cleared. The latch holding the last read value is not affected by a $\overline{\text{MCLR}}$ nor BOR Reset. After these Resets, the RBIF flag will continue to be set if a mismatch is present.

> **Note:** If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set. Furthermore, since a read or write on a port affects all bits of that port, care must be taken when using multiple pins in Interrupt-on-change mode. Changes on one pin may not be seen while servicing changes on another pin.

REGISTER 4-3: PORTB – PORTB REGISTER (ADDRESS: 06h OR 106h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
bit 7							bit 0

bit 7-0 **RB<7:0>**: PORTB I/O Pin bits
1 = Port pin is >V_{IH}
0 = Port pin is <V_{IL}

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

REGISTER 4-4: TRISB – PORTB TRI-STATE REGISTER (ADDRESS: 86h, 186h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
bit 7							bit 0

bit 7-0 **TRISB<7:0>:** PORTB Tri-State Control bits
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output

Note: TRISB<7:6> always reads ‘1’ in XT, HS and LP OSC modes.

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

REGISTER 4-5: IOCB – PORTB INTERRUPT-ON-CHANGE REGISTER (ADDRESS: 96h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—
bit 7							bit 0

bit 7-4 **IOCB<7:4>:** Interrupt-on-change bits
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

bit 3-0 **Unimplemented:** Read as ‘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

REGISTER 4-6: WPUB – WEAK PULL-UP REGISTER (ADDRESS: 95h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 0

bit 7-0 **WPUB<7:0>:** Weak Pull-up Register bits

1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{RBPU}}$ must be enabled for individual pull-ups to be enabled.

2: The weak pull-up device is automatically disabled if the pin is in Output mode (TRISB<7:0> = 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

4.3.3 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTB pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the LCD or interrupts, refer to the appropriate section in this data sheet.

4.3.3.1 RB0/INT/SEG0

Figure 4-9 shows the diagram for this pin. The RB0 pin is configurable to function as one of the following:

- a general purpose I/O
- an external edge triggered interrupt
- an analog output for the LCD

4.3.3.2 RB1/SEG1

Figure 4-9 shows the diagram for this pin. The RB1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

4.3.3.3 RB2/SEG2

Figure 4-9 shows the diagram for this pin. The RB2 pin is configurable to function as one of the following:

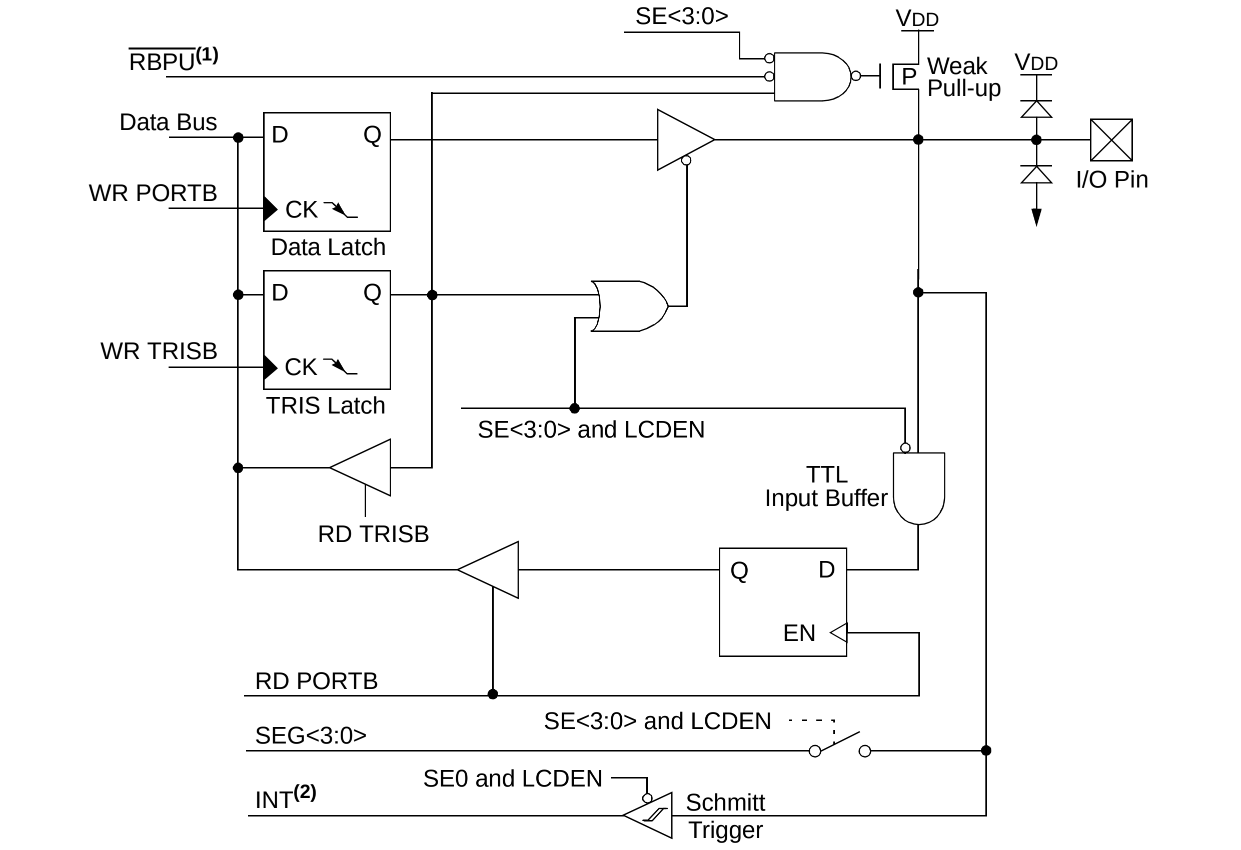
- a general purpose I/O
- an analog output for the LCD

4.3.3.4 RB3/SEG3

Figure 4-9 shows the diagram for this pin. The RB3 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 4-9: BLOCK DIAGRAM OF RB<3:0>



Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.

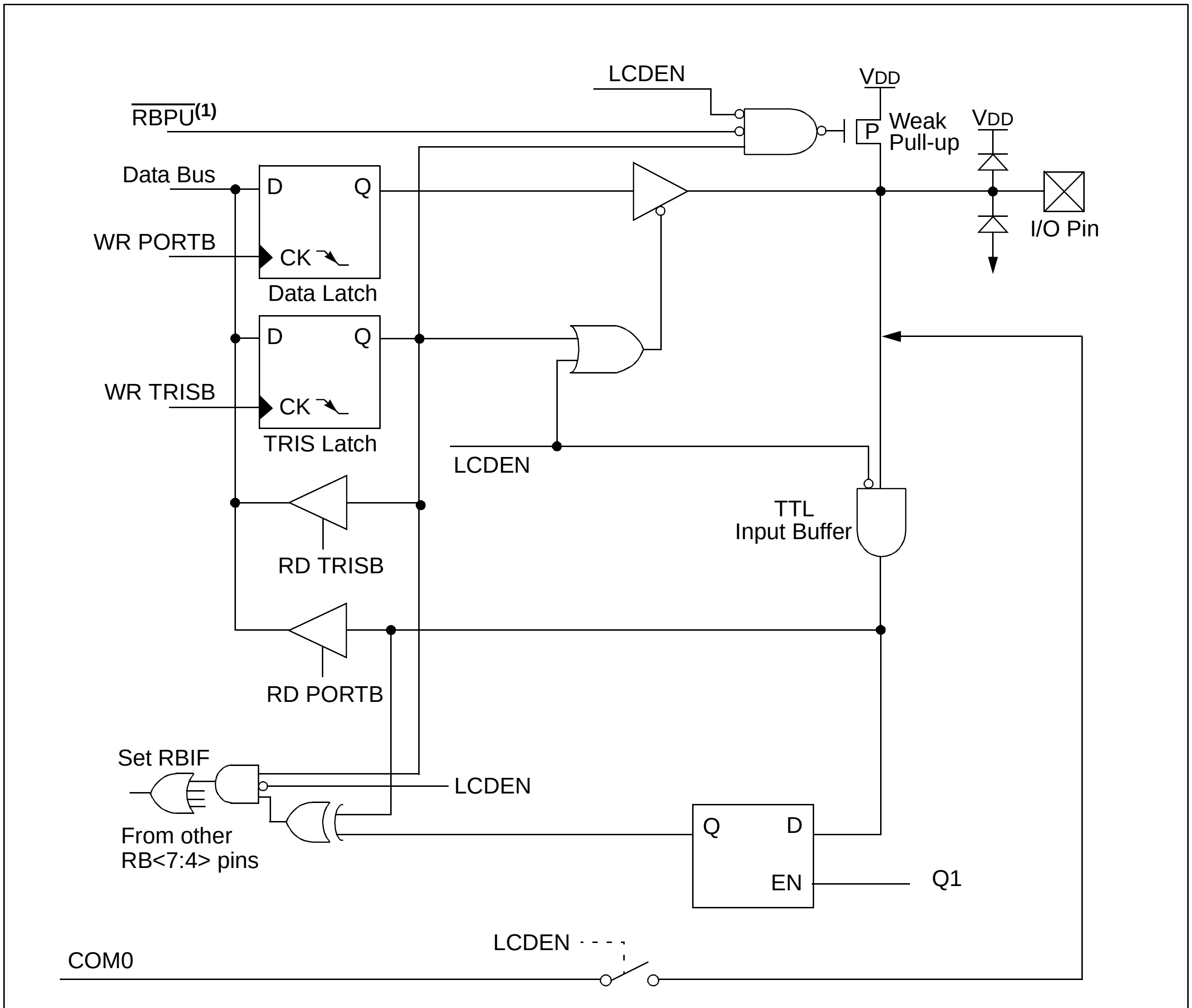
2: RB0 only.

4.3.3.5 RB4/COM0

Figure 4-10 shows the diagram for this pin. The RB4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 4-10: BLOCK DIAGRAM OF RB4



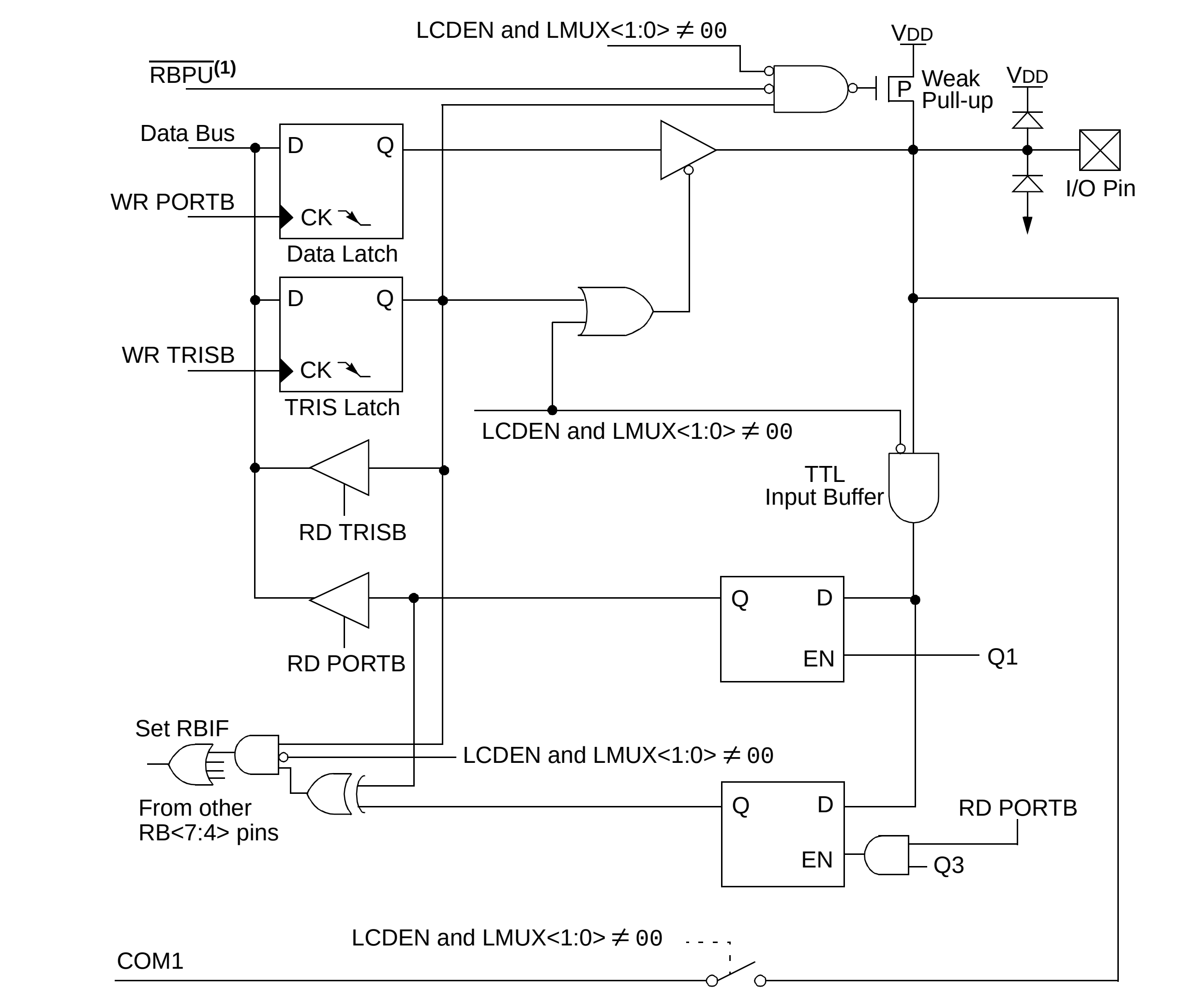
Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.

4.3.3.6 RB5/COM1

Figure 4-11 shows the diagram for this pin. The RB5 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 4-11: BLOCK DIAGRAM OF RB5



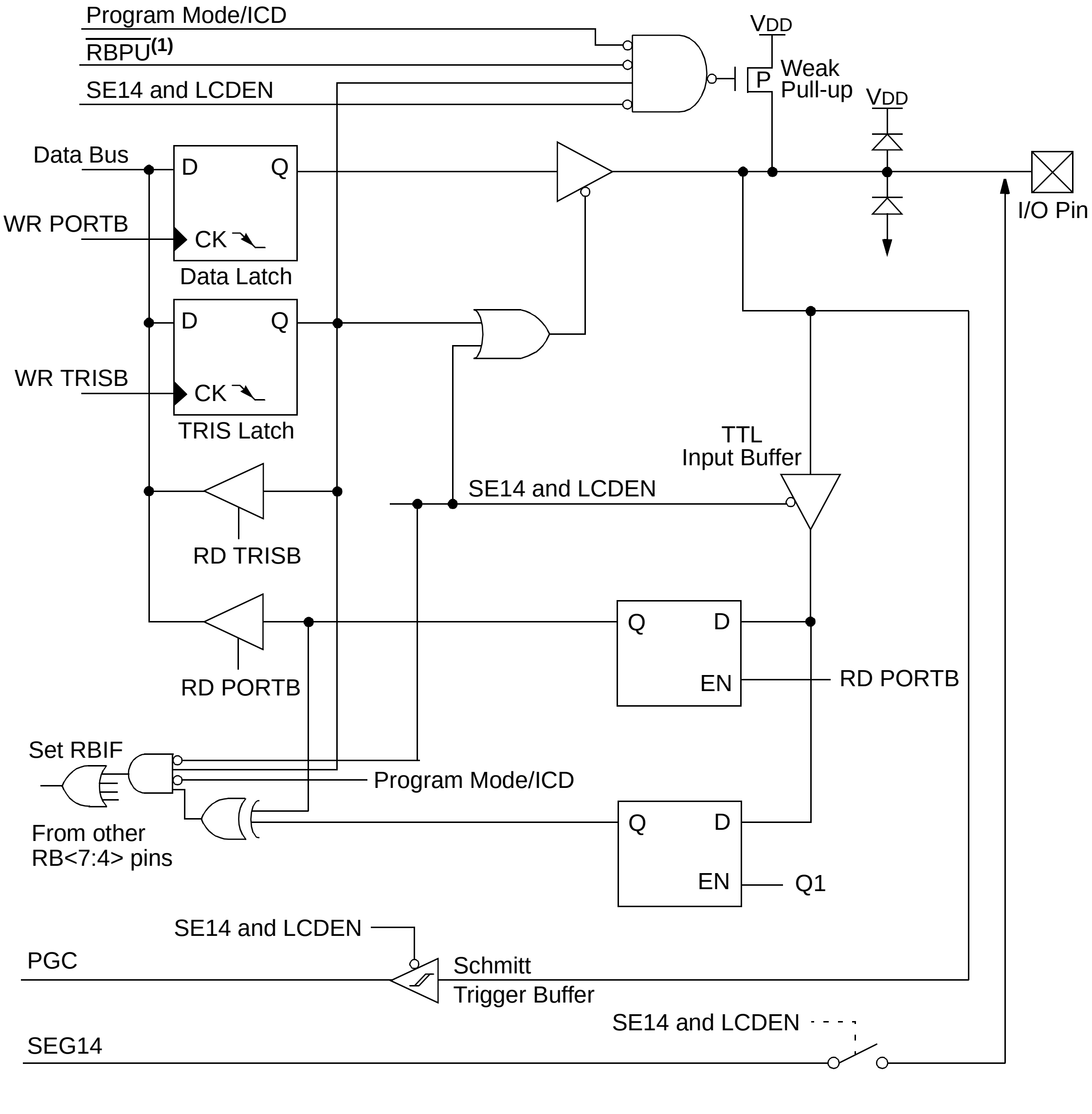
Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit.

4.3.3.7 RB6/ICSPCLK/ICDCK/SEG14

Figure 4-12 shows the diagram for this pin. The RB6 pin is configurable to function as one of the following:

- a general purpose I/O
- an In-Circuit Serial Programming™ clock
- an ICD clock I/O
- an analog output for the LCD

FIGURE 4-12: BLOCK DIAGRAM OF RB6



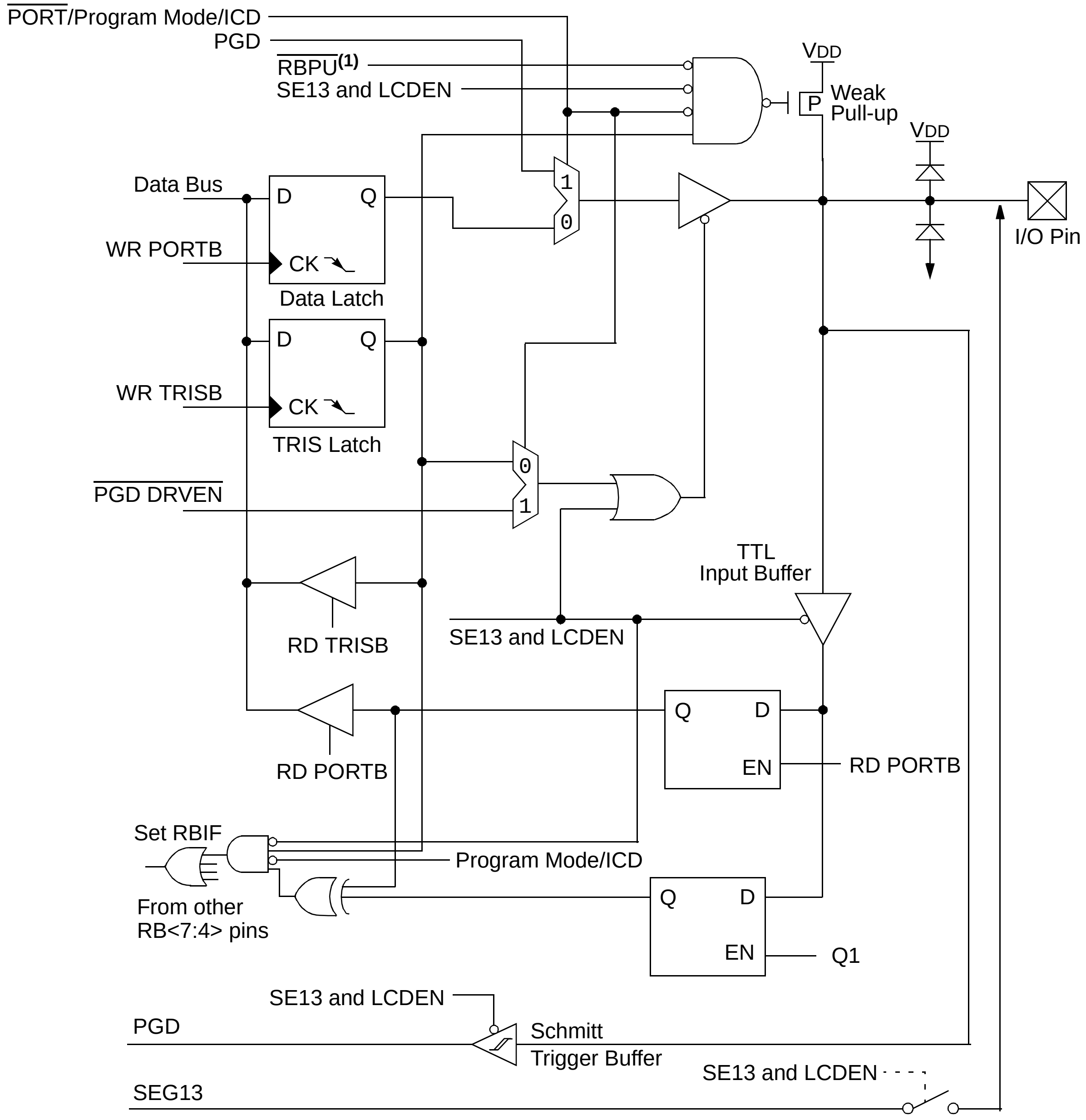
Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit.

4.3.3.8 RB7/ICSPDAT/ICDDAT/SEG13

Figure 4-13 shows the diagram for this pin. The RB7 pin is configurable to function as one of the following:

- a general purpose I/O
- an In-Circuit Serial Programming™ I/O
- an ICD data I/O
- an analog output for the LCD

FIGURE 4-13: BLOCK DIAGRAM OF RB7



Note 1: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit.

TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	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
06h/106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h/186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111
96h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	0001 0011	0001 0011
11Ch	LCDSE0(1)	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu
11Dh	LCDSE1(1)	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTB.

Note 1: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

4.4 PORTC and TRISC Registers

PORTC is an 8-bit bidirectional port. PORTC is multiplexed with several peripheral functions. PORTC pins have Schmitt Trigger input buffers.

All PORTC pins have latch bits (PORTC register). They, when written, will modify the contents of the PORTC latch; thus, modifying the value driven out on a pin if the corresponding TRISC bit is configured for output.

Note: Analog lines that carry LCD signals (i.e., SEGx, VLCDy, where x and y are segment and LCD bias voltage identifiers) are shown as direct connections to the device pins. The signals are outputs from the LCD module and may be tri-stated, depending on the configuration of the LCD module.

EXAMPLE 4-3: INITIALIZING PORTC

```
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
CLRF    PORTC      ;Init PORTC
BSF     STATUS,RP0 ;Bank 1
BCF     STATUS,RP1 ;
MOVLW   FFh        ;Set RC<7:0> as inputs
MOVWF   TRISC      ;
BCF     STATUS,RP0 ;Bank 2
BSF     STATUS,RP1 ;
CLRF    LCDCON     ;Disable VLCD<3:1>
                   ;inputs on RC<2:0>
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
```

REGISTER 4-7: PORTC – PORTC REGISTER (ADDRESS: 07h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

bit 7-0 **RC<7:0>**: PORTC I/O Pin bits

1 = Port pin is >V_{IH}

0 = Port pin is <V_{IL}

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

REGISTER 4-8: TRISC – PORTC TRI-STATE REGISTER (ADDRESS: 87h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

bit 7-0 **TRISC<7:0>:** PORTC Tri-State Control bits

1 = PORTC pin configured as an input (tri-stated)

0 = PORTC pin configured as an output

Note: TRISC<7:6> always reads ‘1’ in XT, HS and LP OSC modes.

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

4.4.1 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTC pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the LCD or SSP, refer to the appropriate section in this data sheet.

4.4.1.1 RC0/VLCD1

Figure 4-14 shows the diagram for this pin. The RC0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the LCD bias voltage

4.4.1.2 RC1/VLCD2

Figure 4-15 shows the diagram for this pin. The RC1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the LCD bias voltage

4.4.1.3 RC2/VLCD3

Figure 4-16 shows the diagram for this pin. The RC2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the LCD bias voltage

FIGURE 4-14: BLOCK DIAGRAM OF RC0

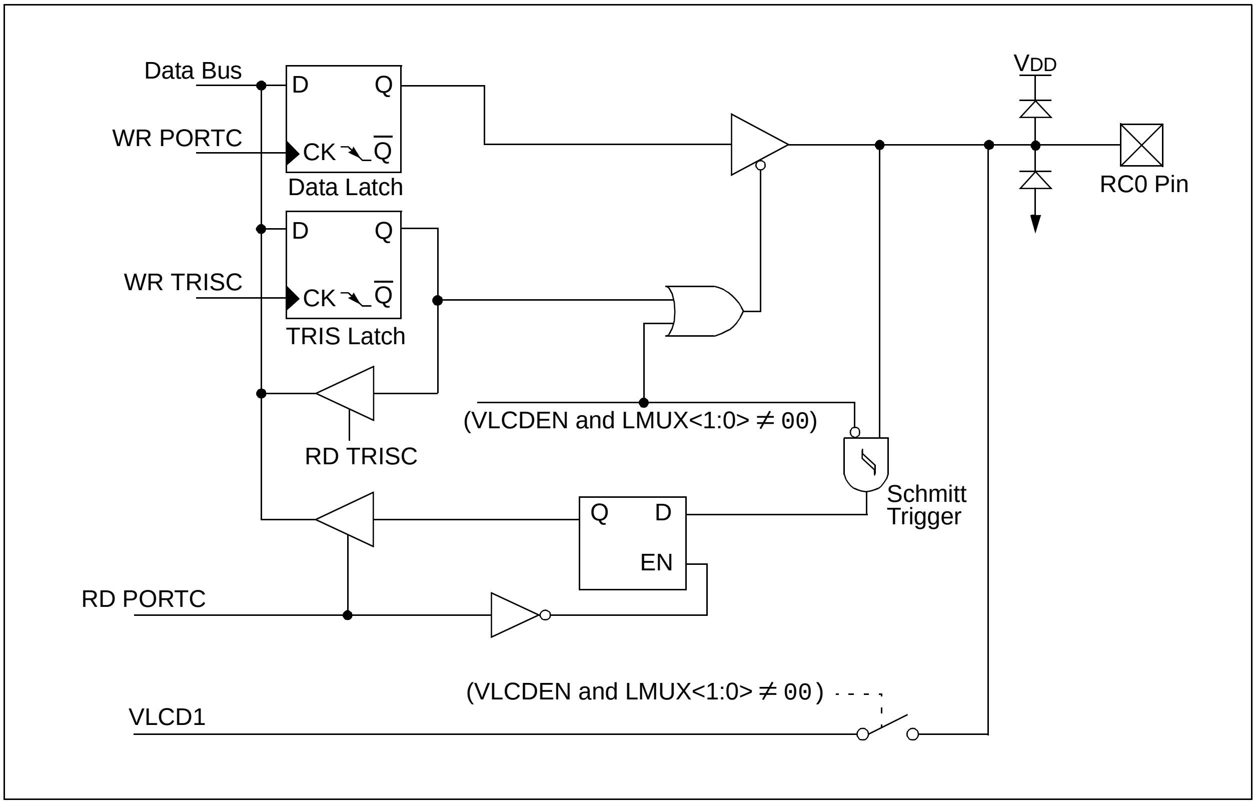


FIGURE 4-15: BLOCK DIAGRAM OF RC1

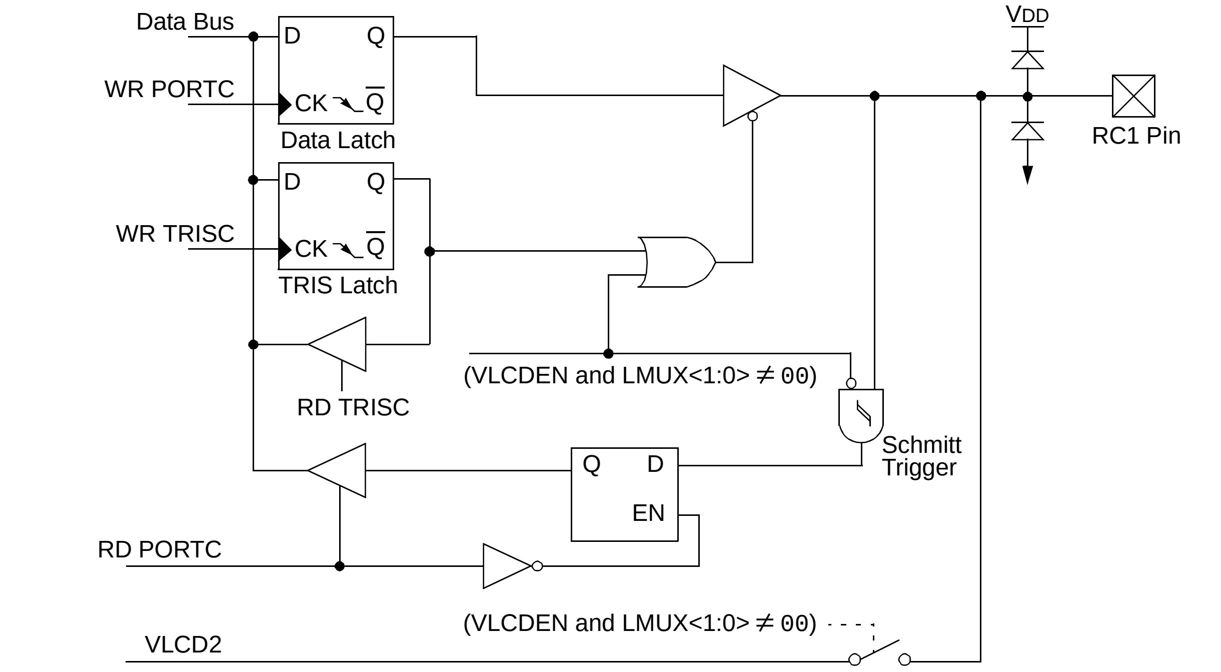
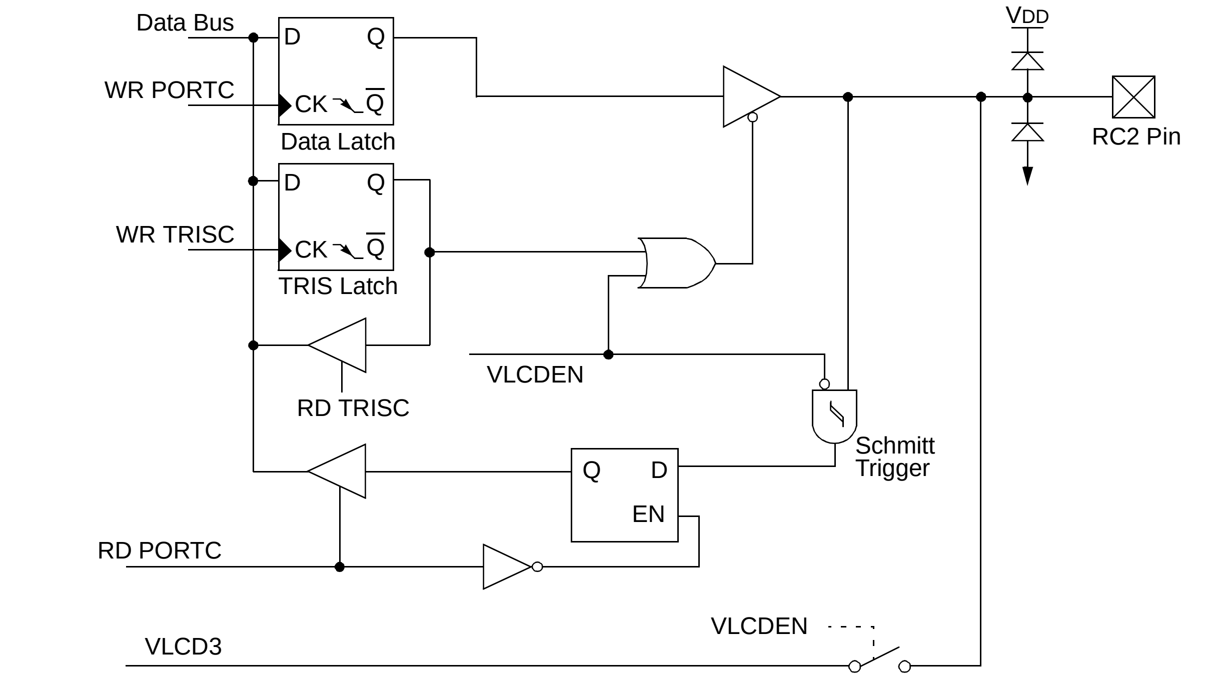


FIGURE 4-16: BLOCK DIAGRAM OF RC2

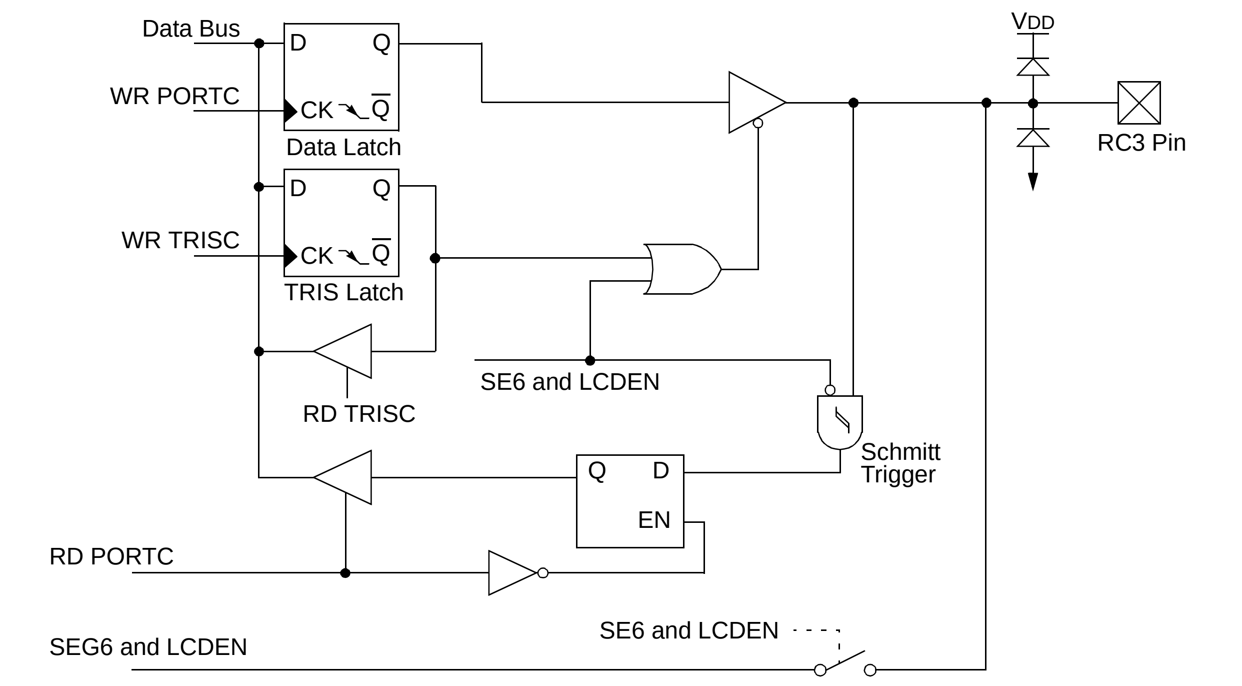


4.4.1.4 RC3/SEG6

Figure 4-17 shows the diagram for this pin. The RC3 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 4-17: BLOCK DIAGRAM OF RC3

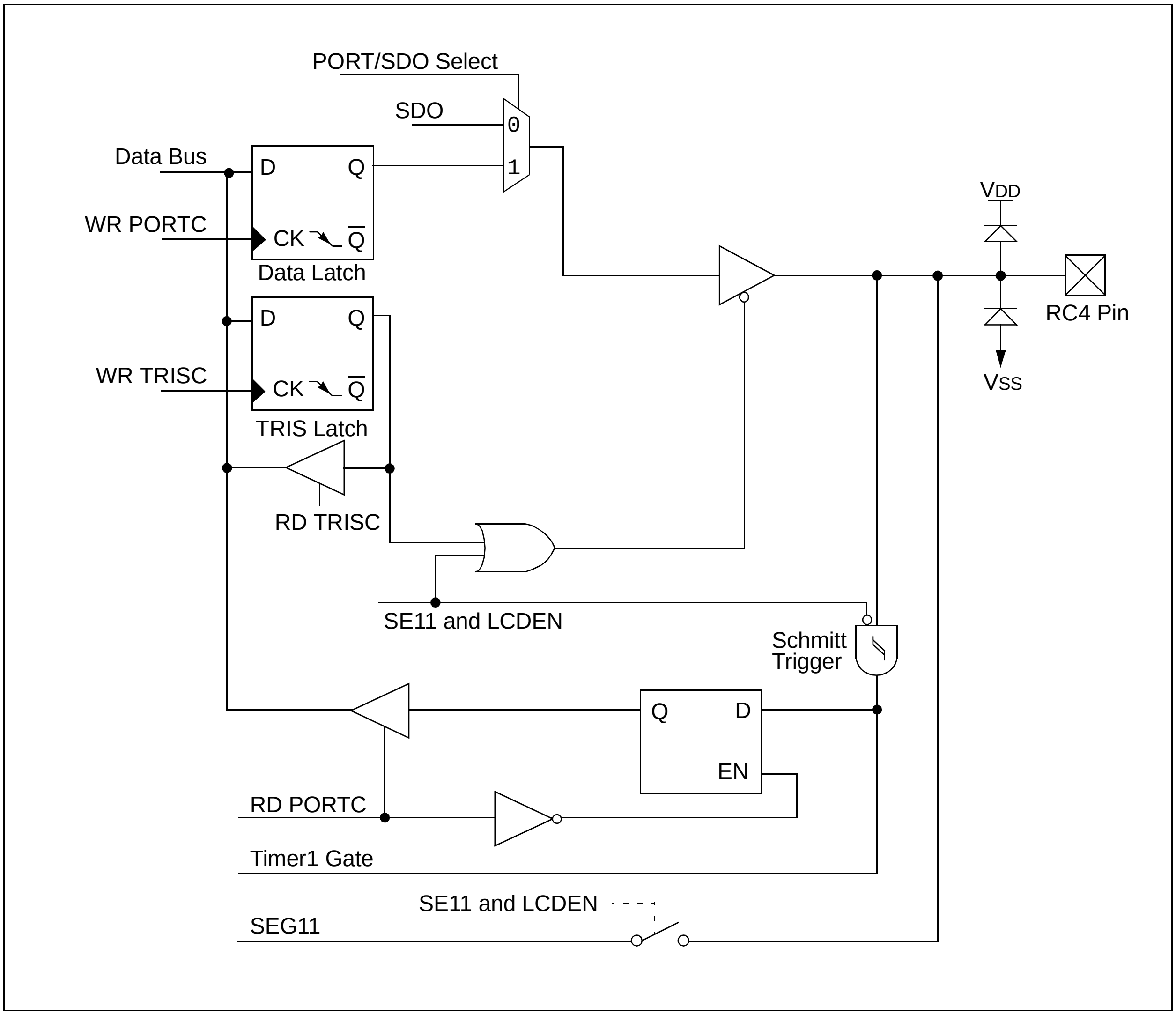


4.4.1.5 RC4/$\overline{\text{T1G}}$/SDO/SEG11

Figure 4-18 shows the diagram for this pin. The RC4 pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 gate input
- a serial data output
- an analog output for the LCD

FIGURE 4-18: BLOCK DIAGRAM OF RC4

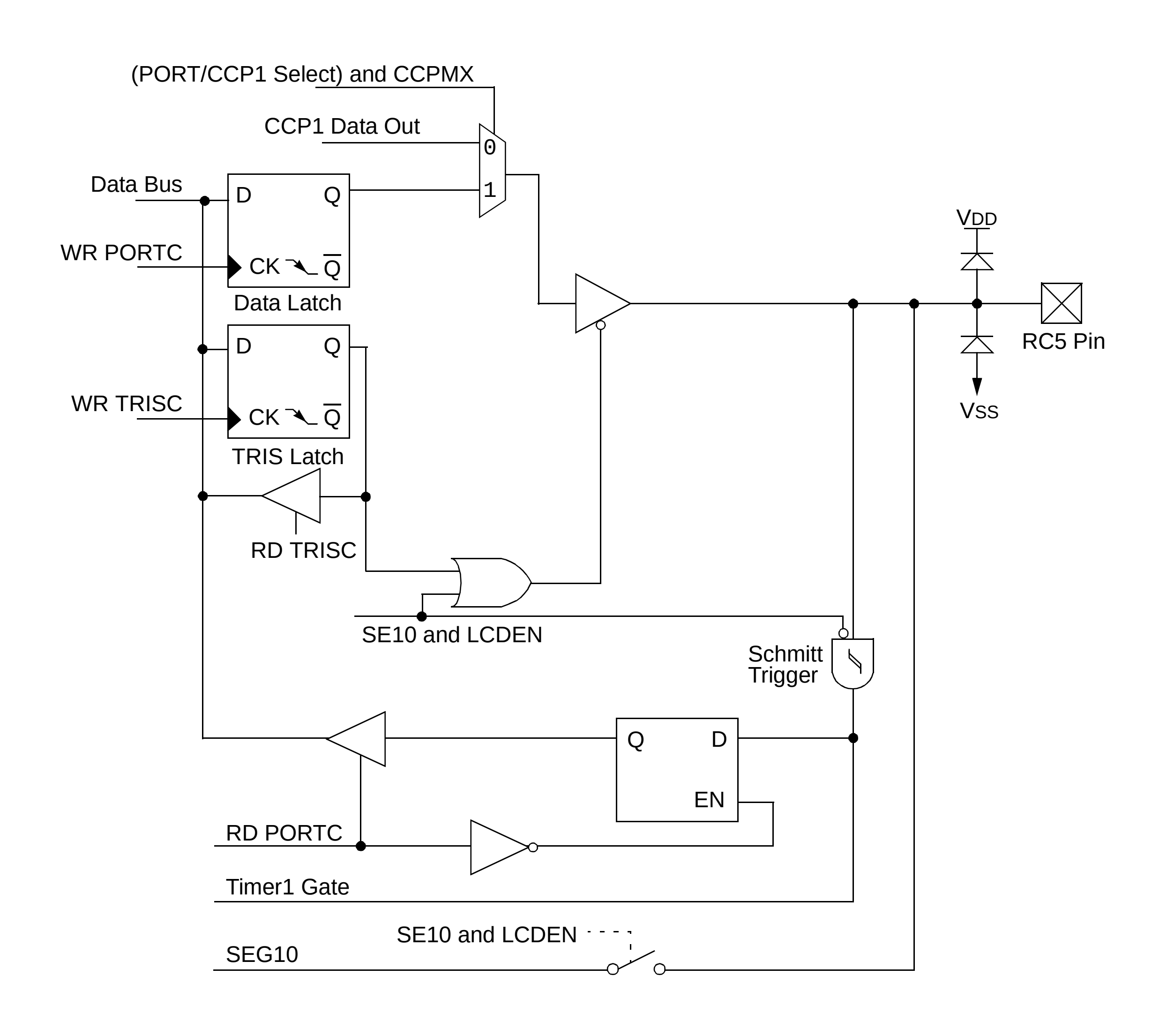


4.4.1.6 RC5/T1CKI/CCP1/SEG10

Figure 4-19 shows the diagram for this pin. The RC5 pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 clock input
- a Capture input, Compare output or PWM output
- an analog output for the LCD

FIGURE 4-19: BLOCK DIAGRAM OF RC5

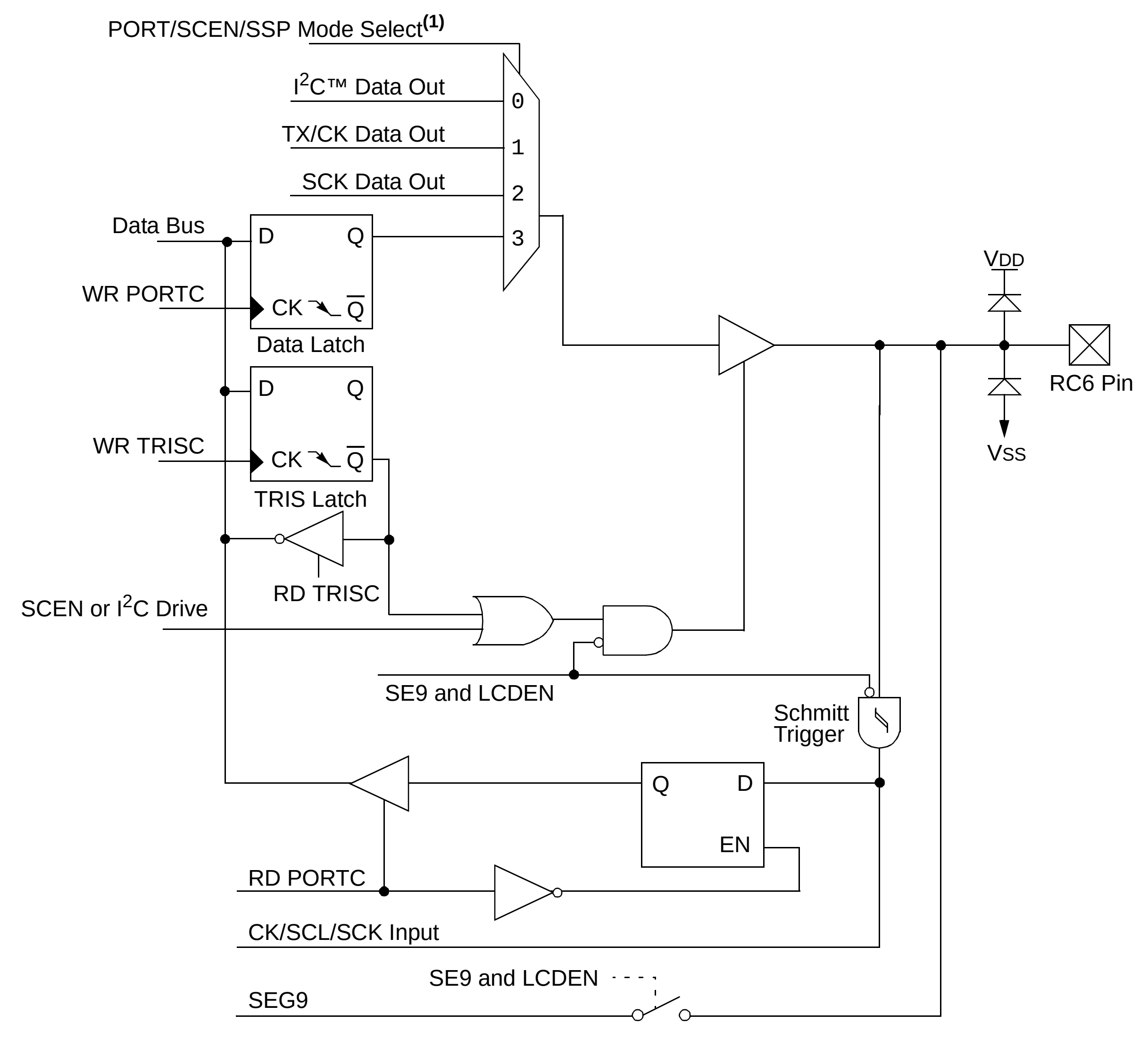


4.4.1.7 RC6/TX/CK/SCK/SCL/SEG9

Figure 4-20 shows the diagram for this pin. The RC6 pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial output
- a synchronous clock I/O
- a SPI clock I/O
- an I^{2}C data I/O
- an analog output for the LCD

FIGURE 4-20: BLOCK DIAGRAM OF RC6



Note 1: If all three data output sources are enabled, the following priority order will be used:

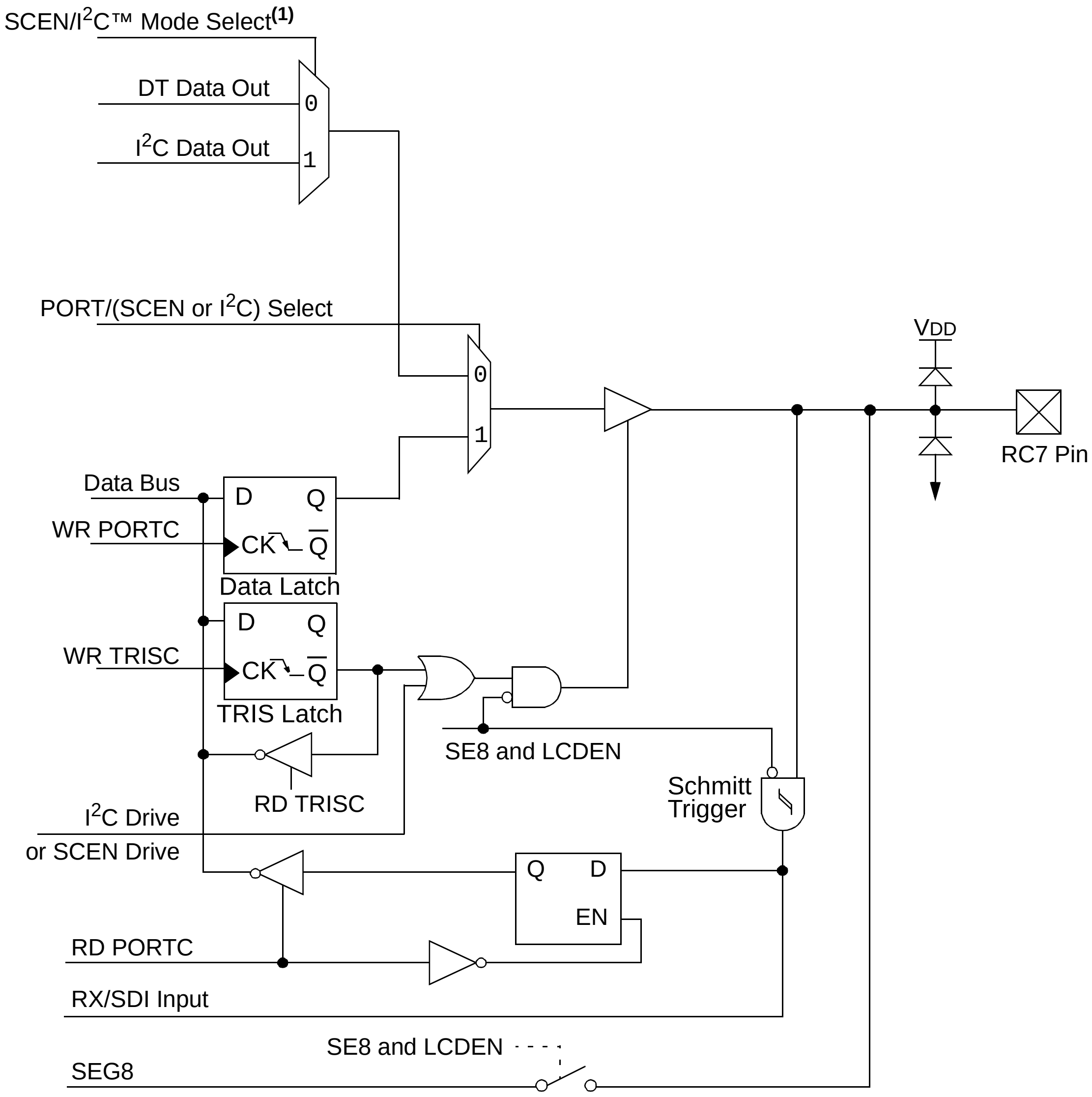
- USART data
- SSP data
- PORT data

4.4.1.8 RC7/RX/DT/SDI/SDA/SEG8

Figure 4-21 shows the diagram for this pin. The RC7 pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial input
- a synchronous serial data I/O
- a SPI data I/O
- an I^{2}C data I/O
- an analog output for the LCD

FIGURE 4-21: BLOCK DIAGRAM OF RC7



Note 1: If SSP and USART outputs are both enabled, the USART data output will have priority over the SSP data output. Both SSP and USART data outputs will have priority over the PORT data output.

TABLE 4-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	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
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	`xxxx xxxx`	`uuuu uuuu`
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	`0000 0000`	`uuuu uuuu`
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	`0000 0000`	`0000 0000`
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	`--00 0000`	`--00 0000`
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 000x`	`0000 000x`
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`1111 1111`	`1111 1111`
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	`0001 0011`	`0001 0011`
11Ch	LCDSE0(1)	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	`0000 0000`	`uuuu uuuu`
11Dh	LCDSE1(1)	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	`0000 0000`	`uuuu uuuu`

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC.

Note 1: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

4.5 PORTD and TRISD Registers

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configured as an input or output.

PORTD is only available on the PIC16F914 and PIC16F917.

> **Note:** Analog lines that carry LCD signals (i.e., SEGx, COMy, where x and y are segment and common identifiers) are shown as direct connections to the device pins. The signals are outputs from the LCD module and may be tri-stated, depending on the configuration of the LCD module.

EXAMPLE 4-4: INITIALIZING PORTD

```
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
CLRF    PORTD      ;Init PORTD
BSF     STATUS,RP0 ;Bank 1
BCF     STATUS,RP1 ;
MOVLW   FFh        ;Set RD<7:0> as inputs
MOVWF   TRISD      ;
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
```

REGISTER 4-9: PORTD – PORTD REGISTER (ADDRESS: 08h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

bit 7 bit 0

bit 7-0 **RD<7:0>**: PORTD I/O Pin bits
1 = Port pin is >V_{IH}
0 = Port pin is <V_{IL}

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

REGISTER 4-10: TRISD – PORTD TRI-STATE REGISTER (ADDRESS: 88h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0

bit 7 bit 0

bit 7-0 **TRISD<7:0>:** PORTD Tri-State Control bits
1 = PORTD pin configured as an input (tri-stated)
0 = PORTD pin configured as an output

Note: TRISD<7:6> always reads ‘1’ in XT, HS and LP OSC modes.

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

4.5.1 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTD pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D, refer to the appropriate section in this data sheet.

4.5.1.1 RD0/COM3

Figure 4-22 shows the diagram for this pin. The RD0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D

4.5.1.2 RD1

Figure 4-23 shows the diagram for this pin. The RD1 pin is configurable to function as one of the following:

- a general purpose I/O

4.5.1.3 RD2/CCP2

Figure 4-24 shows the diagram for this pin. The RD2 pin is configurable to function as one of the following:

- a general purpose I/O
- a Capture input, Compare output or PWM output

4.5.1.4 RD3/SEG16

Figure 4-25 shows the diagram for this pin. The RD3 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

4.5.1.5 RD4/SEG17

Figure 4-25 shows the diagram for this pin. The RD4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

4.5.1.6 RD5/SEG18

Figure 4-25 shows the diagram for this pin. The RD5 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

4.5.1.7 RD6/SEG19

Figure 4-25 shows the diagram for this pin. The RD6 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

4.5.1.8 RD7/SEG20

Figure 4-25 shows the diagram for this pin. The RD7 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 4-22: BLOCK DIAGRAM OF RD0

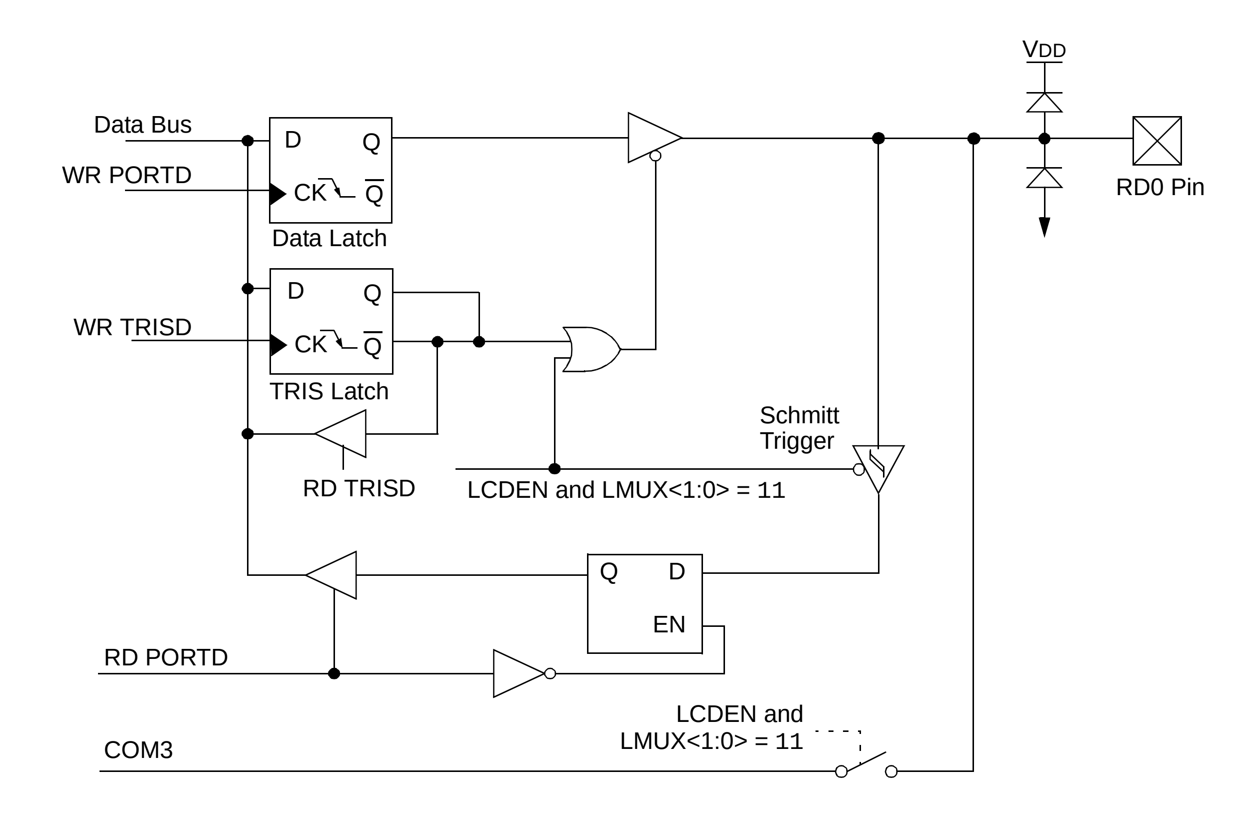


FIGURE 4-23: BLOCK DIAGRAM OF RD1

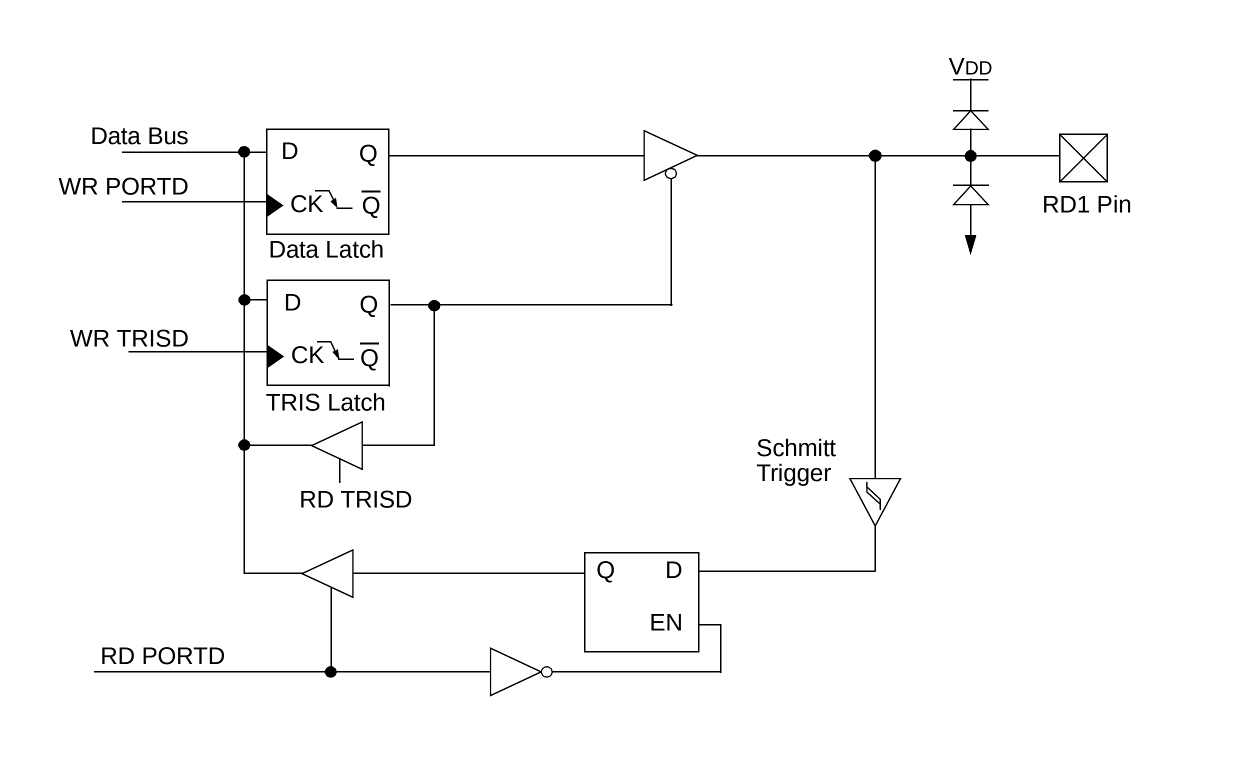


FIGURE 4-24: BLOCK DIAGRAM OF RD2

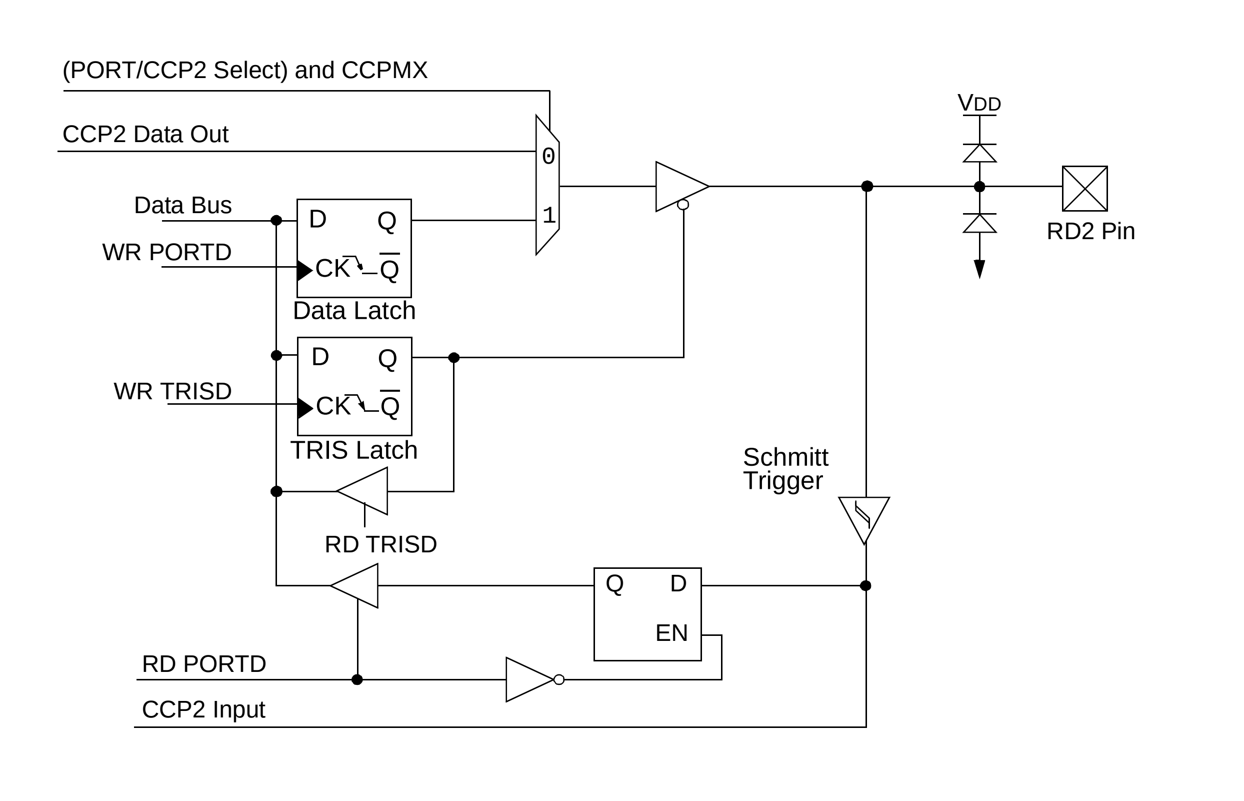


FIGURE 4-25: BLOCK DIAGRAM OF RD<7:3>

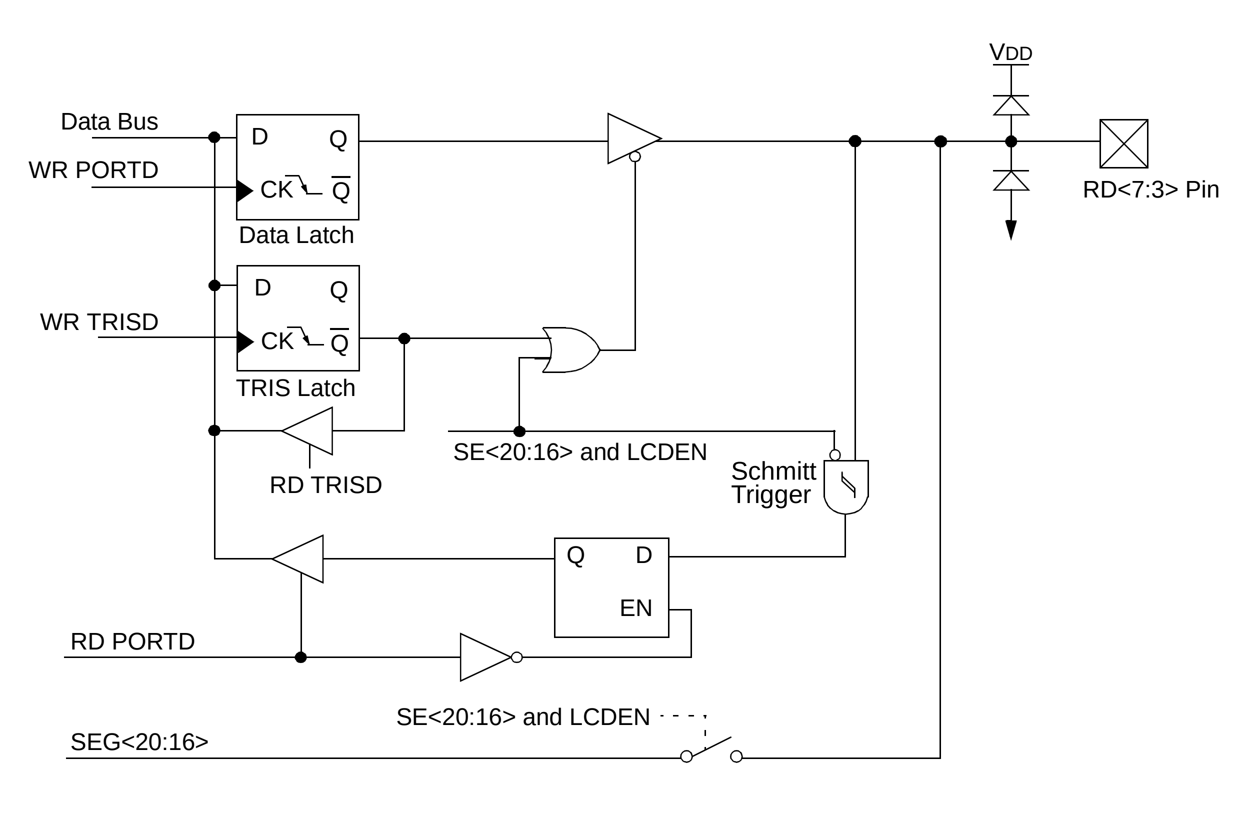


TABLE 4-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	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
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	`xxxx xxxx`	`uuuu uuuu`
1Dh(2)	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	`--00 0000`	`--00 0000`
88h	TRISD(2)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	`1111 1111`	`1111 1111`
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	`0001 0011`	`0001 0011`
11Eh	LCDSE2(1,2)	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	`0000 0000`	`uuuu uuuu`

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘`0`’. Shaded cells are not used by PORTC.

Note 1: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

2: PIC16F914/917 only.

4.6 PORTE and TRISE Registers

PORTE is a 4-bit port with Schmitt Trigger input buffers. RE<2:0> are individually configured as inputs or outputs. RE3 is only available as an input if MCLRE is '0' in Configuration Word (Register 16-1).

RE<2:0> are only available on the PIC16F914 and PIC16F917.

> **Note:** Analog lines that carry LCD signals (i.e., SEGx, where x are segment identifiers) are shown as direct connections to the device pins. The signals are outputs from the LCD module and may be tri-stated, depending on the configuration of the LCD module.

EXAMPLE 4-5: INITIALIZING PORTE

```
BCF     STATUS,RP0     ;Bank 0
BCF     STATUS,RP1     ;
CLRF    PORTE          ;Init PORTE
BSF     STATUS,RP0     ;Bank 1
BCF     STATUS,RP1     ;
MOVLW   0Fh            ;Set RE<3:0> as inputs
MOVWF   TRISE          ;
CLRF    ANSEL          ;Make RE<2:0> as I/O's
BCF     STATUS,RP0     ;Bank 0
BCF     STATUS,RP1     ;
```

REGISTER 4-11: PORTE – PORTE REGISTER (ADDRESS: 09h)

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	RE3	RE2	RE1	RE0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **RE<3:0>**: PORTE I/O Pin bits
1 = Port pin is >VIH
0 = Port pin is <VIL

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

REGISTER 4-12: TRISE – PORTE TRI-STATE REGISTER (ADDRESS: 89h)

U-0	U-0	U-0	U-0	R-1	R/W-1	R/W-1	R/W-1
—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **TRISE3:** Data Direction bit. RE3 is always an input, so this bit always reads as a '1'

bit 2-0 **TRISE<2:0>:** Data Direction bits

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

4.6.1 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTE pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D, refer to the appropriate section in this data sheet.

4.6.1.1 RE0/AN5/SEG21

Figure 4-26 shows the diagram for this pin. The RE0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog output for the LCD

4.6.1.2 RE1/AN6/SEG22

Figure 4-26 shows the diagram for this pin. The RE1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog output for the LCD

4.6.1.3 RE2/AN7/SEG23

Figure 4-26 shows the diagram for this pin. The RE2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog output for the LCD

4.6.1.4 RE3/$\overline{\text{MCLR}}$/VPP

Figure 4-27 shows the diagram for this pin. The RE3 pin is configurable to function as one of the following:

- a digital input only
- as Master Clear Reset with weak pull-up
- a programming voltage reference input

FIGURE 4-26: BLOCK DIAGRAM OF RE<2:0>

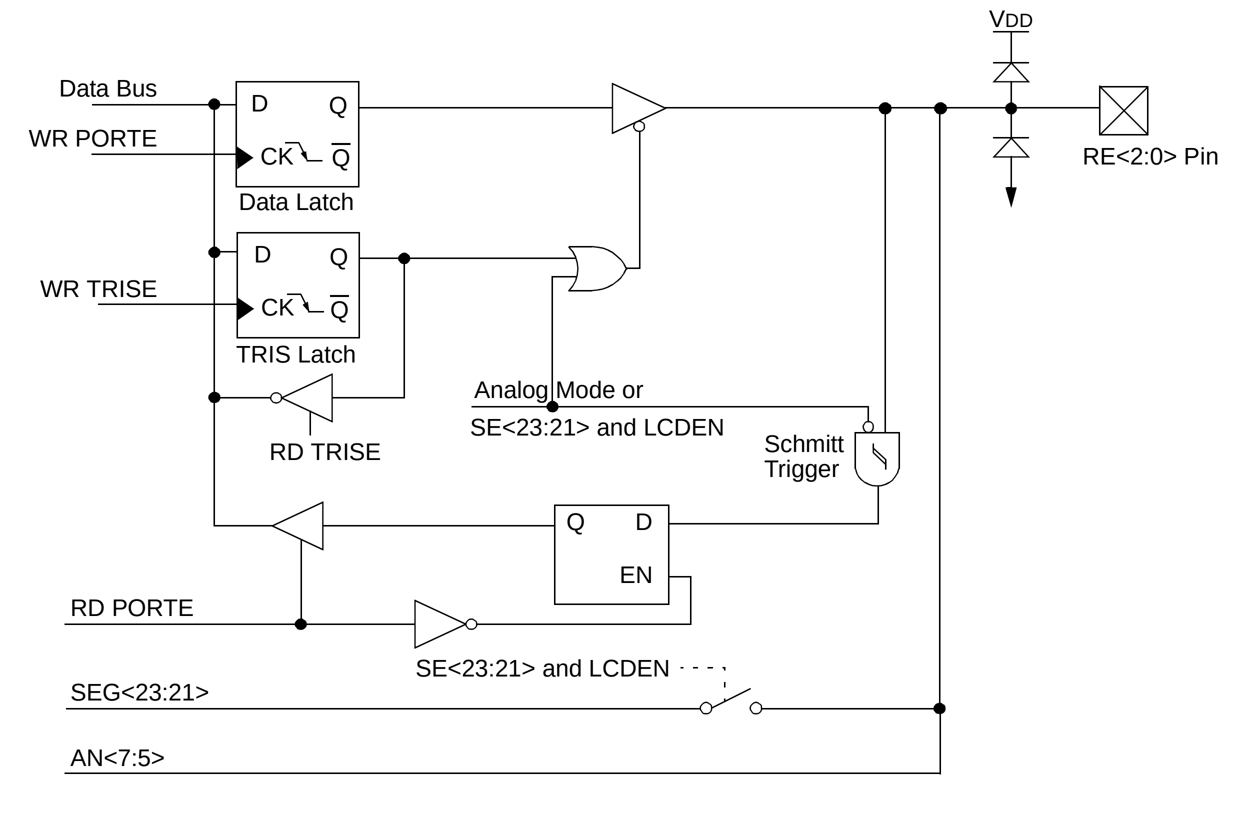
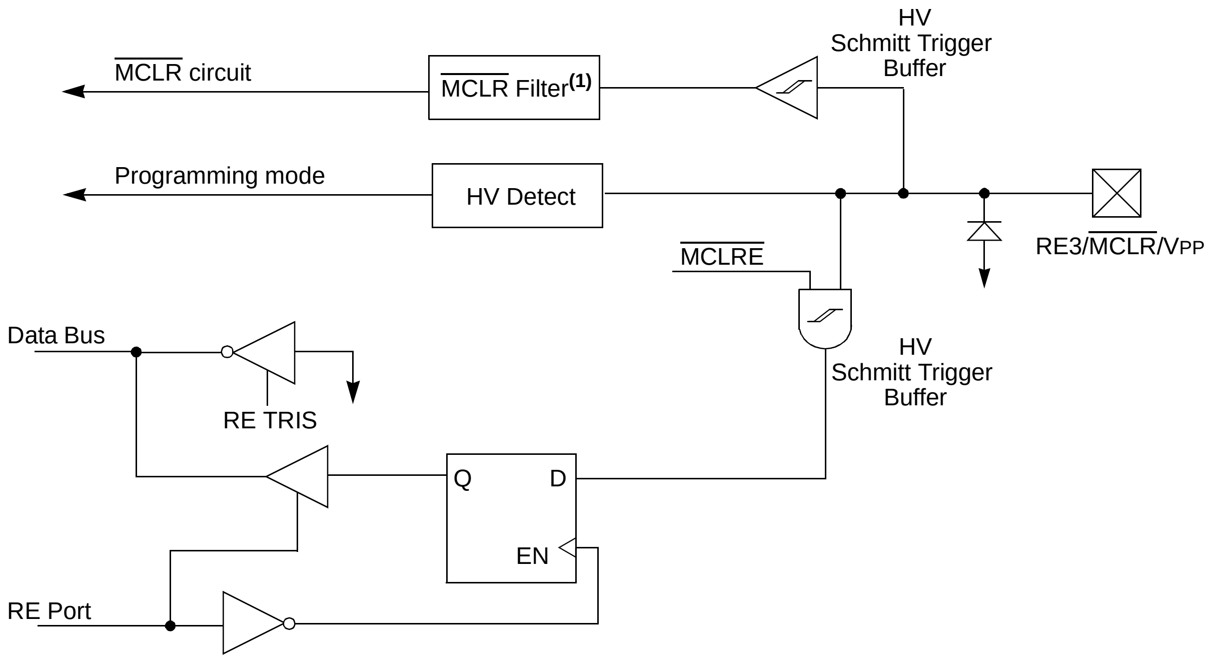


FIGURE 4-27: BLOCK DIAGRAM OF RE3/$\overline{\text{MCLR}}$/VPP


HV
Schmitt Trigger
Buffer
$\overline{\text{MCLR}}$ circuit
$\overline{\text{MCLR}}$ Filter(1)
Programming mode
HV Detect
$\overline{\text{MCLRE}}$
RE3/$\overline{\text{MCLR}}$/VPP
Data Bus
RE TRIS
HV
Schmitt Trigger
Buffer
Q
D
EN
RE Port

Note 1: The $\overline{\text{MCLR}}$ filter is bypassed in Emulation mode.

TABLE 4-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	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
09h	PORTE	—	—	—	—	RE3	RE2	RE1	RE0	`---- xxxx`	`---- uuuu`
1Fh	ADCON0	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	`0000 0000`	`0000 0000`
89h	TRISE	—	—	—	—	TRISE3(3)	TRISE2(2)	TRISE1(2)	TRISE0(2)	`---- 1111`	`---- 1111`
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	`1111 1111`	`1111 1111`
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	`0001 0011`	`0001 0011`
11Eh	LCDSE2(1,2)	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	`0000 0000`	`uuuu uuuu`

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTC.

Note 1: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

2: PIC16F914/917 only.

3: Bit is read only; TRISE = 1 always.

NOTES:

5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

> **Note:** Additional information on the Timer0 module is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

5.1 Timer0 Operation

Timer mode is selected by clearing the T0CS bit (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

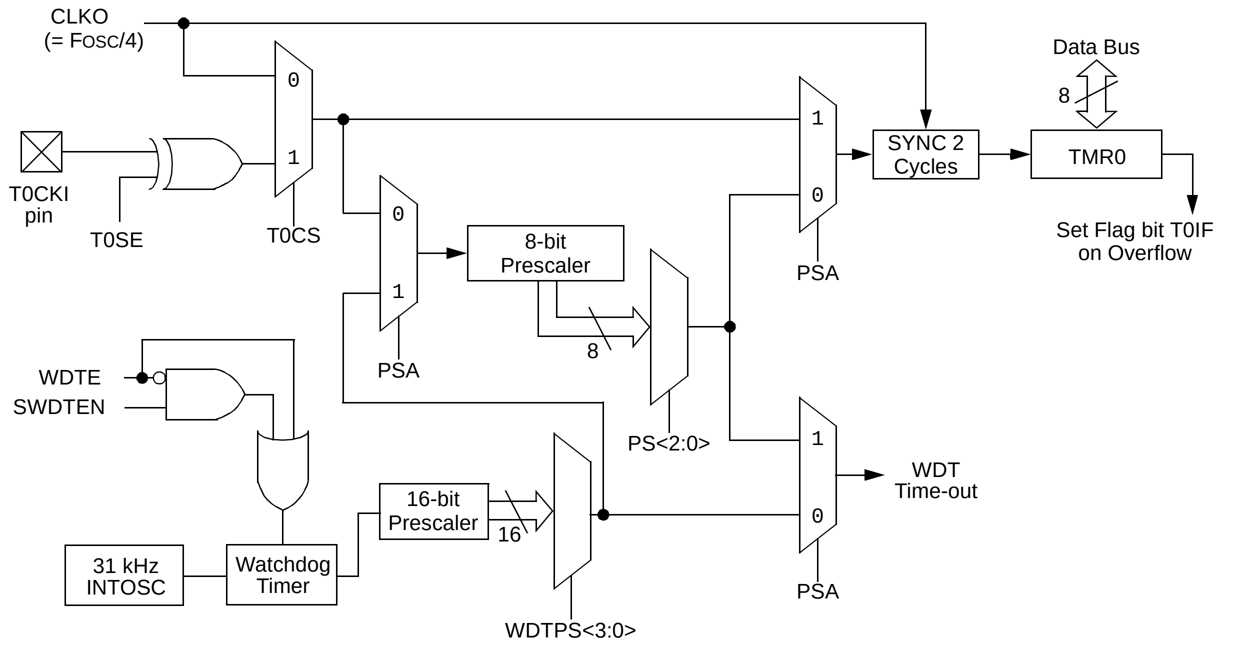
Counter mode is selected by setting the T0CS bit (OPTION_REG<5>). In this mode, the Timer0 module will increment either on every rising or falling edge of pin RA4/C1OUT/T0CKI/SEG4. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION_REG<4>). Clearing the T0SE bit selects the rising edge.

> **Note:** Counter mode has specific external clock requirements. Additional information on these requirements is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

5.2 Timer0 Interrupt

A Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from Sleep, since the timer is shut off during Sleep.

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note 1: T0SE, T0CS, PSA and PS<2:0> are bits in the Option register; WDTPS<3:0> are bits in the WDTCON register.

5.3 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI, with the internal phase clocks, is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

REGISTER 5-1: OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7 bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values in WPUA register

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT/SEG0 pin
0 = Interrupt on falling edge of RB0/INT/SEG0 pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA4/C1OUT/T0CKI/SEG4I pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/C1OUT/T0CKI/SEG4 pin
0 = Increment on low-to-high transition on RA4/C1OUT/T0CKI/SEG4 pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate[1]
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Note 1: A dedicated 16-bit WDT postscaler is available for the PIC16F91X. See **Section 16.6 “Watchdog Timer (WDT)”** for more information.

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

5.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. For simplicity, this counter will be referred to as “prescaler” throughout this data sheet. The prescaler assignment is controlled in software by the control bit PSA (OPTION_REG<3>). Clearing the PSA bit will assign the prescaler to Timer0. Prescale values are selectable via the PS<2:0> bits (OPTION_REG<2:0>).

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1,  MOVWF 1, BSF 1,  x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

5.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 5-1 and Example 5-2) must be executed when changing the prescaler assignment from Timer0 to WDT.

EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)

```
BCF     STATUS,RP0      ;Bank 0
CLRWDT                  ;Clear WDT
CLRF    TMR0            ;Clear TMR0 and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’00101111’     ;Required if desired
MOVWF   OPTION_REG      ; PS2:PS0 is
CLRWDT                  ; 000 or 001
                        ;
MOVLW   b’00101xxx’     ;Set postscaler to
MOVWF   OPTION_REG      ; desired WDT rate
BCF     STATUS,RP0      ;Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 5-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)

```
CLRWDT                  ;Clear WDT and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’xxxx0xxx’     ;Select TMR0,
                        ; prescale, and
                        ; clock source
MOVWF   OPTION_REG      ;
BCF     STATUS,RP0      ;Bank 0
```

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	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
01h	TMR0	Timer0 Module register								xxxx xxxx	uuuu uuuu
0Bh/10Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

Legend: - = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

NOTES:

6.0 TIMER1 MODULE WITH GATE CONTROL

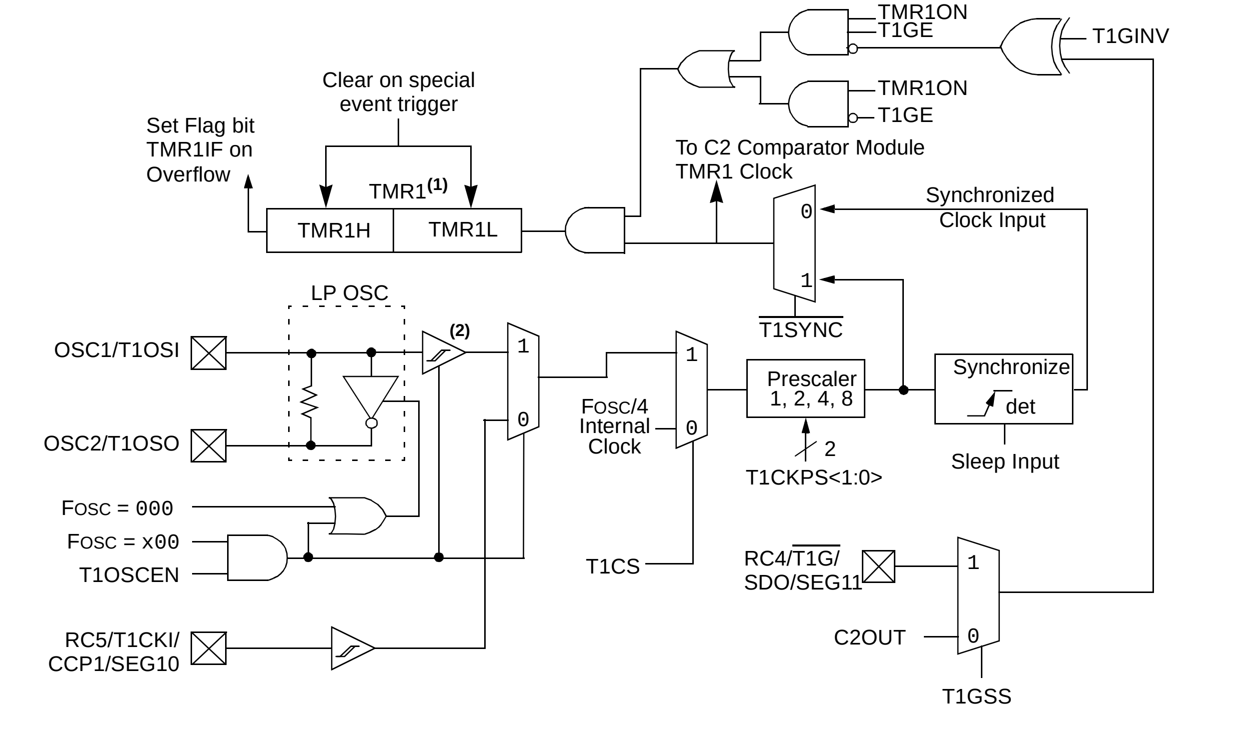
The PIC16F91X has a 16-bit timer. Figure 6-1 shows the basic block diagram of the Timer1 module. Timer1 has the following features:

- 16-bit timer/counter (TMR1H:TMR1L)
- Readable and writable
- Internal or external clock selection
- Synchronous or asynchronous operation
- Interrupt-on-overflow from FFFFh to 0000h
- Wake-up upon overflow (Asynchronous mode)
- Optional external enable input
  - Selectable gate source: $\overline{\text{T1G}}$ or C2 output (T1GSS)
  - Selectable gate polarity (T1GINV)
- Optional LP oscillator

The Timer1 Control register (T1CON), shown in Register 6-1, is used to enable/disable Timer1 and select the various features of the Timer1 module.

Note:	Additional information on timer modules is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

FIGURE 6-1: TIMER1 ON THE PIC16F91X BLOCK DIAGRAM



Note 1: Timer1 increments on the rising edge.

2: ST Buffer is low-power type when using LP oscillator or high-speed type when using T1CKI.

6.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit timer with prescaler
- 16-bit synchronous counter
- 16-bit asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

In the Timer1 module, the module clock can be gated by the Timer1 gate, which can be selected as either the $\overline{\text{T1G}}$ pin or Comparator 2 output.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKO), Timer1 can use the LP oscillator as a clock source.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge.

6.2 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 Interrupt Flag bit (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 Interrupt Enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>)

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note: The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

6.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

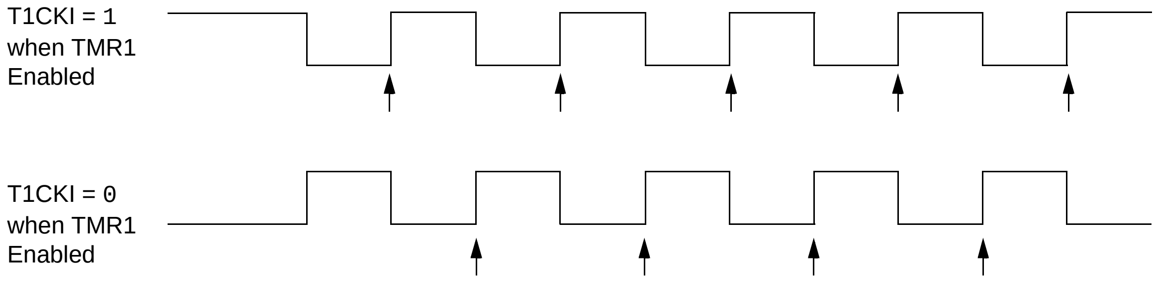
6.4 Timer1 Gate

Timer1 gate source is software configurable to be the $\overline{\text{T1G}}$ pin or the output of Comparator 2. This allows the device to directly time external events using $\overline{\text{T1G}}$ or analog events using Comparator 2. See CMCON1 (Register 8-2) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site (www.microchip.com).

Note: T1GE bit (T1CON<6>) must be set to use either $\overline{\text{T1G}}$ or C2OUT as the Timer1 gate source. See Register 8-2 for more information on selecting the Timer1 gate source.

Timer1 gate can be inverted using the T1GINV bit (T1CON<7>), whether it originates from the $\overline{\text{T1G}}$ pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

FIGURE 6-2: TIMER1 INCREMENTING EDGE



Note 1: Arrows indicate counter increments.

2: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

REGISTER 6-1: T1CON – TIMER1 CONTROL REGISTER (ADDRESS: 10h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON

bit 7 bit 0

bit 7 **T1GINV:** Timer1 Gate Invert bit[(1)]
1 = Timer1 gate is inverted
0 = Timer1 gate is not inverted

bit 6 **T1GE:** Timer1 Gate Enable bit[(2)]
If TMR1ON = 0:
This bit is ignored.
If TMR1ON = 1:
1 = Timer1 gate is enabled
0 = Timer1 gate is disabled

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale Value
10 = 1:4 Prescale Value
01 = 1:2 Prescale Value
00 = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
If INTOSC without CLKO oscillator is active:
1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off
Else:
This bit is ignored.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from RC5/T1CKI/CCP1/SEG10 pin or T1OSC (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

Note 1: T1GINV bit inverts the Timer1 gate logic, regardless of source.

2: T1GE bit must be set to use either $\overline{\text{T1G}}$ pin or C2OUT, as selected by the T1GSS bit (CMCON1<1>), as a Timer1 gate source.

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

6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt-on-overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 6.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	The ANSEL (91h) and CMCON0 (9Ch) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

6.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L, while the timer is running from an external asynchronous clock, will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Examples in the "*PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

6.6 TIMER1 OSCILLATOR

To minimize the multiplexing of peripherals on the I/O ports, the dedicated TMR1 oscillator, which is normally used for TMR1 real time clock applications, is eliminated. Instead, the TMR1 module can enable the LP oscillator.

If the microcontroller is programmed to run from INTOSC with no CLKO or LP oscillator:

1. Setting the T1OSCEN bit to '1' will enable the LP oscillator to clock TMR1 while the microcontroller is clocked from either the INTOSC or LP oscillator. Note that the T1OSC and LP oscillators share the same circuitry. Therefore, when LP oscillator is selected and T1OSC is enabled, both the microcontroller and the Timer1 module share the same clock source.
2. Sleep mode does not shut off the LP oscillator operation (i.e., if the INTOSC oscillator runs the microcontroller, and T1OSCEN = 1 (TMR1 is running from the LP oscillator), then the LP oscillator will continue to run during Sleep mode.

In all oscillator modes **except** for INTOSC with no CLKOUT and LP, the T1OSC enable option is unavailable and is ignored.

Note:	When INTOSC without CLKO oscillator is selected and T1OSCEN = 1, the LP oscillator will run continuously independent of the TMR1ON bit.

6.7 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 or CCP2 module is configured in Compare mode to generate a "special event trigger" (CCP1M<3:0> = 1011), this signal will reset Timer1.

Note:	The special event triggers from the CCP1 and CCP2 modules will not set interrupt flag bit, TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1 or CCP2, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL register pair effectively becomes the period register for Timer1.

6.8 Resetting of Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR, or any other Reset, except by the CCP1 and CCP2 special event triggers.

T1CON register is reset to 00h on a Power-on Reset, or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

6.9 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- Timer1 must be on (T1CON<0>)
- TMR1IE bit (PIE1<0>) must be set
- PEIE bit (INTCON<6>) must be set

The device will wake-up on an overflow. If the GIE bit (INTCON<7>) is set, the device will wake-up and jump to the Interrupt Service Routine (0004h) on an overflow. If the GIE bit is clear, execution will continue with the next instruction.

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER1

Addr	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
0Bh/ 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as ‘0’. Shaded cells are not used by the Timer1 module.

7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

Timer2 has a control register shown in Register 7-1. TMR2 can be shut-off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption. Figure 7-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

7.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 register is readable and writable, and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPSx (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 7-1: T2CON – TIMER2 CONTROL REGISTER (ADDRESS: 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscale Select bits

0000 =1:1 Postscale
0001 =1:2 Postscale
•
•
•
1111 =1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 =Prescaler is 1
01 =Prescaler is 4
1x =Prescaler is 16

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

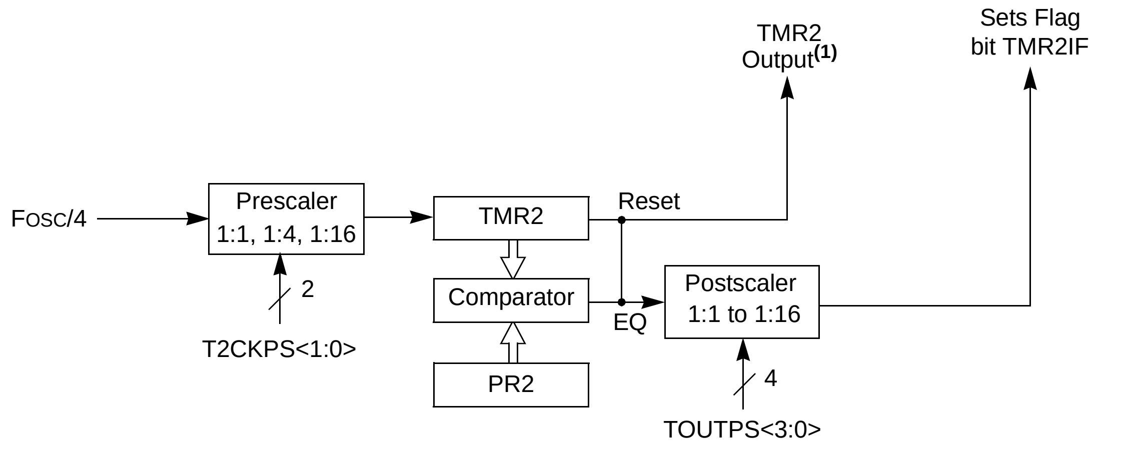
7.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

7.3 Timer2 Output

The output of TMR2 (before the postscaler) is fed to the SSP module, which optionally uses it to generate the shift clock.

FIGURE 7-1: TIMER2 BLOCK DIAGRAM



Note 1: TMR2 register output can be software selected by the SSP module as a baud clock.

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2

Addr	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
0Bh/ 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
11h	TMR2	Holding register for the 8-bit TMR2 register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
92h	PR2	Timer2 Period register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer2 module.

NOTES:

8.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with I/O port pins RA<3:0>, while the outputs are multiplexed to pins RA<5:4>. An on-chip Comparator Voltage Reference (CVREF) can also be applied to the inputs of the comparators.

The CMCON0 register (Register 8-1) controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 8-3.

REGISTER 8-1: CMCON0 – COMPARATOR CONFIGURATION REGISTER (ADDRESS: 9Ch)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7							bit 0

bit 7 **C2OUT:** Comparator 2 Output bit
When C2INV = `0`;
`1` = C2 VIN+ > C2 VIN-
`0` = C2 VIN+ < C2 VIN-
When C2INV = `1`;
`0` = C2 VIN+ > C2 VIN-
`1` = C2 VIN+ < C2 VIN-

bit 6 **C1OUT:** Comparator 1 Output bit
When C1INV = `0`;
`1` = C1 VIN+ > C1 VIN-
`0` = C1 VIN+ < C1 VIN-
When C1INV = `1`;
`0` = C1 VIN+ > C1 VIN-
`1` = C1 VIN+ < C1 VIN-

bit 5 **C2INV:** Comparator 2 Output Inversion bit
`1` = C2 Output inverted
`0` = C2 Output not inverted

bit 4 **C1INV:** Comparator 1 Output Inversion bit
`1` = C1 Output inverted
`0` = C1 Output not inverted

bit 3 **CIS:** Comparator Input Switch bit
When CM<2:0> = `010`:
`1` = C1 VIN- connects to RA3/AN3/C1+/VREF+/SEG15
C2 VIN- connects to RA2/AN2/C2+/VREF-/COM2
`0` = C1 VIN- connects to RA0/AN0/C1-/SEG12
C2 VIN- connects to RA1/AN1/C2-/SEG7
When CM<2:0> = `001`:
`1` = C1 VIN- connects to RA3/AN3/C1+/VREF+/SEG15
`0` = C1 VIN- connects to RA0/AN0/C1-/SEG12

bit 2-0 **CM<2:0>:** Comparator Mode bits[(1)]
See Figure 8-3 for comparator modes and CM<2:0> bit settings.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

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

8.1 Comparator Operation

A single comparator is shown in Figure 8-1 along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 8-1 represent the uncertainty due to input offsets and response time.

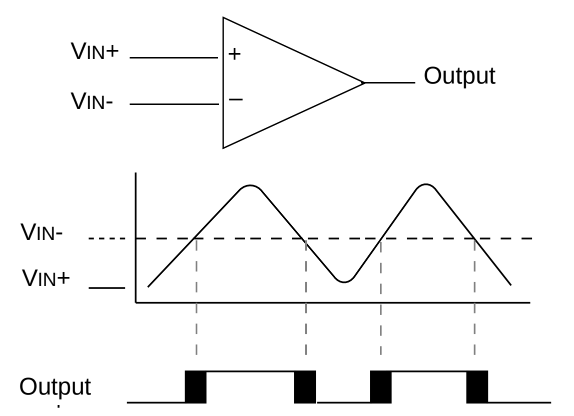
Note:	To use CIN+ and CIN- pins as analog inputs, the appropriate bits must be programmed in the CMCON0 (9Ch) register.

The polarity of the comparator output can be inverted by setting the CxINV bits (CMCON0<5:4>). Clearing CxINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-1.

TABLE 8-1: OUTPUT STATE VS. INPUT CONDITIONS

Input Conditions	CINV	CxOUT
VIN- > VIN+	0	0
VIN- < VIN+	0	1
VIN- > VIN+	1	1
VIN- < VIN+	1	0

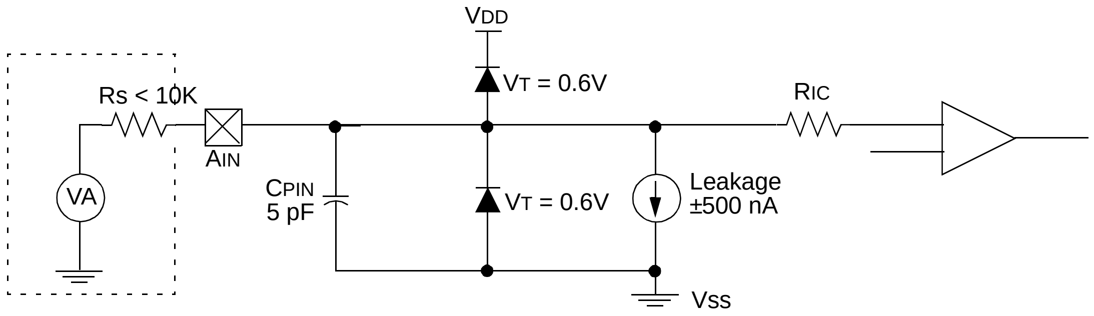
FIGURE 8-1: SINGLE COMPARATOR



8.2 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 8-2. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage.

FIGURE 8-2: ANALOG INPUT MODEL



Legend: CPIN= Input Capacitance
VT = Threshold Voltage
ILEAKAGE= Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage

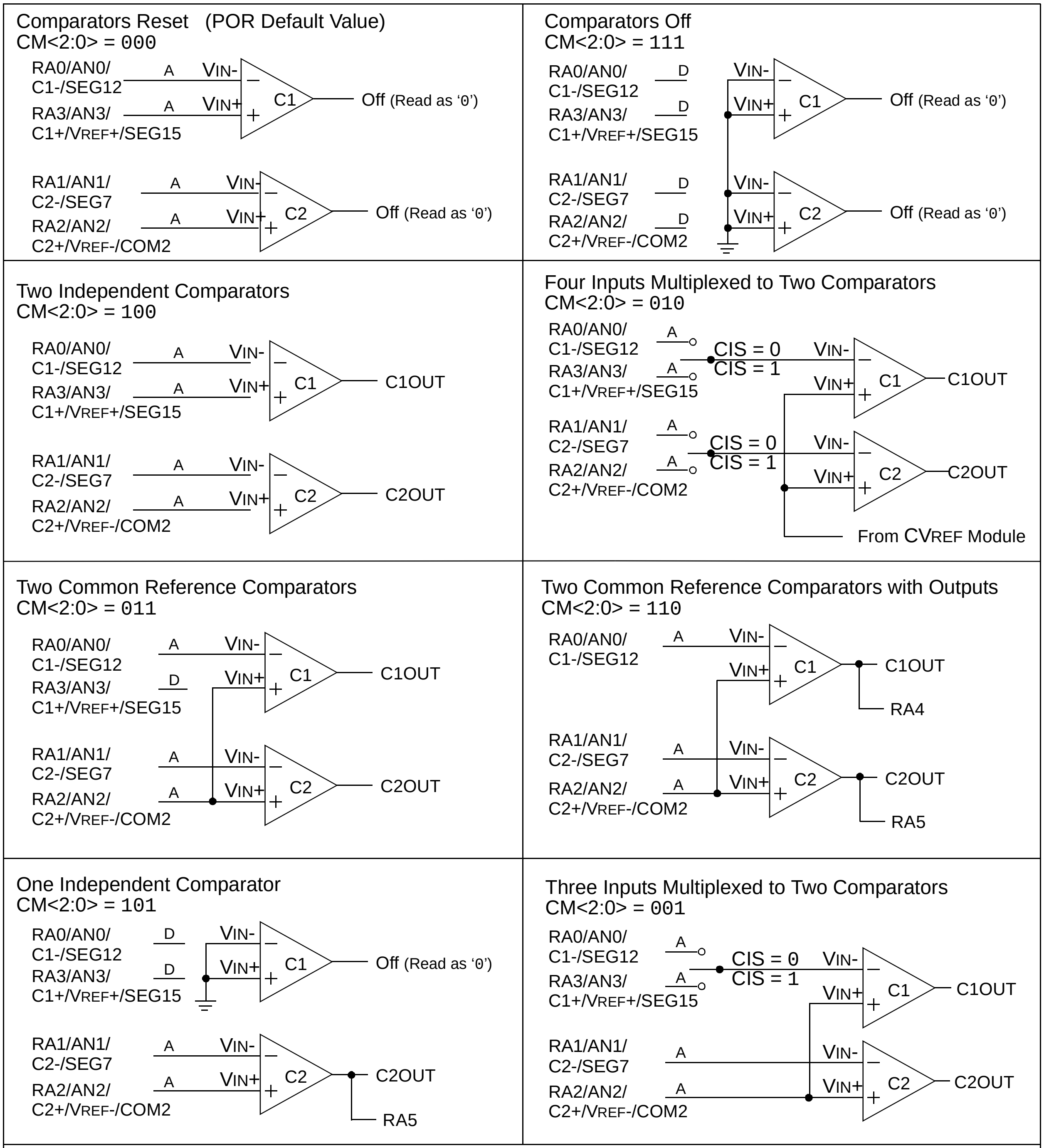
8.3 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON0 register is used to select these modes. Figure 8-3 shows the eight possible modes.

If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 19.0 “Electrical Specifications”**.

Note:	Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur.

FIGURE 8-3: COMPARATOR I/O OPERATING MODES



Legend:

A = Analog Input, port reads zeros always. D = Digital Input. CIS (CMCON0<3>) is the Comparator Input Switch.

FIGURE 8-4: MODIFIED COMPARATOR C1 OUTPUT BLOCK DIAGRAM

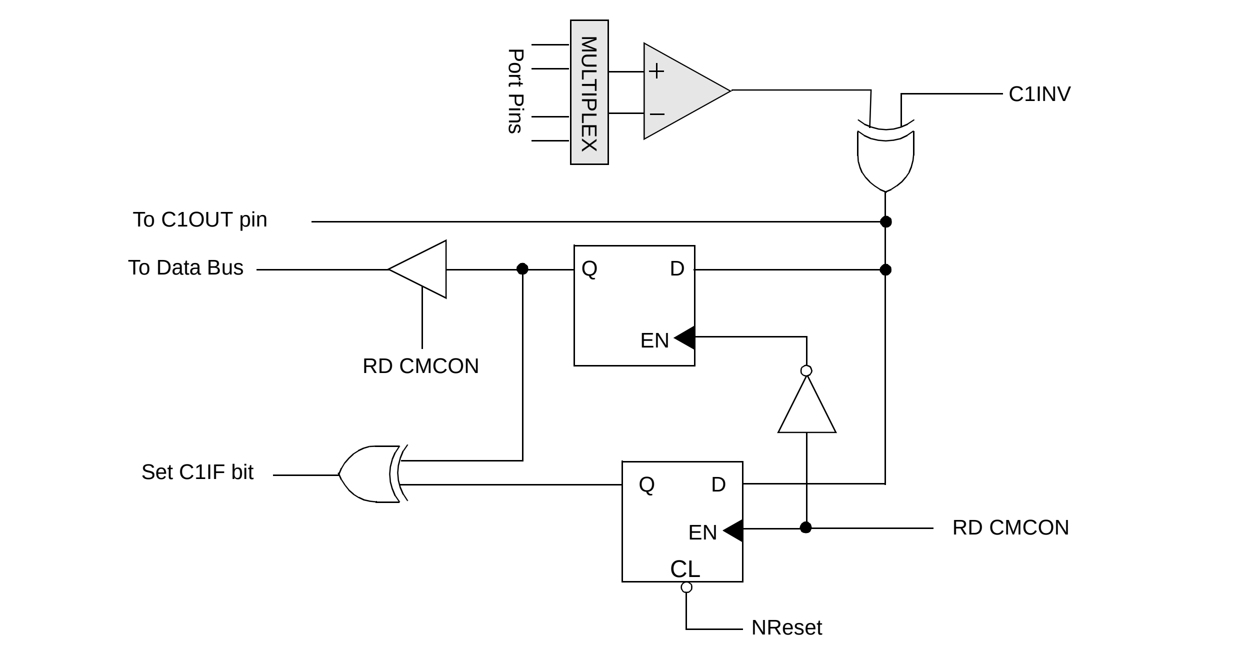
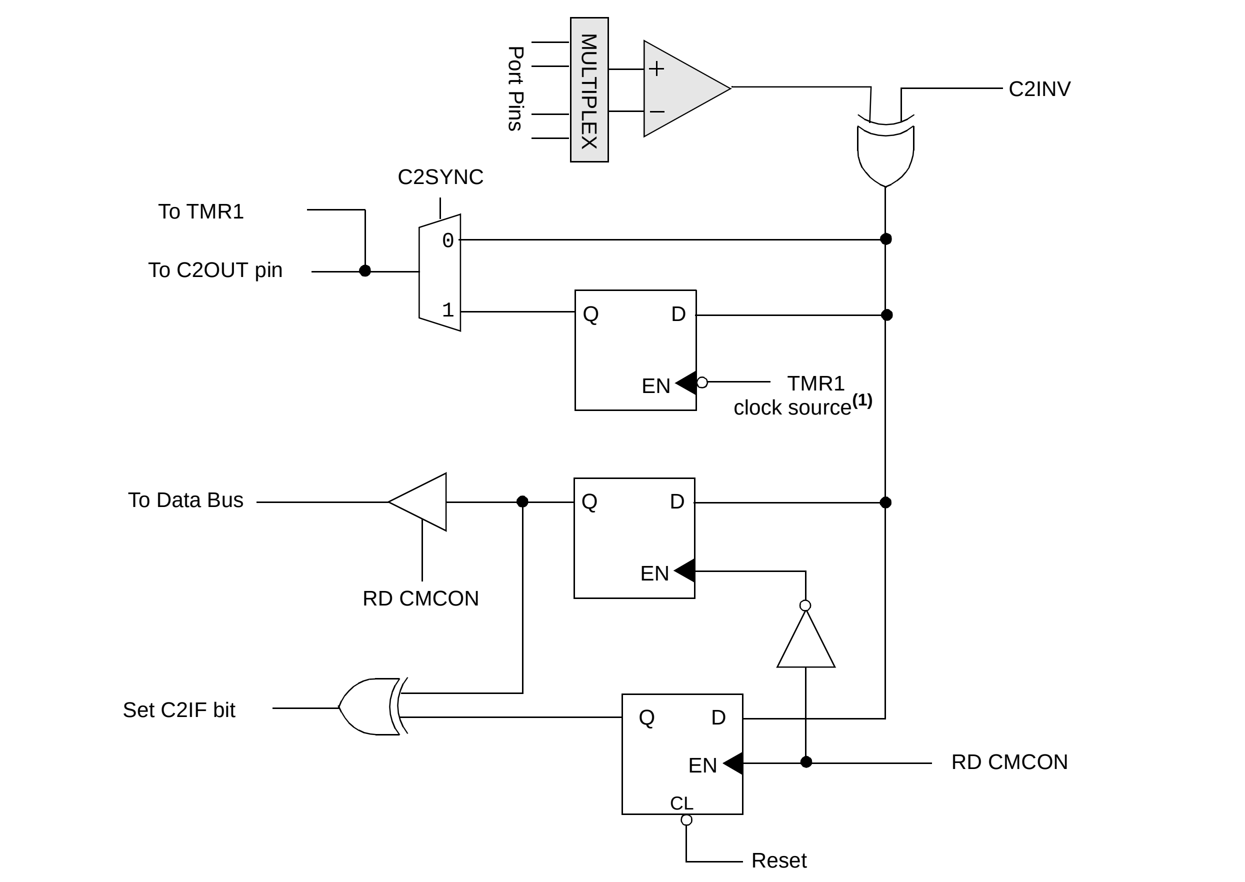


FIGURE 8-5: COMPARATOR C2 OUTPUT BLOCK DIAGRAM



Note 1: Comparator 2 output is latched on falling edge of T1 clock source.

REGISTER 8-2: CMCON1 – COMPARATOR CONFIGURATION REGISTER (ADDRESS: 97h)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
—	—	—	—	—	—	T1GSS	C2SYNC
bit 7							bit 0

bit 7-2: **Unimplemented**: Read as ‘0’

bit 1 **T1GSS:** Timer1 Gate Source Select bit
1 = Timer1 gate source is $\overline{\text{T1G}}$ pin (RC4 must be configured as digital input)
0 = Timer1 gate source is Comparator 2 Output

bit 0 **C2SYNC:** Comparator 2 Synchronize bit
1 = C2 output synchronized with falling edge of Timer1 clock
0 = C2 output not synchronized with Timer1 clock

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

8.4 Comparator Outputs

The comparator outputs are read through the CMCON0 register. These bits are read-only. The comparator outputs may also be directly output to the RA4 and RA5 I/O pins. When enabled, multiplexers in the output path of the RA4 and RA5 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 8-4 and Figure 8-5 show the output block diagram for Comparator 1 and 2.

The TRIS bits will still function as an output enable/disable for the RA4 and RA5 pins while in this mode.

The polarity of the comparator outputs can be changed using the C1INV and C2INV bits (CMCON0<5:4>).

Timer1 gate source can be configured to use the $\overline{\text{T1G}}$ pin or Comparator 2 output as selected by the T1GSS bit (CMCON1<1>). This feature can be used to time the duration or interval of analog events. The output of Comparator 2 can also be synchronized with Timer1 by setting the C2SYNC bit (CMCON1<0>). When enabled, the output of Comparator 2 is latched on the falling edge of Timer1 clock source. If a prescaler is used with Timer1, Comparator 2 is latched after the prescaler. To prevent a race condition, the Comparator 2 output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See (Figure 8-5), Comparator 2 Block Diagram and (Figure 6-1), Timer1 Block Diagram for more information.

It is recommended to synchronize Comparator 2 with Timer1 by setting the C2SYNC bit when Comparator 2 is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if Comparator 2 changes during an increment.

8.5 Comparator Interrupts

The comparator interrupt flags are set whenever there is a change in the output value of its respective comparator. Software will need to maintain information about the status of the output bits, as read from CMCON0<7:6>, to determine the actual change that has occurred. The CxIF bits, PIR2<6:5>, are the Comparator Interrupt flags. This bit must be reset in software by clearing it to ‘0’. Since it is also possible to write a ‘1’ to this register, a simulated interrupt may be initiated.

The CxIE bits (PIE2<6:5>) and the PEIE bit (INTCON<6>) must be set to enable the interrupts. In addition, the GIE bit must also be set. If any of these bits are cleared, the interrupt is not enabled, though the CxIF bits will still be set if an interrupt condition occurs.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON0. This will end the mismatch condition.

b) Clear flag bit CxIF

A mismatch condition will continue to set flag bit CxIF. Reading CMCON0 will end the mismatch condition and allow flag bits CxIF to be cleared.

Note: If a change in the CMCON0 register (CxOUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CxIF (PIR2<6:5>) interrupt flag may not get set.

8.6 Comparator Reference

The comparator module also allows the selection of an internally generated voltage reference for one of the comparator inputs. The VRCON register, Register 8-3, controls the voltage reference module shown in Figure 8-6.

8.6.1 CONFIGURING THE VOLTAGE REFERENCE

The voltage reference can output 32 distinct voltage levels; 16 in a high range and 16 in a low range.

The following equation determines the output voltages:

EQUATION 8-1:

VRR = `1` *(low range): CVREF = (VR3:VR0/24) x VDD*

VRR = `0` *(high range):*

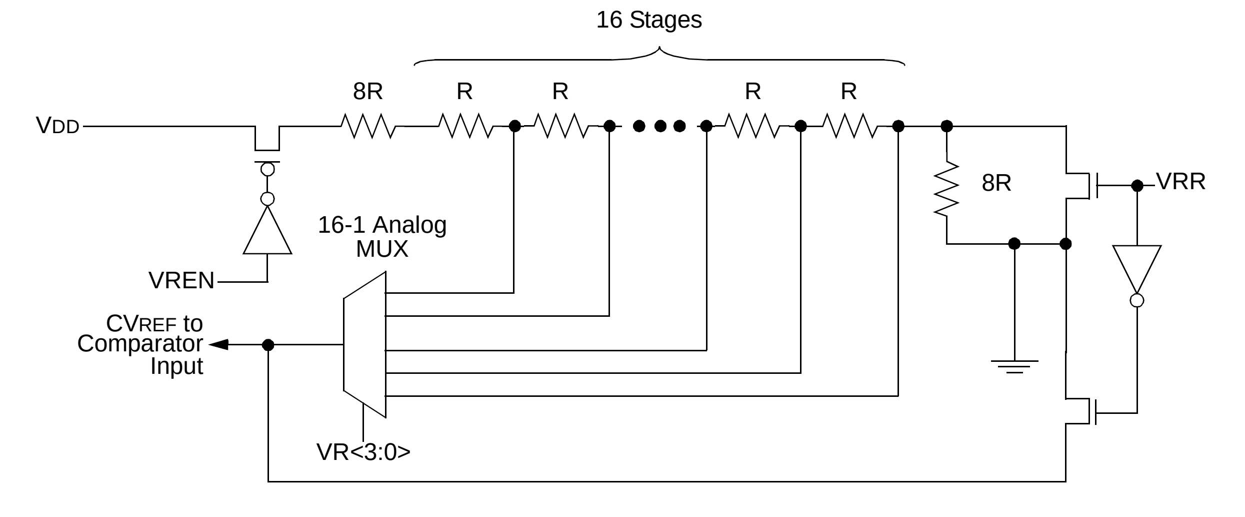
CVREF = (VDD/4) + (VR3:VR0 x VDD/32)

8.6.2 VOLTAGE REFERENCE ACCURACY/ERROR

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-6) keep CVREF from approaching VSS or VDD. The exception is when the module is disabled by clearing the VREN bit (VRCON<7>). When disabled, the reference voltage is VSS when VR<3:0> = `0000`. This allows the comparators to detect a zero-crossing and not consume CVREF module current.

The voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the comparator voltage reference can be found in **Section 19.0 "Electrical Specifications"**.

FIGURE 8-6: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



8.7 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is ensured to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (Table 19-10).

8.8 Operation During Sleep

The comparators and voltage reference, if enabled before entering Sleep mode, remain active during Sleep. This results in higher Sleep currents than shown in the power-down specifications. The additional current consumed by the comparator and the voltage reference is shown separately in the specifications. To minimize power consumption while in Sleep mode, turn off the comparator, CM<2:0> = `111`, and voltage reference, VRCON<7> = `0`.

While the comparator is enabled during Sleep, an interrupt will wake-up the device. If the GIE bit (INTCON<7>) is set, the device will jump to the interrupt vector (0004h), and if clear, continues execution with the next instruction. If the device wakes up from Sleep, the contents of the CMCON0, CMCON1 and VRCON registers are not affected.

8.9 Effects of a Reset

A device Reset forces the CMCON0, CMCON1 and VRCON registers to their Reset states. This forces the comparator module to be in the Comparator Reset mode, CM<2:0> = `000` and the voltage reference to its off state. Thus, all potential inputs are analog inputs with the comparator and voltage reference disabled to consume the smallest current possible.

REGISTER 8-3: VRCON – VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS: 9Dh)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	—	VRR	—	VR3	VR2	VR1	VR0

bit 7 bit 0

bit 7 **VREN:** CVREF Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down, no IDD drain and CVREF = VSS.

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **VRR:** CVREF Range Selection bit
1 = Low range
0 = High range

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **VR<3:0>:** CVREF value selection 0 ≤ VR<3:0> ≤ 15
When VRR = 1: CVREF = (VR<3:0>/24) * VDD
When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) * VDD

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

TABLE 8-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	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
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
9Ch	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as ‘0’. Shaded cells are not used by the comparator or Comparator Voltage Reference module.

9.0 LIQUID CRYSTAL DISPLAY (LCD) DRIVER MODULE

The Liquid Crystal Display (LCD) driver module generates the timing control to drive a static or multiplexed LCD panel. In the PIC16F914/917 devices (PIC16F914/917), the module drives the panels of up to four commons and up to 24 segments and in the PIC16F913/916 devices (PIC16F913/916), the module drives the panels of up to four commons and up to 16 segments. It also provides control of the LCD pixel data.

The LCD driver module supports:

- Direct driving of LCD panel
- Three LCD clock sources with selectable prescaler
- Up to four commons:
  - Static
  - 1/2 multiplex
  - 1/3 multiplex
  - 1/4 multiplex
- Up to 24 (in PIC16F914/917 devices)/16 (in PIC16F913/916 devices) segments
- Static, 1/2 or 1/3 LCD Bias

The module has 32 registers:

- LCD Control Register (LCDCON)
- LCD Phase Register (LCDPS)
- Three LCD Segment Enable Registers (LCDSE<2:0>)
- 24 LCD Data Registers (LCDDATA<11:0>)

The LCDCON register, shown in Register 9-1, controls the operation of the LCD driver module. The LCDPS register, shown in Register 9-2, configures the LCD clock source prescaler and the type of waveform; Type-A or Type-B. The LCDSE<2:0> registers configure the functions of the port pins:

• LCDSE0	SE<7:0>
• LCDSE1	SE<15:8>
• LCDSE2	SE<23:16>

As an example, LCDSEn is detailed in Register 9-3.

Note: The LCDSE2 register is not implemented in PIC16F913/916 devices.

Once the module is initialized for the LCD panel, the individual bits of the LCDDATA<11:0> registers are cleared/set to represent a clear/dark pixel, respectively:

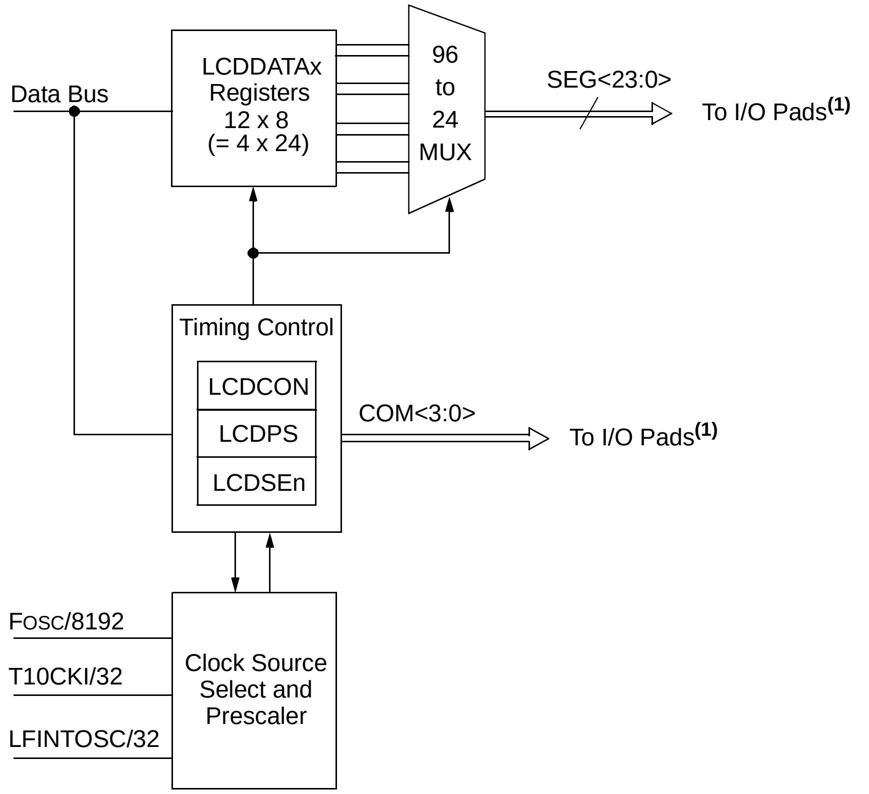
• LCDDATA0	SEG7COM0:SEG0COM0
• LCDDATA1	SEG15COM0:SEG8COM0
• LCDDATA2	SEG23COM0:SEG16COM0
• LCDDATA3	SEG7COM1:SEG0COM1
• LCDDATA4	SEG15COM1:SEG8COM1
• LCDDATA5	SEG23COM1:SEG16COM1
• LCDDATA6	SEG7COM2:SEG0COM2
• LCDDATA7	SEG15COM2:SEG8COM2
• LCDDATA8	SEG23COM2:SEG16COM2
• LCDDATA9	SEG7COM3:SEG0COM3
• LCDDATA10	SEG15COM3:SEG8COM3
• LCDDATA11	SEG23COM3:SEG16COM3

As an example, LCDDATAx is detailed in Register 9-4.

Once the module is configured, the LCDEN (LCDCON<7>) bit is used to enable or disable the LCD module. The LCD panel can also operate during Sleep by clearing the SLPEN (LCDCON<6>) bit.

Note: Writing into the registers LCDDATA2, LCDDATA5, LCDDATA8 and LCDDATA11 in PIC16F913/916 devices will not affect the status of any pixel and these registers can be used as General Purpose Registers.

FIGURE 9-1: LCD DRIVER MODULE BLOCK DIAGRAM



Note 1: These are not directly connected to the I/O pads. See **Section 4.0 "I/O Ports"** for more detail.

REGISTER 9-1: LCDCON – LIQUID CRYSTAL DISPLAY CONTROL REGISTER (ADDRESS: 107h)

R/W-0	R/W-0	R/C-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1
LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0
bit 7							bit 0

bit 7 **LCDEN:** LCD Driver Enable bit

1 = LCD driver module is enabled
0 = LCD driver module is disabled

bit 6 **SLPEN:** LCD Driver Enable in Sleep mode bit

1 = LCD driver module is disabled in Sleep mode
0 = LCD driver module is enabled in Sleep mode

bit 5 **WERR:** LCD Write Failed Error bit

1 = LCDDATAx register written while LCDPS<WA> = 0 (must be cleared in software)
0 = No LCD write error

bit 4 **VLCDEN:** LCD Bias Voltage Pins Enable bit

1 = VLCD pins are enabled
0 = VLCD pins are disabled

bit 3-2 **CS<1:0>:** Clock Source Select bits

00 = FOSC/8192
01 = T13CKI (Timer1)/32
1x = INTOSC (31.25 kHz)/32

bit 1-0 **LMUX<1:0>:** Commons Select bits

LMUX<1:0>	Multiplex	Maximum Number of Pixels (PIC16F913/916)	Maximum Number of Pixels (PIC16F914/917)	Bias
00	Static (COM0)	16	48	Static
01	1/2 (COM<1:0>)	32	96	1/2 or 1/3
10	1/3 (COM<2:0>)	48	144	1/2 or 1/3
11	1/4 (COM<3:0>)	60[1]	192	1/3

Note 1: On PIC16F913/916 devices, COM3 and SEG15 are shared on one pin, limiting the device from driving 64 pixels.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
C = Only clearable bit	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown
- n = Value at POR			

REGISTER 9-2: LCDPS – LCD PRESCALER SELECT REGISTER (ADDRESS: 108h)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0
bit 7							bit 0

bit 7 **WFT:** Waveform Type Select bit

1 = Type-B waveform (phase changes on each frame boundary)
0 = Type-A waveform (phase changes within each common type)

bit 6 **BIASMD:** Bias Mode Select bit

When LMUX<1:0> = 00:
0 = Static Bias mode (do not set this bit to ‘1’)

When LMUX<1:0> = 01:
1 = 1/2 Bias mode
0 = 1/3 Bias mode

When LMUX<1:0> = 10:
1 = 1/2 Bias mode
0 = 1/3 Bias mode

When LMUX<1:0> = 11:
0 = 1/3 Bias mode (do not set this bit to ‘1’)

bit 5 **LCDA:** LCD Active Status bit

1 = LCD driver module is active
0 = LCD driver module is inactive

bit 4 **WA:** LCD Write Allow Status bit

1 = Write into the LCDDATAx registers is allowed
0 = Write into the LCDDATAx registers is not allowed

bit 3-0 **LP<3:0>:** LCD Prescaler Select bits

1111 = 1:16
1110 = 1:15
1101 = 1:14
1100 = 1:13
1011 = 1:12
1010 = 1:11
1001 = 1:10
1000 = 1:9
0111 = 1:8
0110 = 1:7
0101 = 1:6
0100 = 1:5
0011 = 1:4
0010 = 1:3
0001 = 1:2
0000 = 1:1

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

REGISTER 9-3: LCDSEn – LCD SEGMENT REGISTERS (ADDRESS: 11Ch, 11Dh, OR 11Eh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEn	SEn	SEn	SEn	SEn	SEn	SEn	SEn
bit 7							bit 0

bit 7-0 **SEn:** Segment Enable bits

1 = Segment function of the pin is enabled
0 = I/O function of the pin is enabled

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

REGISTER 9-4: LCDDATAx – LCD DATA REGISTERS (ADDRESS: 110h-119h, 11Ah, 11Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy
bit 7							bit 0

bit 7-0 **SEGx-COMy:** Pixel On bits

1 = Pixel on (dark)
0 = Pixel off (clear)

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

9.1 LCD Clock Source Selection

The LCD driver module has 3 possible clock sources:

- FOSC/8192
- T13CKI-Clock/32
- INTOSC/32

The first clock source is the system clock divided by 8192 (FOSC/8192). This divider ratio is chosen to provide about 1 kHz output when the system clock is 8 MHz. The divider is not programmable. Instead, the LCD prescaler bits, LCDPS<3:0>, are used to set the LCD frame clock rate.

The second clock source is the Timer1 oscillator/32. This also gives about 1 kHz when a 32.768 kHz crystal is used with the Timer1 oscillator. To use the Timer1 oscillator as a clock source, the T1OSCEN (T1CON<3>) bit should be set.

The third clock source is a 31.25 kHz internal RC oscillator/32 (LFINTOSC), which provides approximately 1 kHz output.

The second and third clock sources may be used to continue running the LCD while the processor is in Sleep.

Using the bits, CS<1:0> (LCDCON<3:2>), any of these clock sources can be selected.

9.1.1 LCD PRESCALER

A 16-bit counter is available as a prescaler for the LCD clock. The prescaler is not directly readable or writable; its value is set by the LP<3:0> bits (LCDPS<3:0>), which determine the prescaler assignment and prescale ratio.

The prescale values from 1:1 through 1:16.

9.2 LCD Bias Types

The LCD driver module can be configured into three bias types:

- Static Bias (2 voltage levels: VSS and VDD)
- 1/2 Bias (3 voltage levels: VSS, 1/2 VDD and VDD)
- 1/3 Bias (4 voltage levels: VSS, 1/3 VDD, 2/3 VDD and VDD)

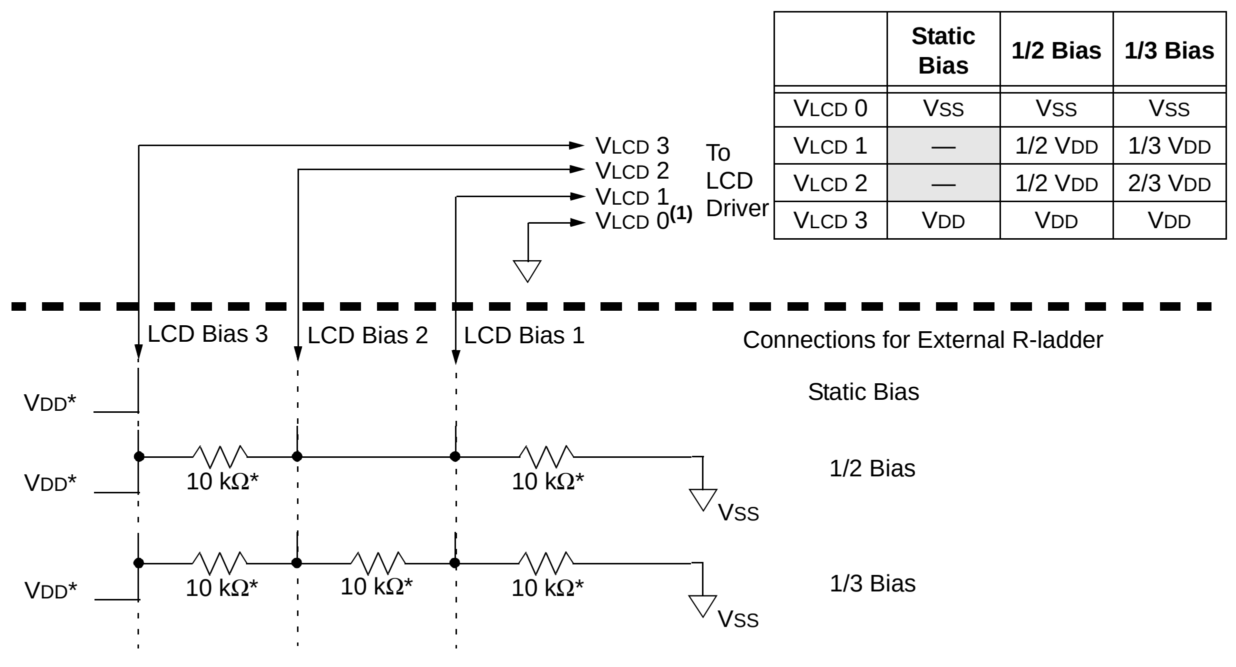
This module uses an external resistor ladder to generate the LCD bias voltages.

The external resistor ladder should be connected to the Bias 1 pin, Bias 2 pin, Bias 3 pin and VSS. The Bias 3 pin should also be connected to VDD.

Figure 9-2 shows the proper way to connect the resistor ladder to the Bias pins.

Note:	VLCD pins used to supply LCD bias voltage are enabled on power-up (POR) and must be disabled by the user by clearing LCDCON<4>, the VLCDEN bit, (see Register 9-1).

FIGURE 9-2: LCD BIAS RESISTOR LADDER CONNECTION DIAGRAM



	Static Bias	1/2 Bias	1/3 Bias
VLCD 0	VSS	VSS	VSS
VLCD 1	—	1/2 VDD	1/3 VDD
VLCD 2	—	1/2 VDD	2/3 VDD
VLCD 3	VDD	VDD	VDD

* These values are provided for design guidance only and should be optimized for the application by the designer.

Note 1: Internal connection.

9.3 LCD Multiplex Types

The LCD driver module can be configured into four multiplex types:

- Static (only COM0 used)
- 1/2 multiplex (COM0 and COM1 are used)
- 1/3 multiplex (COM0, COM1 and COM2 are used)
- 1/4 multiplex (all COM0, COM1, COM2 and COM3 are used)

The LMUX<1:0> setting decides the function of RB5, RA2 or either RA3 or RD0 pins (see Table 9-1 for details).

If the pin is a digital I/O, the corresponding TRIS bit controls the data direction. If the pin is a COM drive, then the TRIS setting of that pin is overridden.

Note: On a Power-on Reset, the LMUX<1:0> bits are '11'.

TABLE 9-1: PORTC<2:0> FUNCTION

LMUX <1:0>	RA3/RD0(1)	RA2	RB5
00	Digital I/O	Digital I/O	Digital I/O
01	Digital I/O	Digital I/O	COM1 Driver
10	Digital I/O	COM2 Driver	COM1 Driver
11	COM3 Driver	COM2 Driver	COM1 Driver

Note 1: RA3 for PIC16F913/916, RD0 for PIC16F914/917

9.4 Segment Enables

The LCDSEn registers are used to select the pin function for each segment pin. The selection allows each pin to operate as either an LCD segment driver or as one of the pin's alternate functions. To configure the pin as a segment pin, the corresponding bits in the LCDSEn registers must be set to '1'.

If the pin is a digital I/O, the corresponding TRIS bit controls the data direction. Any bit set in the LCDSEn registers overrides any bit settings in the corresponding TRIS register.

Note: On a Power-on Reset, these pins are configured as digital I/O.

9.5 Pixel Control

The LCDDATAx registers contain bits which define the state of each pixel. Each bit defines one unique pixel.

Register 9-4 shows the correlation of each bit in the LCDDATAx registers to the respective common and segment signals.

Any LCD pixel location not being used for display can be used as general purpose RAM.

9.6 LCD Frame Frequency

The rate at which the COM and SEG outputs change is called the LCD frame frequency.

TABLE 9-2: FRAME FREQUENCY FORMULAS

Multiplex	Frame Frequency =
Static	Clock source/(4 x 1 x (LP<3:0> + 1))
1/2	Clock source/(2 x 2 x (LP<3:0> + 1))
1/3	Clock source/(1 x 3 x (LP<3:0> + 1))
1/4	Clock source/(1 x 4 x (LP<3:0> + 1))

Note: Clock source is FOSC/8192, T1OSC/32 or LFINTOSC/32.

TABLE 9-3: APPROXIMATE FRAME FREQUENCY (IN Hz) USING Fosc @ 8 MHz, TIMER1 @ 32.768 kHz OR INTOSC

LP<3:0>	Static	1/2	1/3	1/4
2	85	85	114	85
3	64	64	85	64
4	51	51	68	51
5	43	43	57	43
6	37	37	49	37
7	32	32	43	32

FIGURE 9-3: LCD CLOCK GENERATION

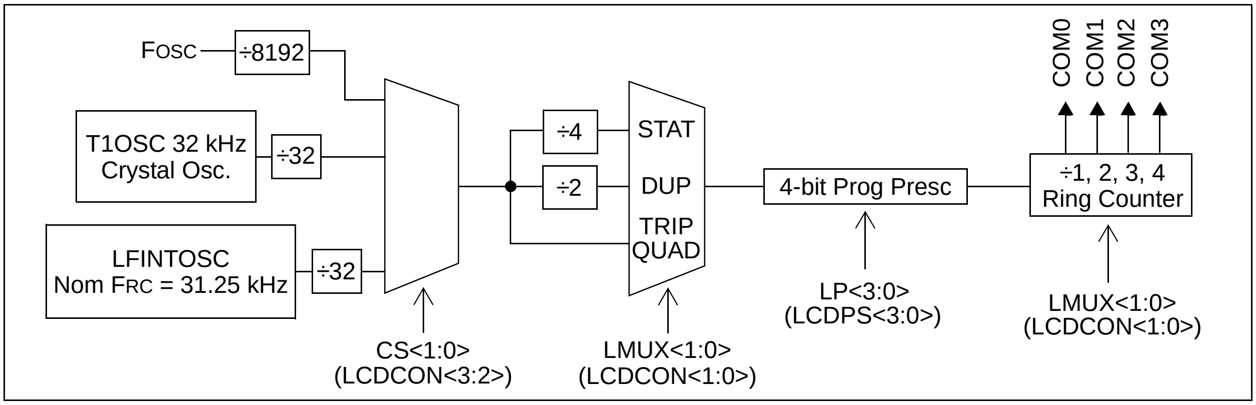


FIGURE 9-4: LCD SEGMENT MAPPING WORKSHEET

LCD Function	COM0		COM1		COM2		COM3		Pin No.	PORT	Alternate Functions
	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment	28/40-pin		
SEG0	LCDDATA0, 0		LCDDATA3, 0		LCDDATA6, 0		LCDDATA9, 0		21/33	RB0	INT
SEG1	LCDDATA0, 1		LCDDATA3, 1		LCDDATA6, 1		LCDDATA9, 1		22/34	RB1	
SEG2	LCDDATA0, 2		LCDDATA3, 2		LCDDATA6, 2		LCDDATA9, 2		23/35	RB2	
SEG3	LCDDATA0, 3		LCDDATA3, 3		LCDDATA6, 3		LCDDATA9, 3		24/36	RB3	
SEG4	LCDDATA0, 4		LCDDATA3, 4		LCDDATA6, 4		LCDDATA9, 4		6/6	RA4	C1OUT/T0CKI
SEG5	LCDDATA0, 5		LCDDATA3, 5		LCDDATA6, 5		LCDDATA9, 5		7/7	RA5	C2OUT/AN4/$\overline{SS}$
SEG6	LCDDATA0, 6		LCDDATA3, 6		LCDDATA6, 6		LCDDATA9, 6		14/18	RC3	
SEG7	LCDDATA0, 7		LCDDATA3, 7		LCDDATA6, 7		LCDDATA9, 7		3/3	RA1	AN1
SEG8	LCDDATA1, 0		LCDDATA4, 0		LCDDATA7, 0		LCDDATA10, 0		18/26	RC7	RX/DT/SDI/SDA
SEG9	LCDDATA1, 1		LCDDATA4, 1		LCDDATA7, 1		LCDDATA10, 1		17/25	RC6	TX/CK/SCK/SCL
SEG10	LCDDATA1, 2		LCDDATA4, 2		LCDDATA7, 2		LCDDATA10, 2		16/24	RC5	T1CKI/CCP1
SEG11	LCDDATA1, 3		LCDDATA4, 3		LCDDATA7, 3		LCDDATA10, 3		15/23	RC4	$\overline{T1G}$/SDO
SEG12	LCDDATA1, 4		LCDDATA4, 4		LCDDATA7, 4		LCDDATA10, 4		2/2	RA0	AN0
SEG13	LCDDATA1, 5		LCDDATA4, 5		LCDDATA7, 5		LCDDATA10, 5		28/40	RB7	ICSPDAT/ICDDAT
SEG14	LCDDATA1, 6		LCDDATA4, 6		LCDDATA7, 6		LCDDATA10, 6		27/39	RB6	ICSPCK/ICDCK
SEG15	LCDDATA1, 7		LCDDATA4, 7		LCDDATA7, 7		LCDDATA10, 7		5/5	RA3	AN3/VREF+
SEG16	LCDDATA2, 0		LCDDATA5, 0		LCDDATA8, 0		LCDDATA11, 0		-/26	RD3	
SEG17	LCDDATA2, 1		LCDDATA5, 1		LCDDATA8, 1		LCDDATA11, 1		-/27	RD4	
SEG18	LCDDATA2, 2		LCDDATA5, 2		LCDDATA8, 2		LCDDATA11, 2		-/28	RD5	
SEG19	LCDDATA2, 3		LCDDATA5, 3		LCDDATA8, 3		LCDDATA11, 3		-/29	RD6	
SEG20	LCDDATA2, 4		LCDDATA5, 4		LCDDATA8, 4		LCDDATA11, 4		-/30	RD7	
SEG21	LCDDATA2, 5		LCDDATA5, 5		LCDDATA8, 5		LCDDATA11, 5		-/8	RE0	AN5
SEG22	LCDDATA2, 6		LCDDATA5, 6		LCDDATA8, 6		LCDDATA11, 6		-/9	RE1	AN6
SEG23	LCDDATA2, 7		LCDDATA5, 7		LCDDATA8, 7		LCDDATA11, 7		-/10	RE2	AN7

▨ PIC16F914/917 only.

9.7 LCD Waveform Generation

LCD waveforms are generated so that the net AC voltage across the dark pixel should be maximized and the net AC voltage across the clear pixel should be minimized. The net DC voltage across any pixel should be zero.

The COM signal represents the time slice for each common, while the SEG contains the pixel data.

The pixel signal (COM-SEG) will have no DC component and it can take only one of the two rms values. The higher rms value will create a dark pixel and a lower rms value will create a clear pixel.

As the number of commons increases, the delta between the two rms values decreases. The delta represents the maximum contrast that the display can have.

The LCDs can be driven by two types of waveform: Type-A and Type-B. In Type-A waveform, the phase changes within each common type, whereas in Type-B waveform, the phase changes on each frame boundary. Thus, Type-A waveform maintains 0 VDC over a single frame, whereas Type-B waveform takes two frames.

> **Note 1:** If Sleep has to be executed with LCD Sleep enabled (LCDCON<SLPEN> is '1'), then care must be taken to execute Sleep only when VDC on all the pixels is '0'.
>
> **2:** When the LCD clock source is FOSC/8192, if Sleep is executed, irrespective of the LCDCON<SLPEN> setting, the LCD goes into Sleep. Thus, take care to see that VDC on all pixels is '0' when Sleep is executed.

Figure 9-5 through Figure 9-15 provide waveforms for static, half-multiplex, one-third-multiplex and quarter-multiplex drives for Type-A and Type-B waveforms.

FIGURE 9-5: TYPE-A/TYPE-B WAVEFORMS IN STATIC DRIVE

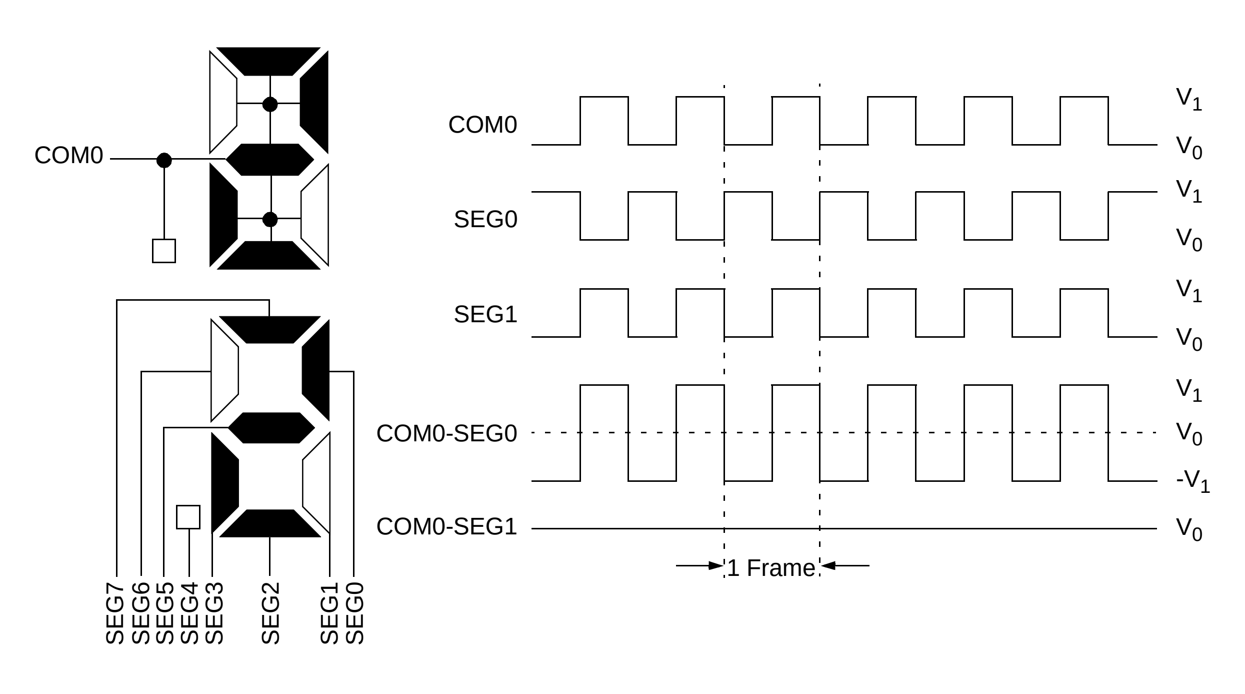


FIGURE 9-6: TYPE-A WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE

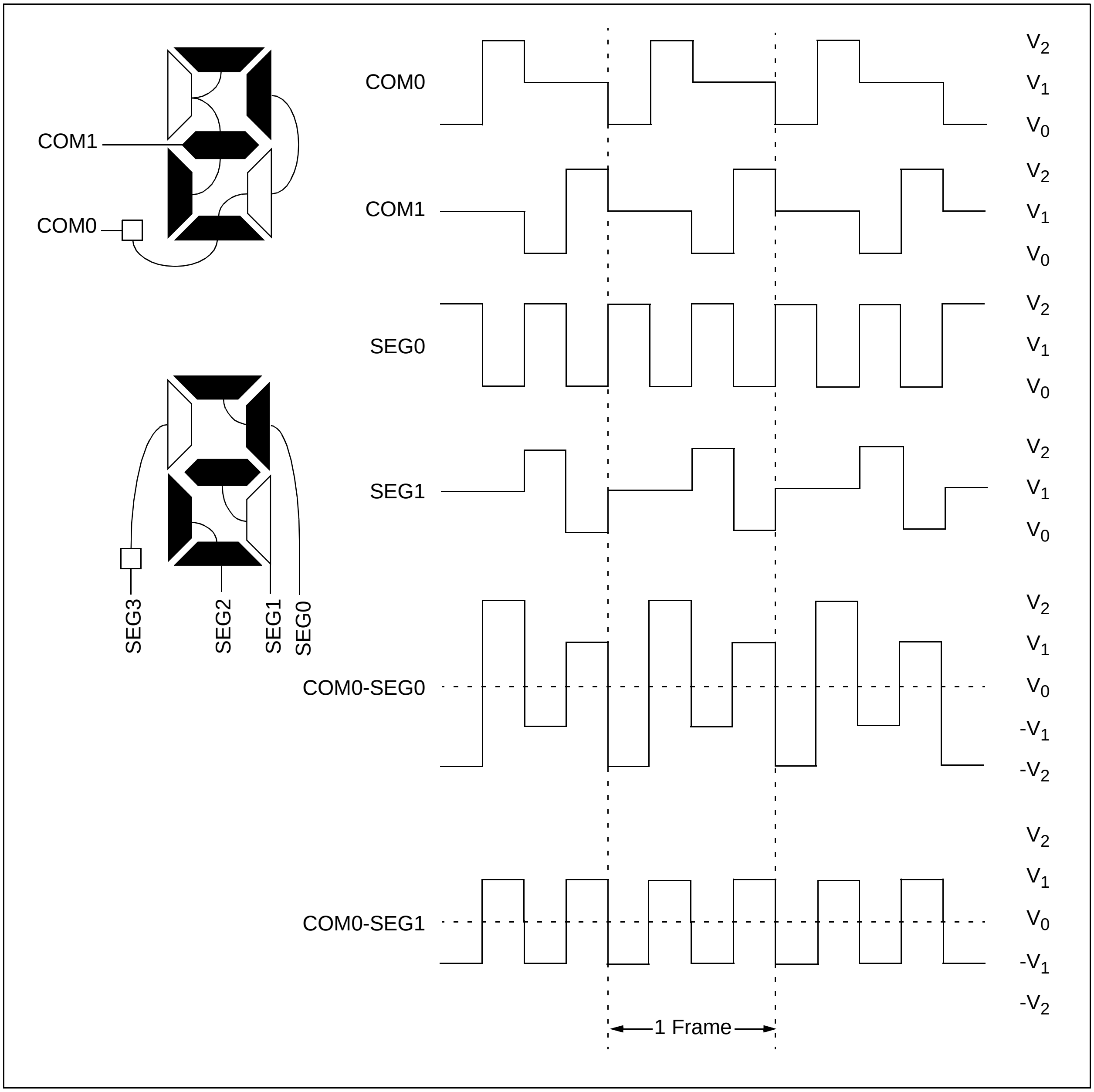


FIGURE 9-7: TYPE-B WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE

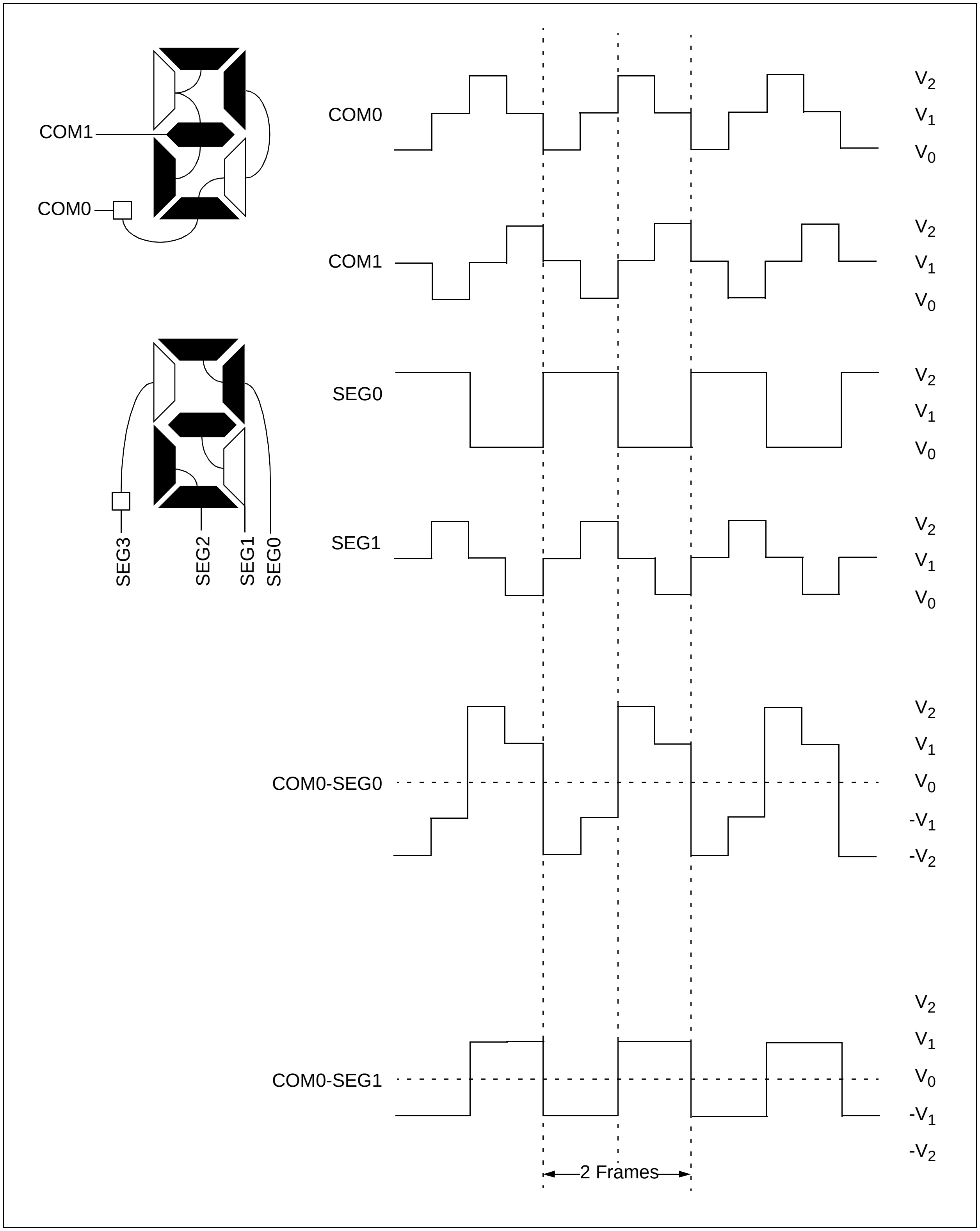


FIGURE 9-8: TYPE-A WAVEFORMS IN 1/2 MUX, 1/3 BIAS DRIVE

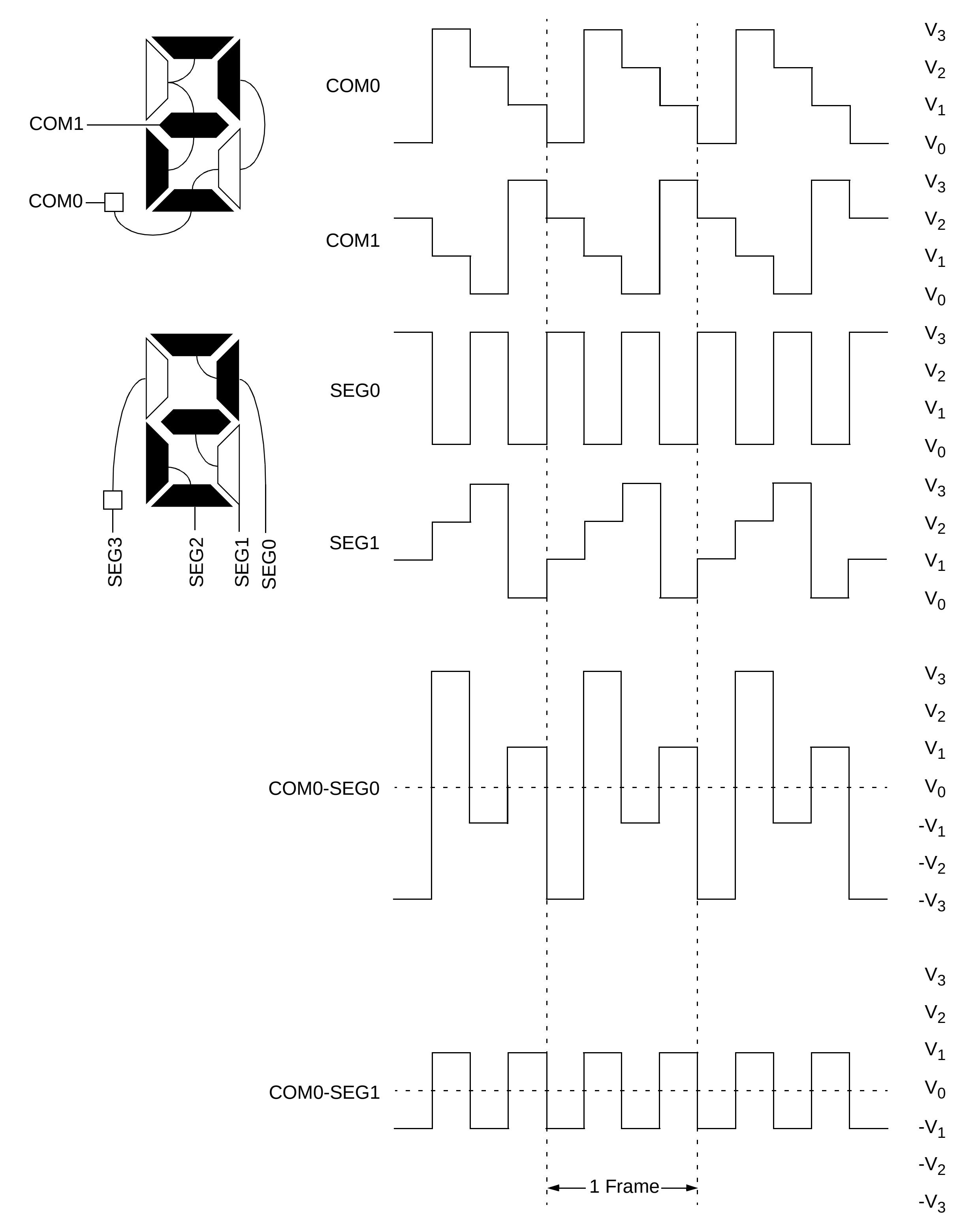


FIGURE 9-9: TYPE-B WAVEFORMS IN 1/2 MUX, 1/3 BIAS DRIVE

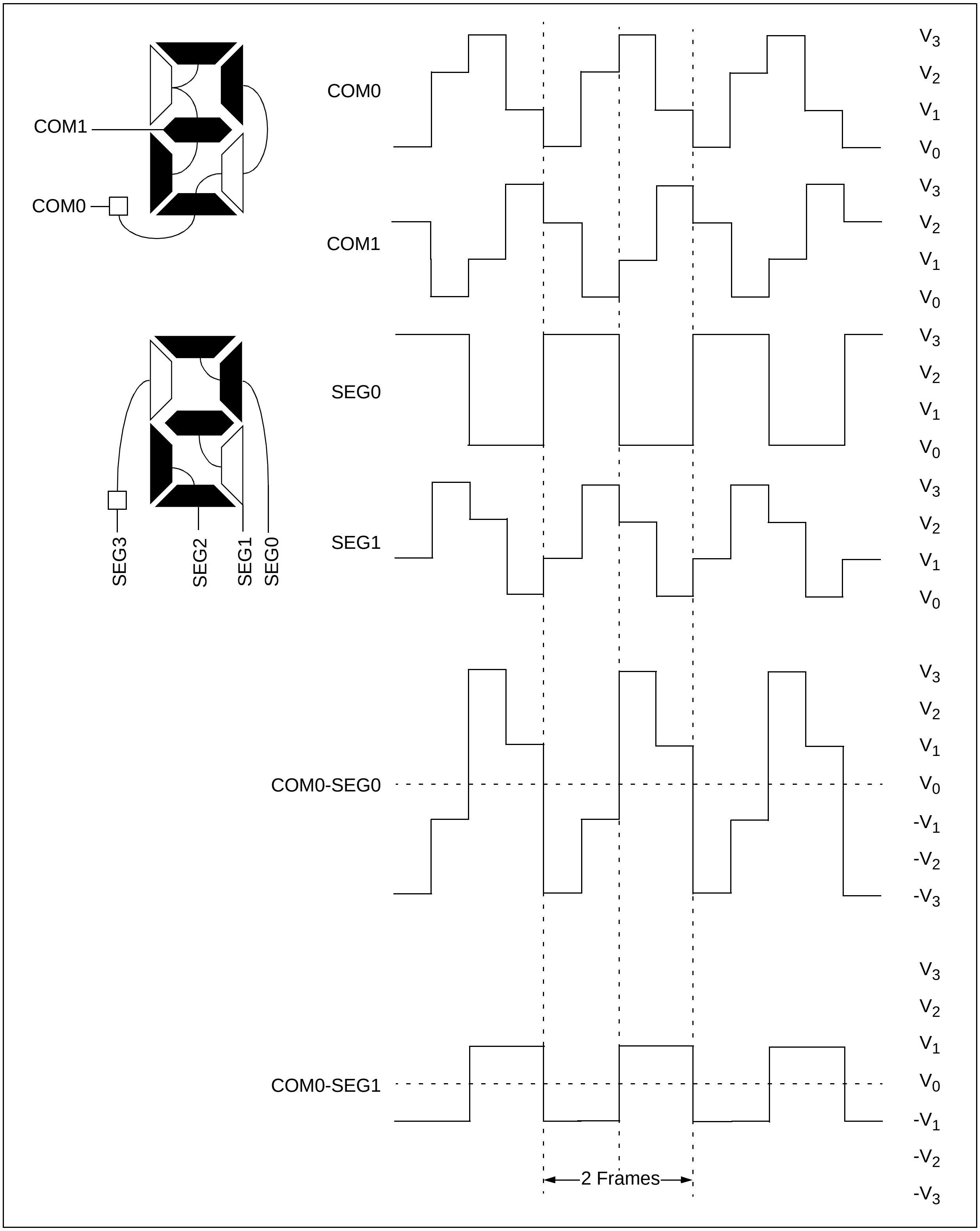


FIGURE 9-10: TYPE-A WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE

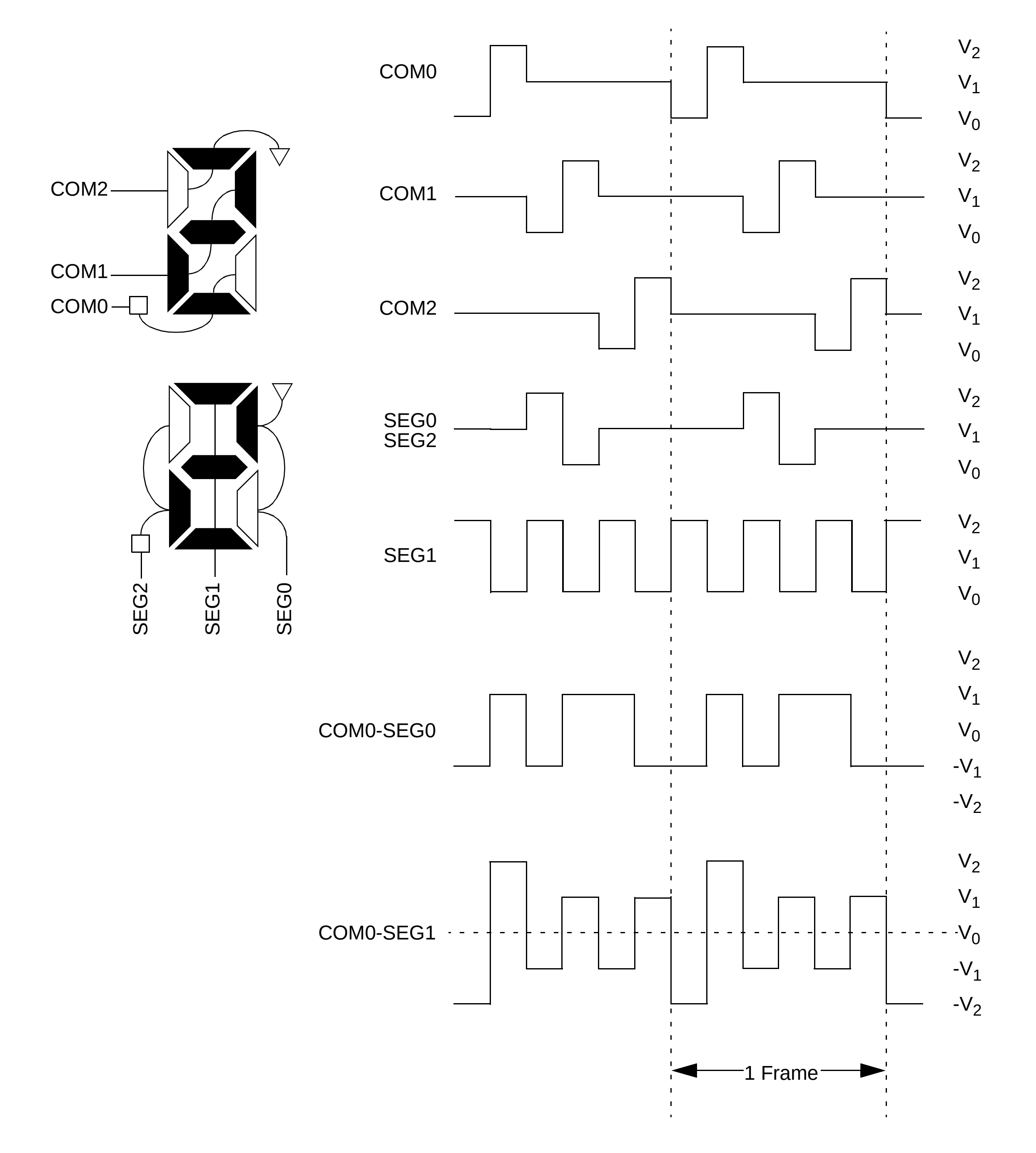


FIGURE 9-11: TYPE-B WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE

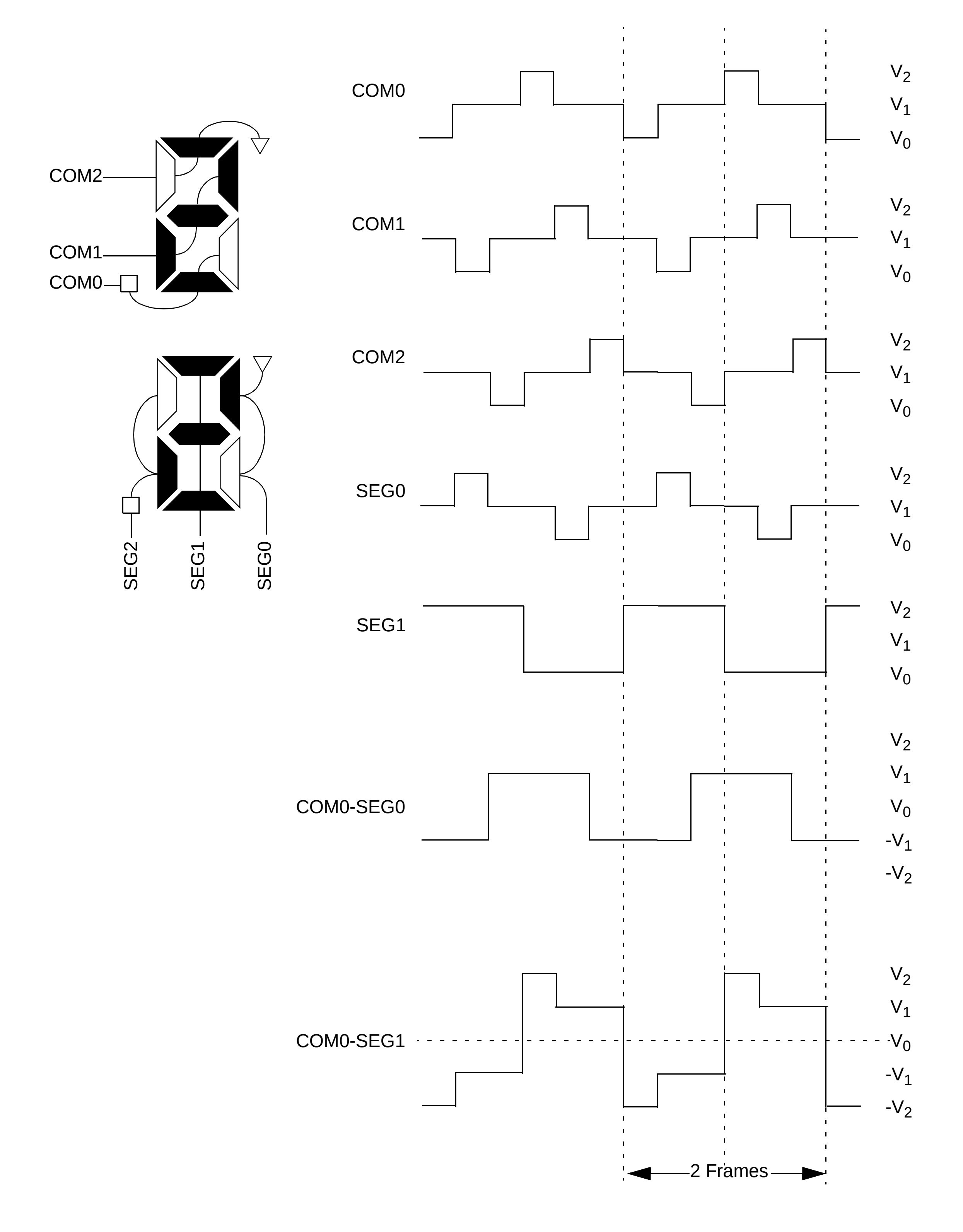


FIGURE 9-12: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE

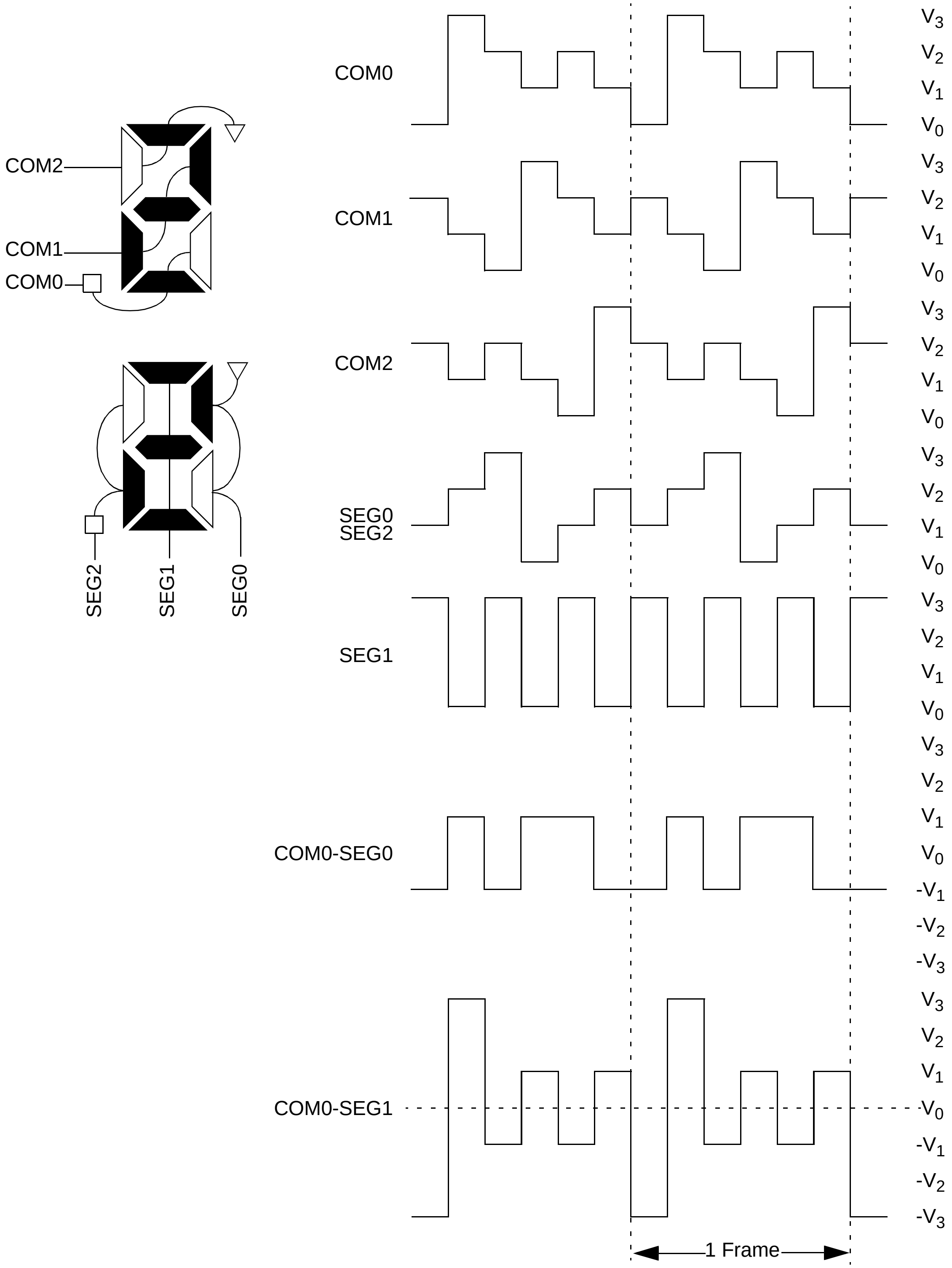


FIGURE 9-13: TYPE-B WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE

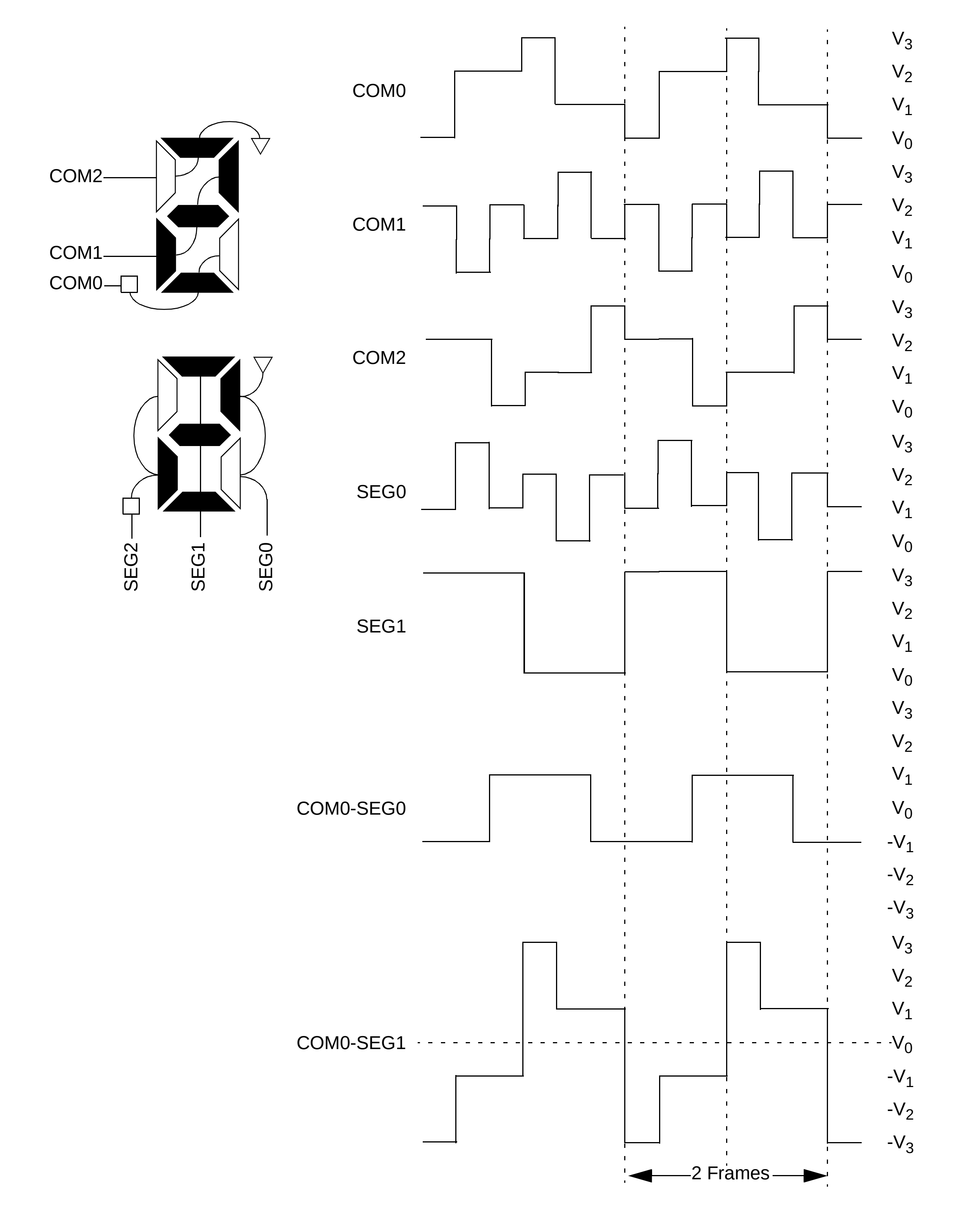


FIGURE 9-14: TYPE-A WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE

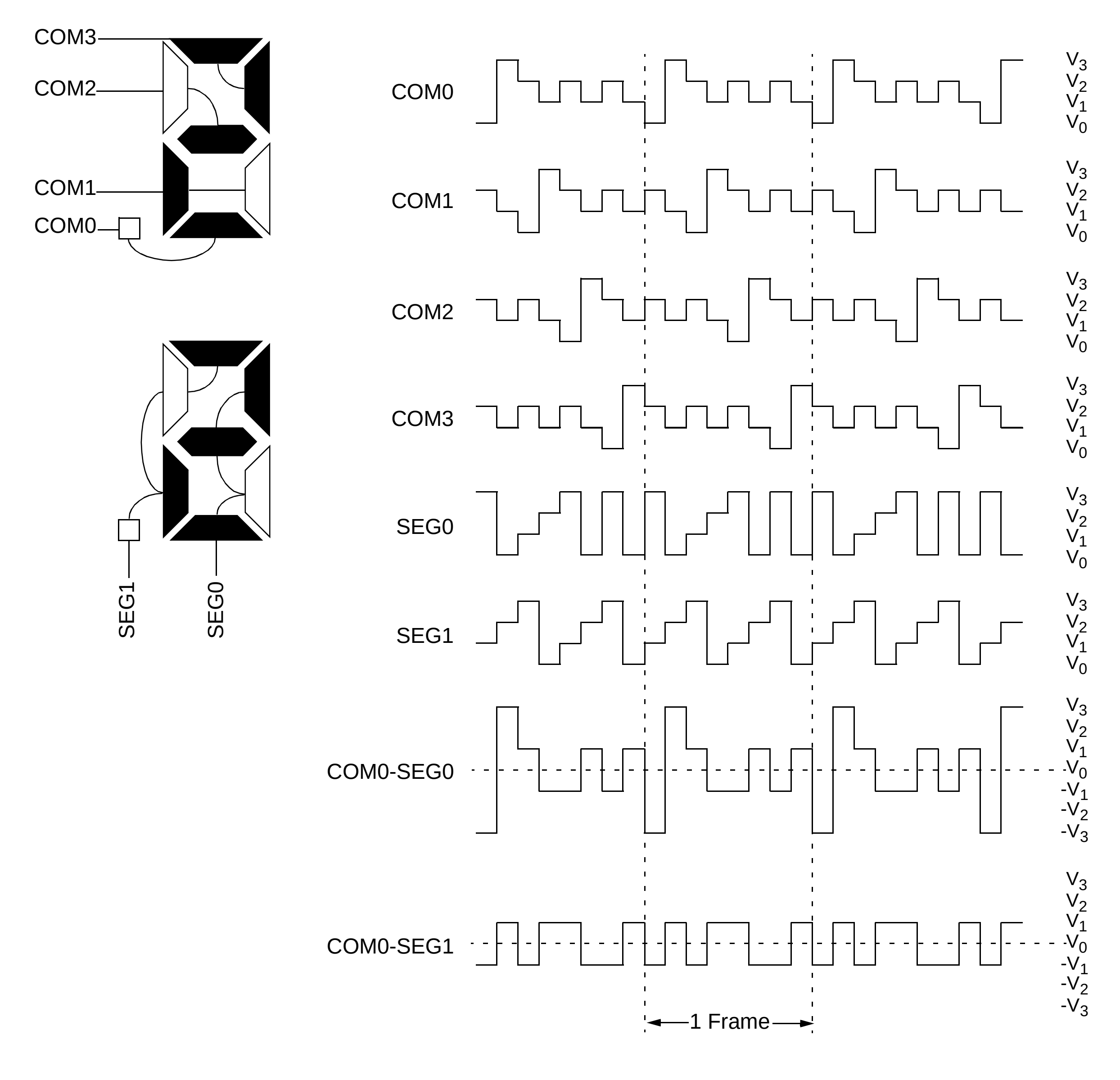
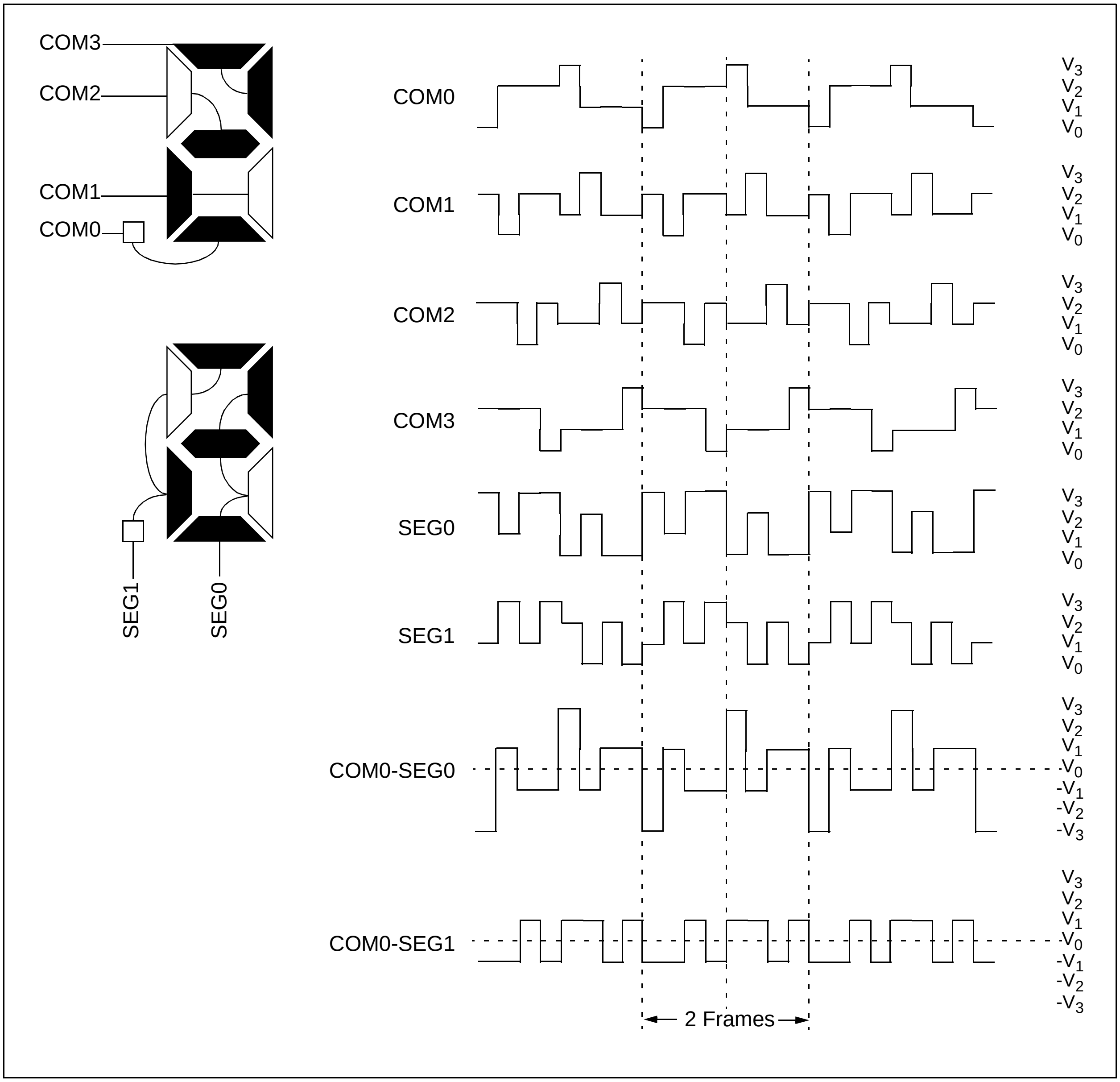


FIGURE 9-15: TYPE-B WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE



9.8 LCD Interrupts

The LCD timing generation provides an interrupt that defines the LCD frame timing. This interrupt can be used to coordinate the writing of the pixel data with the start of a new frame. Writing pixel data at the frame boundary allows a visually crisp transition of the image. This interrupt can also be used to synchronize external events to the LCD.

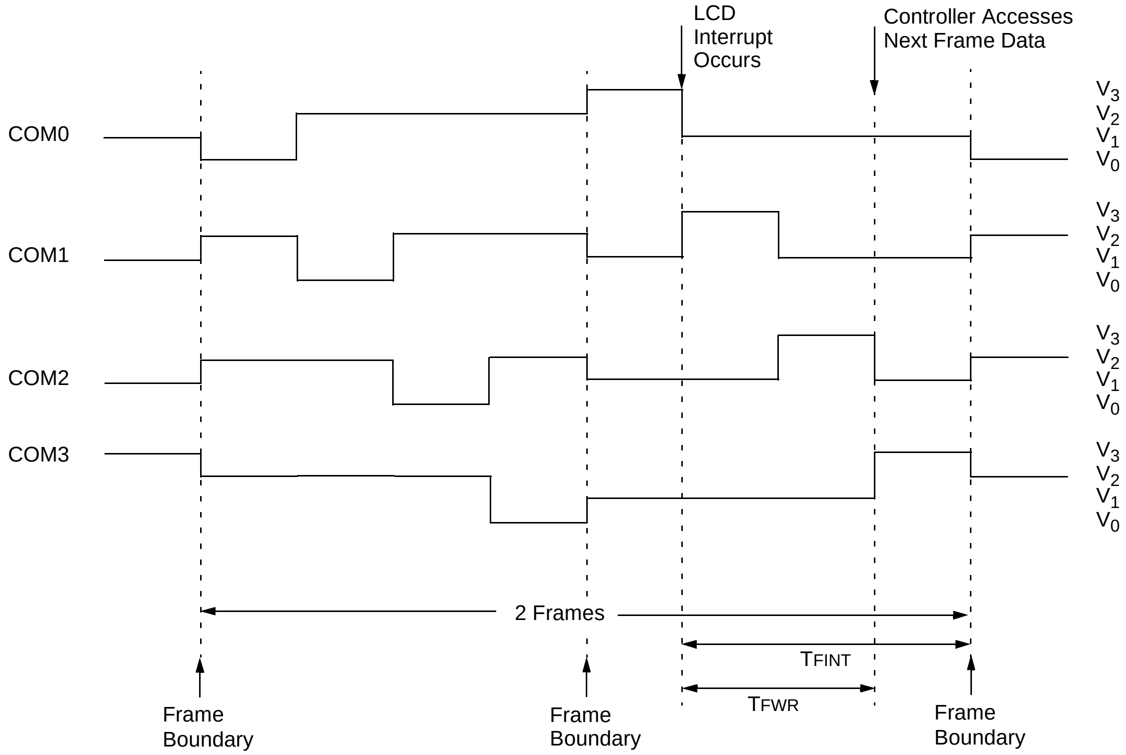
A new frame is defined to begin at the leading edge of the COM0 common signal. The interrupt will be set immediately after the LCD controller completes accessing all pixel data required for a frame. This will occur at a fixed interval before the frame boundary (TFINT), as shown in Figure 9-16. The LCD controller will begin to access data for the next frame within the interval from the interrupt to when the controller begins to access data after the interrupt (TFWR). New data must be written within TFWR, as this is when the LCD controller will begin to access the data for the next frame.

When the LCD driver is running with Type-B waveforms and the LMUX<1:0> bits are not equal to '00', there are some additional issues that must be addressed. Since the DC voltage on the pixel takes two frames to maintain zero volts, the pixel data must not change between subsequent frames. If the pixel data were allowed to change, the waveform for the odd frames would not necessarily be the complement of the waveform generated in the even frames and a DC component would be introduced into the panel. Therefore, when using Type-B waveforms, the user must synchronize the LCD pixel updates to occur within a subframe after the frame interrupt.

To correctly sequence writing while in Type-B, the interrupt will only occur on complete phase intervals. If the user attempts to write when the write is disabled, the WERR (LCDCON<5>) bit is set.

Note:	The interrupt is not generated when the Type-A waveform is selected and when the Type-B with no multiplex (static) is selected.

FIGURE 9-16: WAVEFORMS AND INTERRUPT TIMING IN QUARTER-DUTY CYCLE DRIVE (EXAMPLE – TYPE-B, NON-STATIC)



TFWR = TFRAME/2*(LMUX<1:0> + 1) + TCY/2

TFINT = (TFWR/2 – (2 TCY + 40 ns)) → minimum = 1.5(TFRAME/4) – (2 TCY + 40 ns)
(TFWR/2 – (1 TCY + 40 ns)) → maximum = 1.5(TFRAME/4) – (1 TCY + 40 ns)

9.9 Operation During Sleep

The LCD module can operate during Sleep. The selection is controlled by bit SLPEN (LCDCON<6>). Setting the SLPEN bit allows the LCD module to go to Sleep. Clearing the SLPEN bit allows the module to continue to operate during Sleep.

If a SLEEP instruction is executed and SLPEN = 1, the LCD module will cease all functions and go into a very low current consumption mode. The module will stop operation immediately and drive the minimum LCD voltage on both segment and common lines. Figure 9-17 shows this operation.

To ensure that no DC component is introduced on the panel, the SLEEP instruction should be executed immediately after a LCD frame boundary. The LCD interrupt can be used to determine the frame boundary. See **Section 9.8 "LCD Interrupts"** for the formulas to calculate the delay.

If a SLEEP instruction is executed and SLPEN = 0, the module will continue to display the current contents of the LCDDATA registers. To allow the module to continue operation while in Sleep, the clock source must be either the LFINTOSC or T1OSC external oscillator. While in Sleep, the LCD data cannot be changed. The LCD module current consumption will not decrease in this mode; however, the overall consumption of the device will be lower due to shut down of the core and other peripheral functions.

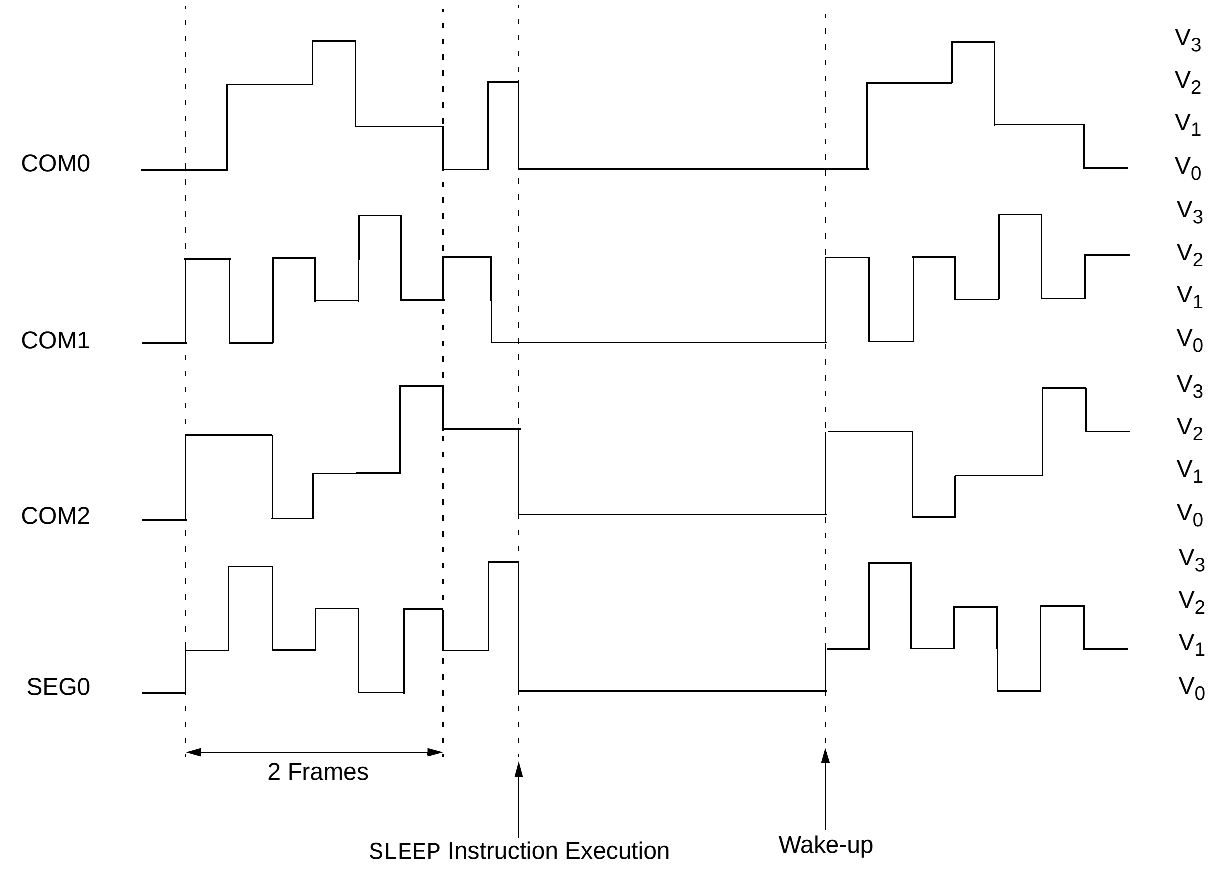
Table 9-4 shows the status of the LCD module during a Sleep while using each of the three available clock sources:

TABLE 9-4: LCD MODULE STATUS DURING SLEEP

Clock Source	SLPEN	Operation During Sleep?
T1OSC	0	Yes
	1	No
LFINTOSC	0	Yes
	1	No
FOSC/4	0	No
	1	No

Note:	The LFINTOSC or external T1OSC oscillator must be used to operate the LCD module during Sleep.

FIGURE 9-17: SLEEP ENTRY/EXIT WHEN SLPEN = 1 OR CS<1:0> = 00



9.10 Configuring the LCD Module

The following is the sequence of steps to configure the LCD module.

1. Select the frame clock prescale using bits LP<3:0> (LCDPS<3:0>).
2. Configure the appropriate pins to function as segment drivers using the LCDSEn registers.
3. Configure the LCD module for the following using the LCDCON register:

   -Multiplex and Bias mode, bits LMUX<1:0>

   -Timing source, bits CS<1:0>

   -Sleep mode, bit SLPEN
4. Write initial values to pixel data registers, LCDDATA0 through LCDDATA11.
5. Clear LCD Interrupt Flag, LCDIF (PIR2<4>) and if desired, enable the interrupt by setting bit LCDIE (PIE2<4>).
6. Enable bias voltage pins (VLCD<3:1>) by setting VLCDEN (LCDCON<4>).
7. Enable the LCD module by setting bit LCDEN (LCDCON<7>).

TABLE 9-5: REGISTERS ASSOCIATED WITH LCD OPERATION

Address	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
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
107h	LCDCON	LCDEN	SLPEN	WERR	VLCDEN	CS1	CS0	LMUX1	LMUX0	0001 0011	0001 0011
108h	LCDPS	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0	0000 0000	0000 0000
110h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	xxxx xxxx	uuuu uuuu
111h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	xxxx xxxx	uuuu uuuu
112h	LCDDATA2(2)	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	xxxx xxxx	uuuu uuuu
113h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	xxxx xxxx	uuuu uuuu
114h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	xxxx xxxx	uuuu uuuu
115h	LCDDATA5(2)	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	xxxx xxxx	uuuu uuuu
116h	LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	xxxx xxxx	uuuu uuuu
117h	LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	xxxx xxxx	uuuu uuuu
118h	LCDDATA8(2)	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	xxxx xxxx	uuuu uuuu
119h	LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	xxxx xxxx	uuuu uuuu
11Ah	LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	xxxx xxxx	uuuu uuuu
11Bh	LCDDATA11(2)	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	xxxx xxxx	uuuu uuuu
11Ch	LCDSE0(3)	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu
11Dh	LCDSE1(3)	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu
11Eh	LCDSE2(2,3)	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	0000 0000	uuuu uuuu

Legend: x = unknown, u = unchanged, — = unimplemented, read as ‘0’. Shaded cells are not used by the LCD module.

Note 1: These pins may be configured as port pins, depending on the oscillator mode selected.

2: PIC16F914/917 only.

3: This register is only initialized by a POR or BOR reset and is unchanged by other resets.

10.0 PROGRAMMABLE LOW-VOLTAGE DETECT (PLVD) MODULE

The Programmable Low-Voltage Detect module is an interrupt driven supply level detection. The voltage detection monitors the internal power supply.

10.1 Voltage Trip Points

The PIC16F91X device supports eight internal PLVD trip points. See Register 10-1 for available PLVD trip point voltages.

10.1.1 PLVD CALIBRATION

The PIC16F91X stores the PLVD calibration values in fuses located in the Calibration Word 2 (2009h). The Calibration Word 2 is not erased when using the specified bulk erase sequence in the "PIC16F91X *Memory Programming Specification"* (DS41244) and thus, does not require reprogramming.

REGISTER 10-1: LVDCON – LOW-VOLTAGE DETECT CONTROL REGISTER (ADDRESS: 109h)

U-0	U-0	R-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	—	IRVST	LVDEN	—	LVDL2	LVDL1	LVDL0
bit 7							bit 0

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

bit 5 **IRVST:** Internal Reference Voltage Stable Status Flag bit
1 = Indicates that the PLVD is stable and PLVD interrupt is reliable
0 = Indicates that the PLVD is not stable and PLVD interrupt should not be enabled

bit 4 **LVDEN:** Low-Voltage Detect Power Enable bit
1 = Enables PLVD, powers up PLVD circuit and supporting reference circuitry
0 = Disables PLVD, powers down PLVD and supporting circuitry

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

bit 2-0 **LVDL<2:0>:** Low-Voltage Detection Limit bits (nominal values)
111 = 4.5V
110 = 4.2V
101 = 4.0V
100 = 2.3V (default)
011 = 2.2V
010 = 2.1V
001 = 2.0V
000 = 1.9V(1)

Note 1: Not tested and below minimum VDD.

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

TABLE 10-1: REGISTERS ASSOCIATED WITH PROGRAMMABLE LOW-VOLTAGE DETECT

Address	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
0Bh/8Bh/10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
109h	LVDCON	—	—	IRVST	LVDEN	—	LVDL2	LVDL1	LVDL0	--00 -100	--00 -100

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used by the PLVD module.

NOTES:

11.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

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

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK/SCK/SCL/SEG9 and RC7/RX/DT/SDI/SDA/SEG8 as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 11-1: TXSTA – TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don’t care

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit

1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC**: USART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **BRGH**: High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

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

REGISTER 11-2: RCSTA – RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

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

bit 7 **SPEN[(1)]:** Serial Port Enable bit

1 = Serial port enabled (configures RC7/RX/DT/SDI/SDA/SEG8 and RC6/TX/CK/SCK/SCL/SEG9 pins as serial port pins)
0 = Serial port disabled

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 continuous receive
0 = Disables continuous receive

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 load of 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 receive 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 (can be parity bit but must be calculated by user firmware)

Note 1: CCP2CON used for PIC16F914/917 only.

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

11.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 11-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRG register can be calculated using the formula in Table 11-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the FOSC/(16 (X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

11.1.1 SAMPLING

The data on the RC7/RX/DT/SDI/SDA/SEG8 pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 11-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = FOSC/(64 (X + 1))	Baud Rate = FOSC/(16 (X + 1))
1	(Synchronous) Baud Rate = FOSC/(4 (X + 1))	N/A

Legend: X = value in SPBRG (0 to 255)

TABLE 11-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	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
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

TABLE 11-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	FOSC = 20 MHz			FOSC = 16 MHz			FOSC = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	FOSC = 4 MHz			FOSC = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

TABLE 11-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	FOSC = 20 MHz			FOSC = 16 MHz			FOSC = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000		0	625.000	-	0

BAUD RATE (K)	FOSC = 4 MHz			FOSC = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

11.2 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 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 is stopped during Sleep.

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

11.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 11-1. 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. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates 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 so the user has to poll this bit in order to determine if 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. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit, TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the Baud Rate Generator (BRG) has produced a shift clock (Figure 11-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 11-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/TX/CK/SCK/SCL/SEG9 pin will revert to high-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (**Section 11.1 "USART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 11-1: USART TRANSMIT BLOCK DIAGRAM

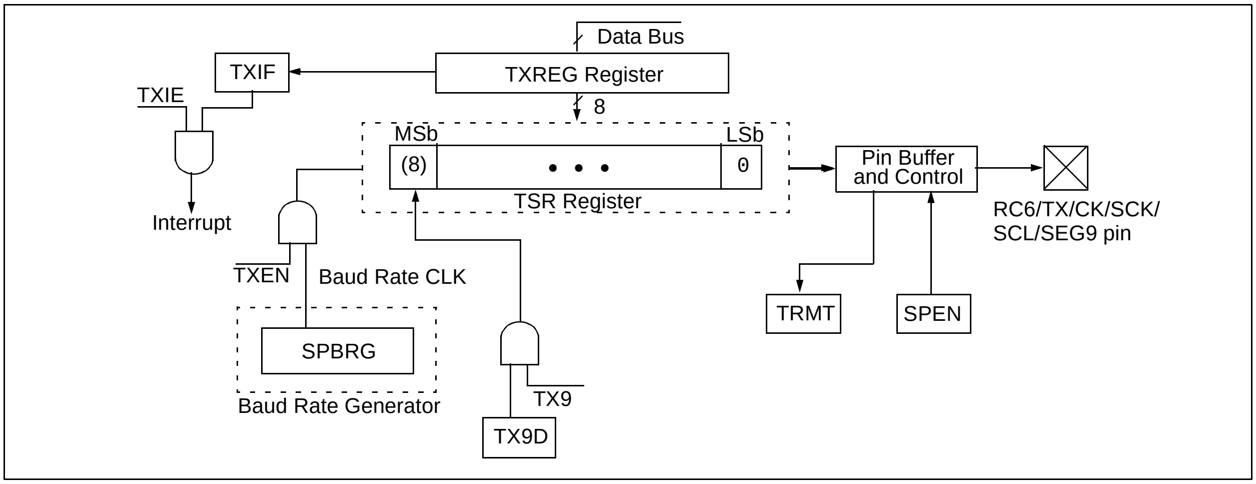


FIGURE 11-2: ASYNCHRONOUS MASTER TRANSMISSION

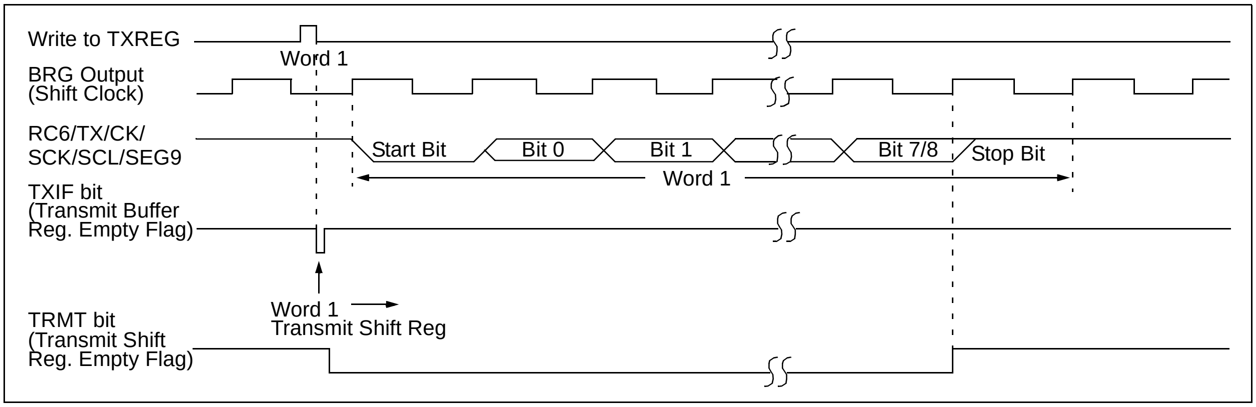
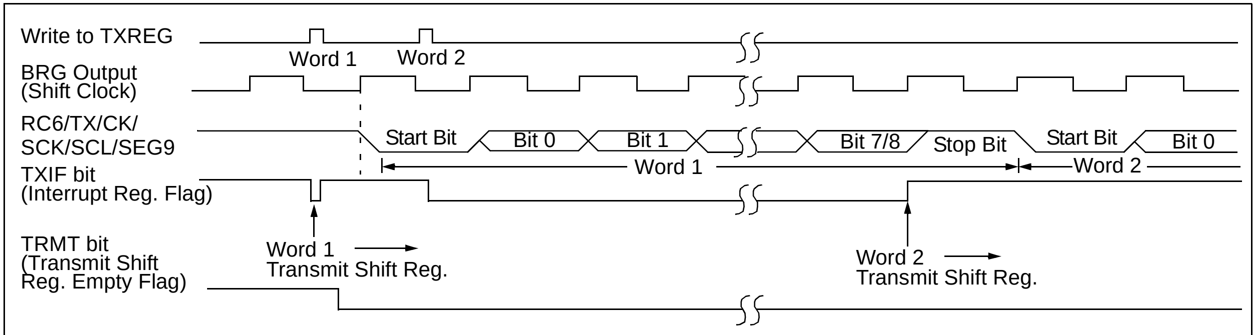

Write to TXREG
Word 1
BRG Output (Shift Clock)
RC6/TX/CK/ SCK/SCL/SEG9
Start Bit
Bit 0
Bit 1
Bit 7/8
Stop Bit
Word 1
TXIF bit (Transmit Buffer Reg. Empty Flag)
Word 1 → Transmit Shift Reg
TRMT bit (Transmit Shift Reg. Empty Flag)

FIGURE 11-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 11-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as ‘0’. Shaded cells are not used for asynchronous transmission.

11.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 11-4. The data is received on the RC7/RX/DT/SDI/SDA/SEG8 pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate; whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the Receive (Serial) Shift Register (RSR). After sampling the Stop bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit, RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the Stop bit of the third byte, if the RCREG register is still full, the Overrun Error bit, OERR (RCSTA<1>), will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited and no further data will be received. It is, therefore, essential to clear error bit OERR if it is set. Framing error bit, FERR (RCSTA<2>), is set if a Stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore, it is essential for the user to read the RCSTA register before reading the RCREG register in order not to lose the old FERR and RX9D information.

When setting up an Asynchronous Reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (**Section 11.1 "USART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 11-4: USART RECEIVE BLOCK DIAGRAM

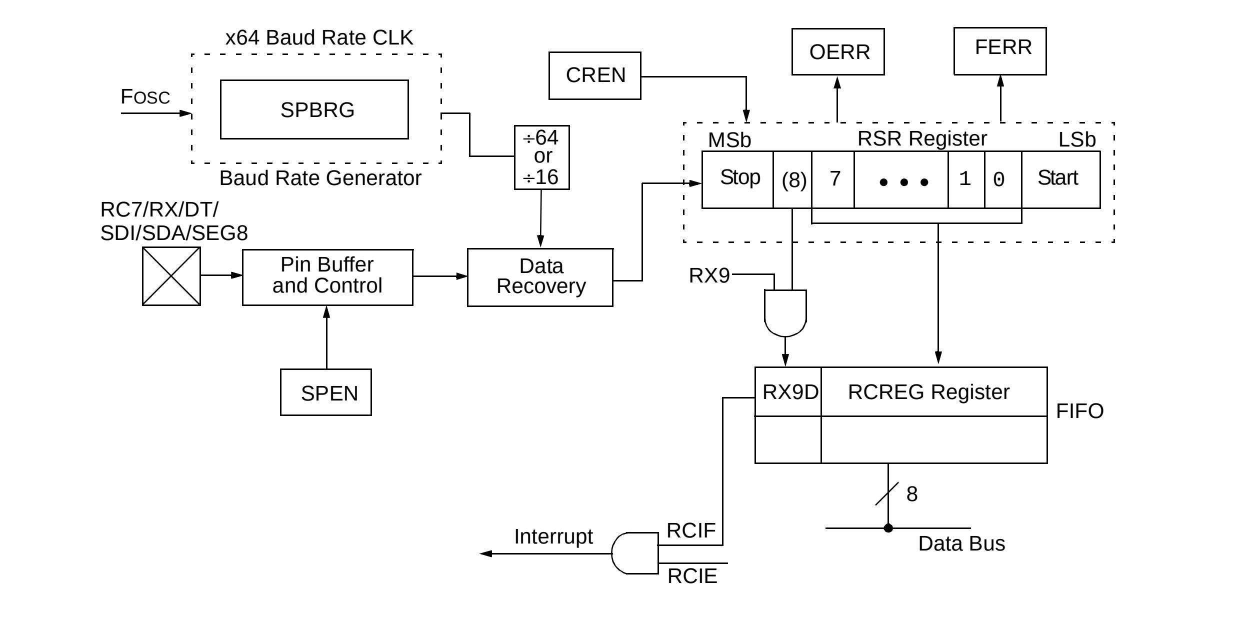
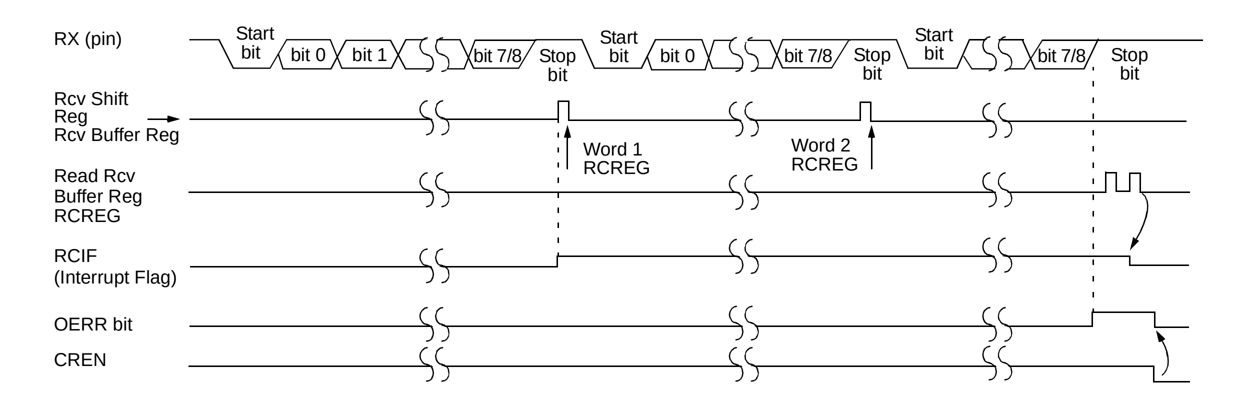


FIGURE 11-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (Receive Buffer) is read after the third word, causing the OERR (Overrun Error) bit to be set.

TABLE 11-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000x`
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 000x`	`0000 000x`
1Ah	RCREG	USART Receive Data Register								`0000 0000`	`0000 0000`
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	`0000 0000`	`0000 0000`
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	`0000 -010`	`0000 -010`
99h	SPBRG	Baud Rate Generator Register								`0000 0000`	`0000 0000`

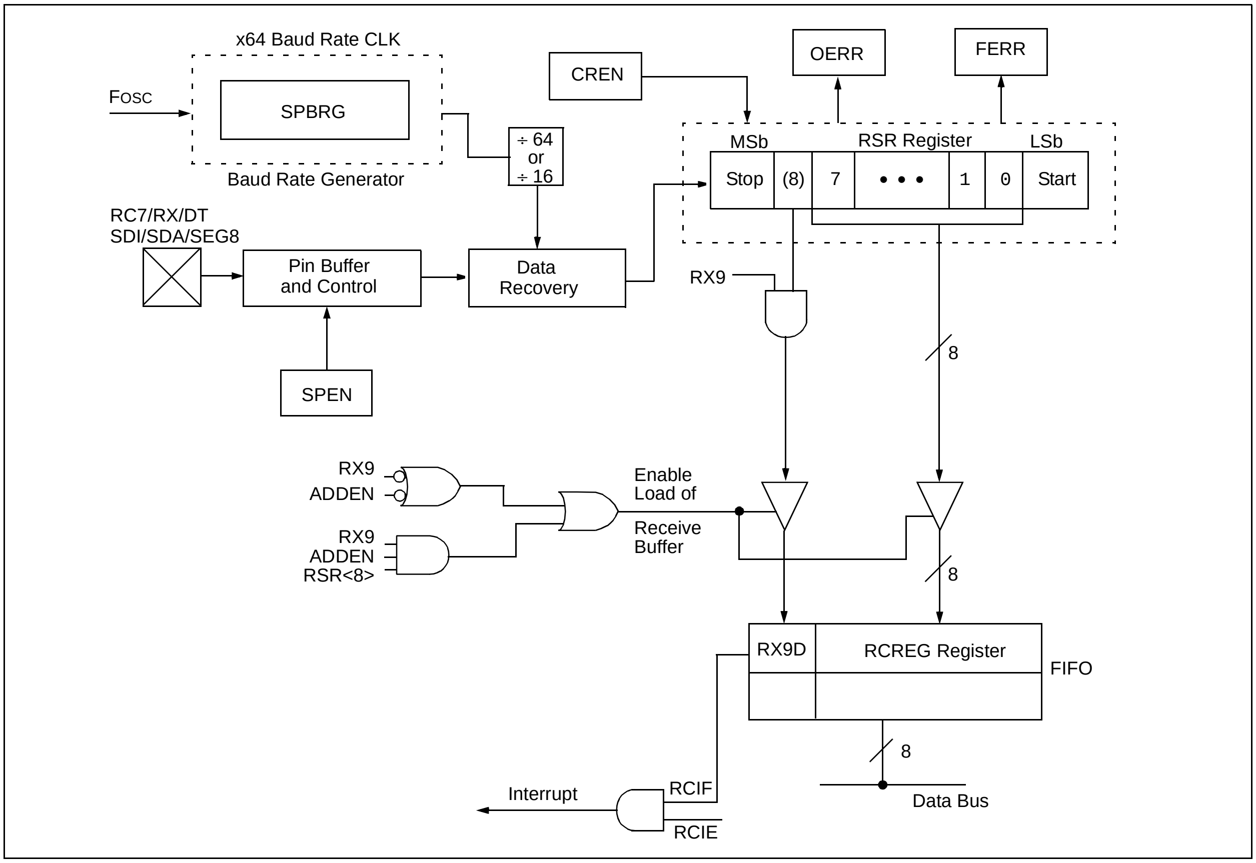
Legend: x = unknown, - = unimplemented locations read as ‘0’. Shaded cells are not used for asynchronous reception.

11.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

When setting up an Asynchronous Reception with address detect enabled:

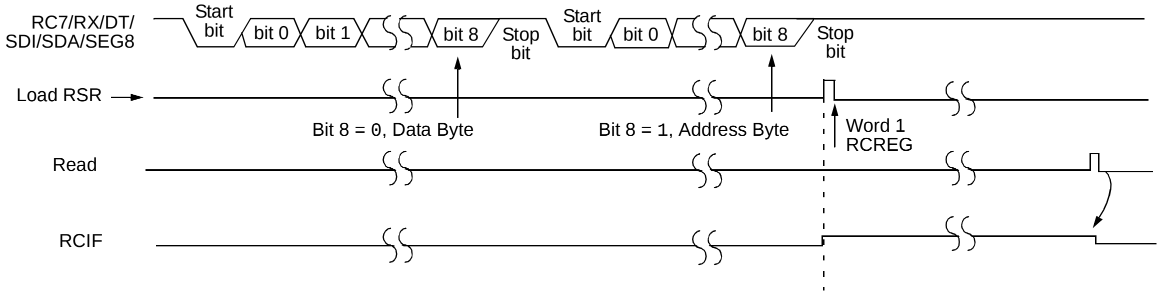
- Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- Set bit RX9 to enable 9-bit reception.
- Set ADDEN to enable address detect.
- Enable the reception by setting enable bit CREN.
- Flag bit RCIF will be set when reception is complete, and an interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register to determine if the device is being addressed.
- If any error occurred, clear the error by clearing enable bit CREN.
- If the device has been addressed, clear the ADDEN bit to allow data bytes and address bytes to be read into the receive buffer and interrupt the CPU.

FIGURE 11-6: USART RECEIVE BLOCK DIAGRAM



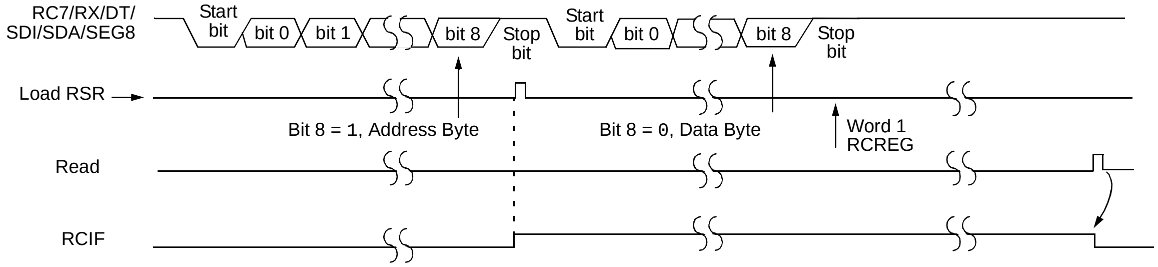
Preliminary

FIGURE 11-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (Receive Buffer) because ADDEN = 1.

FIGURE 11-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (Receive Buffer) because ADDEN was not updated and still = 0.

TABLE 11-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as ‘0’. Shaded cells are not used for asynchronous reception.

11.3 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the RC6/TX/CK/SCK/SCL/SEG9 and RC7/RX/DT/SDI/SDA/SEG8 I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit, CSRC (TXSTA<7>).

11.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 11-6. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit, TXIF (PIR1<4>), is set. The 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. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting enable bit, TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 11-9). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 11-10). This is advantageous when slow baud rates are selected, since the BRG is kept in Reset when bits TXEN, CREN and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally, when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to high-impedance. If either bit CREN or bit SREN is set during a transmission, the transmission is aborted and the DT pin reverts to a high-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting, since bit TXEN is still set. The DT line will immediately switch from High-Impedance Receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit should be written to bit TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the “new” TX9D, the “present” value of bit TX9D is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

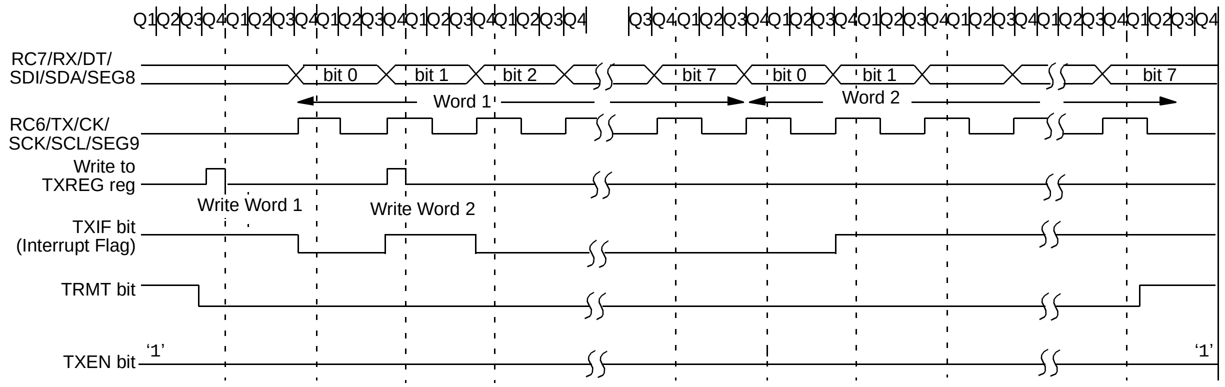
1. Initialize the SPBRG register for the appropriate baud rate (**Section 11.1 “USART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting 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 GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 11-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

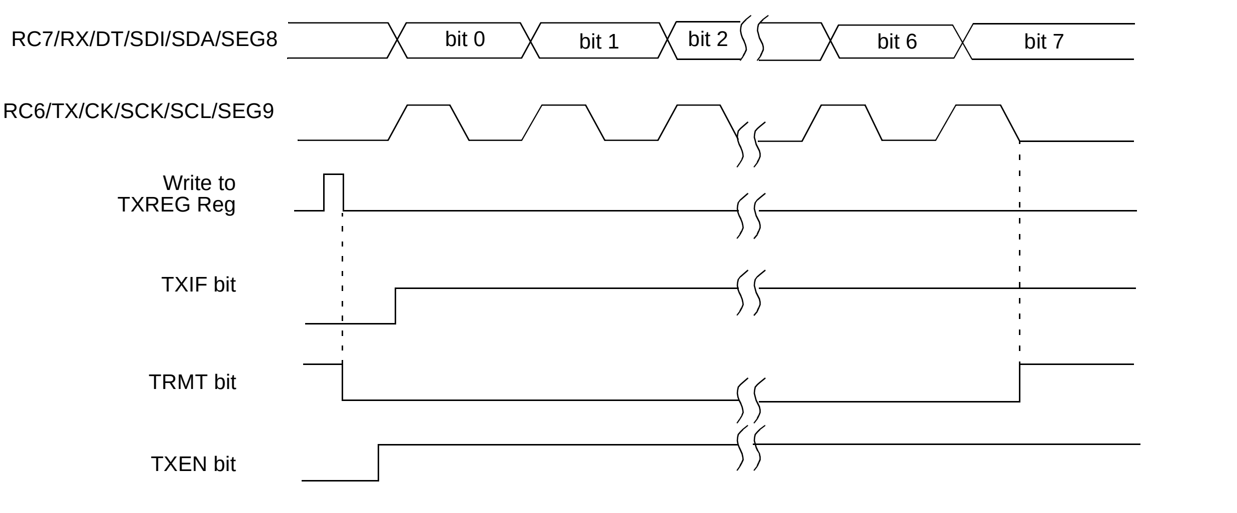
Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

FIGURE 11-9: SYNCHRONOUS TRANSMISSION


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
RC7/RX/DT/SDI/SDA/SEG8
bit 0
bit 1
bit 2
bit 7
bit 0
bit 1
bit 7
Word 1
Word 2
RC6/TX/CK/SCK/SCL/SEG9
Write to TXREG reg
Write Word 1
Write Word 2
TXIF bit (Interrupt Flag)
TRMT bit
TXEN bit
'1'
'1'

Note: Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.

FIGURE 11-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



11.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit, SREN (RCSTA<5>), or enable bit, CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT/SDI/SDA/SEG8 pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit, RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then Overrun Error bit, OERR (RCSTA<1>), is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited so it is essential to clear bit OERR if it is set. The ninth receive bit is buffered the same way as the receive data. Reading the RCREG register will load bit RX9D with a new value, therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

When setting up a Synchronous Master Reception:

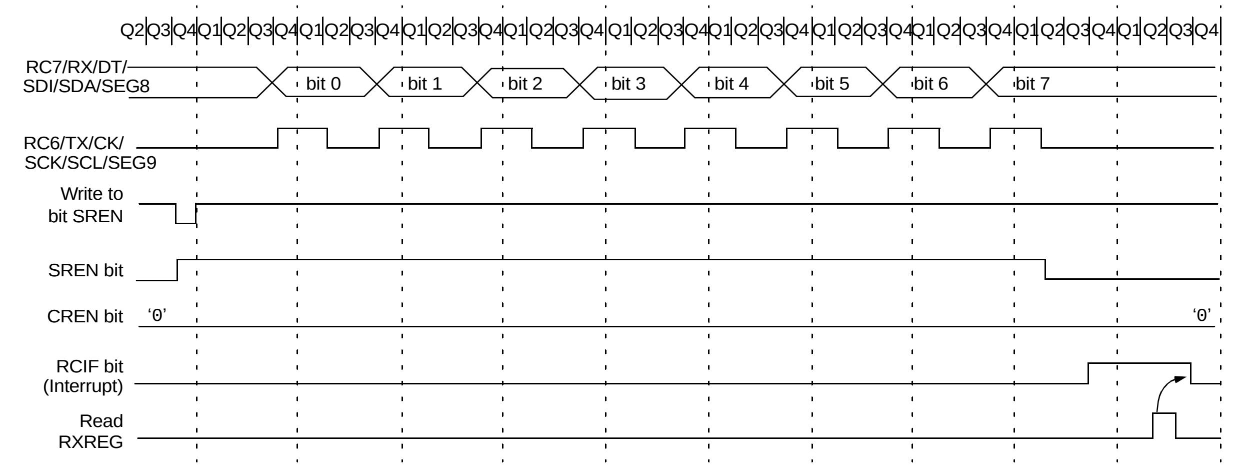
1. Initialize the SPBRG register for the appropriate baud rate (**Section 11.1 “USART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 11-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as ‘0’. Shaded cells are not used for synchronous master reception.

FIGURE 11-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRG = 0.

11.4 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/SCK/SCL/SEG9 pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit, CSRC (TXSTA<7>).

11.4.1 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 and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Transmission, follow these steps:

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, then set enable bit TXIE.
4. If 9-bit transmission is desired, then 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 GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 11-10: 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	Value on all other Resets
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	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.

11.4.2 USART SYNCHRONOUS SLAVE RECEPTION

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

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Reception, follow these steps:

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

TABLE 11-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

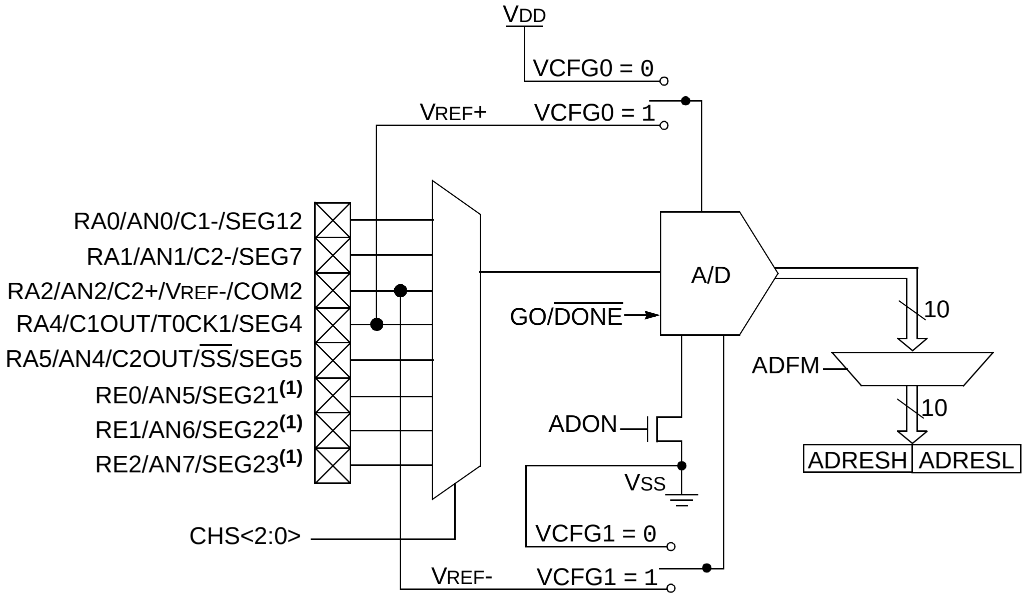
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

12.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC16F91X has up to eight analog inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the result in a 10-bit register. The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 12-1 shows the block diagram of the A/D on the PIC16F91X.

FIGURE 12-1: A/D BLOCK DIAGRAM



Note 1: These channels are only available on PIC16F914/917 devices.

12.1 A/D Configuration and Operation

There are three registers available to control the functionality of the A/D module:

1. ANSEL (Register 12-1)
2. ADCON0 (Register 12-2)
3. ADCON1 (Register 12-3)

12.1.1 ANALOG PORT PINS

The ANS<7:0> bits (ANSEL<7:0>) and the TRIS bits control the operation of the A/D port pins. Set the corresponding TRIS bits to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANSEL bit to disable the digital input buffer.

> **Note:** Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

12.1.2 CHANNEL SELECTION

There are up to eight analog channels on the PIC16F91X, AN<7:0>. The CHS<2:0> bits (ADCON0<4:2>) control which channel is connected to the sample and hold circuit.

12.1.3 VOLTAGE REFERENCE

There are two options for each reference to the A/D converter, VREF+ and VREF-. VREF+ can be connected to either VDD or an externally applied voltage. Alternatively, VREF- can be connected to either VSS or an externally applied voltage. VCFG<1:0> bits are used to select the reference source.

12.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ADCON1<6:4>). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 µs. Table 12-1 shows a few TAD calculations for selected frequencies.

TABLE 12-1: TAD VS. DEVICE OPERATING FREQUENCIES

A/D Clock Source (TAD)		Device Frequency			
Operation	**ADCS<2:0>**	**20 MHz**	**5 MHz**	**4 MHz**	**1.25 MHz**
2 TOSC	000	100 ns(2)	400 ns(2)	500 ns(2)	1.6 µs
4 TOSC	100	200 ns(2)	800 ns(2)	1.0 µs(2)	3.2 µs
8 TOSC	001	400 ns(2)	1.6 µs	2.0 µs	6.4 µs
16 TOSC	101	800 ns(2)	3.2 µs	4.0 µs	12.8 µs(3)
32 TOSC	010	1.6 µs	6.4 µs	8.0 µs(3)	25.6 µs(3)
64 TOSC	110	3.2 µs	12.8 µs(3)	16.0 µs(3)	51.2 µs(3)
A/D RC	x11	2-6 µs(1,4)	2-6 µs(1,4)	2-6 µs(1,4)	2-6 µs(1,4)

Legend: Shaded cells are outside of recommended range.

Note 1: The A/D RC source has a typical TAD time of 4 µs for VDD > 3.0V.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When the device frequency is greater than 1 MHz, the A/D RC clock source is only recommended if the conversion will be performed during Sleep.

12.1.5 STARTING A CONVERSION

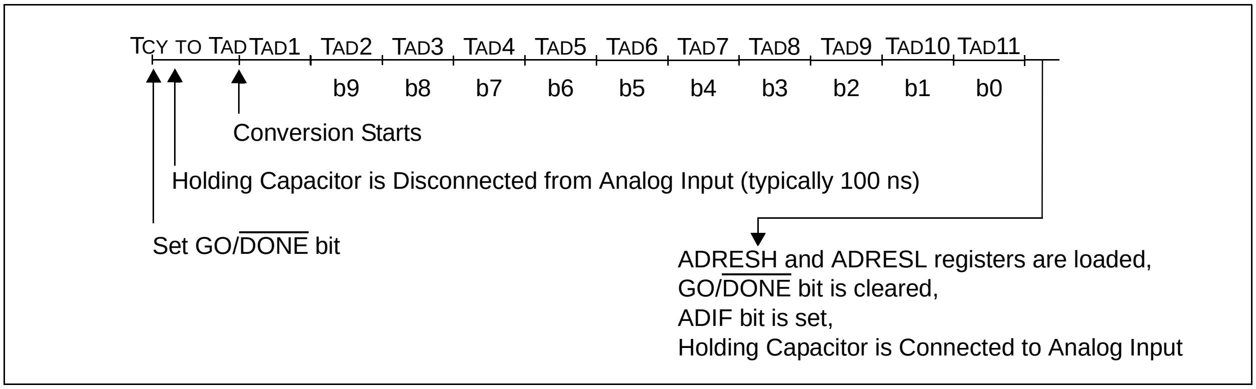
The A/D conversion is initiated by setting the GO/DONE bit (ADCON0<1>). When the conversion is complete, the A/D module:

- Clears the GO/DONE bit
- Sets the ADIF flag (PIR1<6>)
- Generates an interrupt (if enabled)

If the conversion must be aborted, the GO/DONE bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete A/D conversion sample. Instead, the ADRESH:ADRESL registers will retain the value of the previous conversion. After an aborted conversion, a 2 TAD delay is required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

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

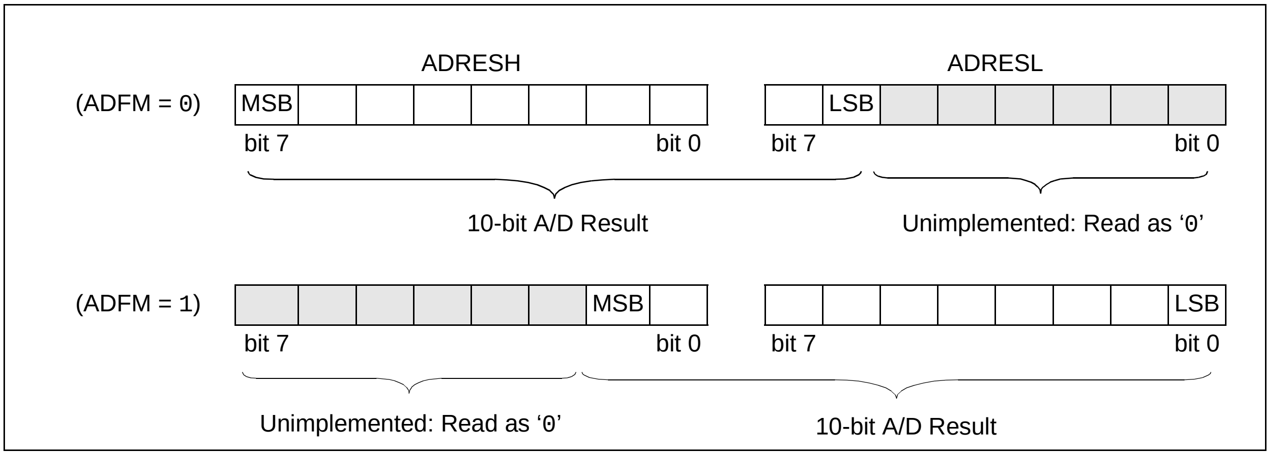
FIGURE 12-2: A/D CONVERSION TAD CYCLES


TCY TO TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion Starts
Holding Capacitor is Disconnected from Analog Input (typically 100 ns)
Set GO/DONE bit
ADRESH and ADRESL registers are loaded,
GO/DONE bit is cleared,
ADIF bit is set,
Holding Capacitor is Connected to Analog Input

12.1.6 CONVERSION OUTPUT

The A/D conversion can be supplied in two formats: left or right shifted. The ADFM bit (ADCON0<7>) controls the output format. Figure 12-3 shows the output formats.

FIGURE 12-3: 10-BIT A/D RESULT FORMAT



REGISTER 12-1: ANSEL – ANALOG SELECT REGISTER (ADDRESS: 91h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7[(2)]	ANS6[(2)]	ANS5[(2)]	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7-0: **ANS<7:0>**: Analog Select bits[(2)]

Select between analog or digital function on pins AN<7:0>, respectively.

1 = Analog input. Pin is assigned as analog input.[(1)]
0 = Digital I/O. Pin is assigned to port or special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

2: ANS<7:5> on PIC16F914/917 only; forced ‘0’ on PIC16F913/916.

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

REGISTER 12-2: ADCON0 – A/D CONTROL REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7 **ADFM:** A/D Result Formed Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG1:** Voltage Reference bit
1 = V_{REF}- pin
0 = V_{SS}

bit 5 **VCFG0:** Voltage Reference bit
1 = V_{REF}+ pin
0 = V_{DD}

bit 4-2 **CHS<2:0>: Analog Channel Select bits**

000	= Channel 00 (AN0)
001	= Channel 01 (AN1)
010	= Channel 02 (AN2)
011	= Channel 03 (AN3)
100	= Channel 04 (AN4)
101	= Channel 05 (AN5)
110	= Channel 06 (AN6)
111	= Channel 07 (AN7)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 ADON: A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut off and consumes no operating current

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

REGISTER 12-3: ADCON1 – A/D CONTROL REGISTER 1 (ADDRESS: 9Fh)

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCS2	ADCS1	ADCS0	—	—	—	—
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **Unimplemented:** Read as ‘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

12.1.7 CONFIGURING THE A/D

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 19.0 “Electrical Specifications”**. After this sample time has elapsed, the A/D conversion can be started.

These steps should be followed for an A/D conversion:

1. Configure the A/D module:
   - Configure analog/digital I/O (ANSEL)
   - Configure voltage reference (ADCON0)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON1)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit (PIR1<6>)
   - Set ADIE bit (PIE1<6>)
   - Set PEIE and GIE bits (INTCON<7:6>)
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (with interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL); clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

EXAMPLE 12-1: A/D CONVERSION

```
;This code block configures the A/D
;for polling, Vdd reference, R/C clock
;and RA0 input.
;
;Conversion start and wait for complete
;polling code included.
;
 BSF    STATUS,RP0     ;Bank 1
 MOVLW  B’01110000’    ;A/D RC clock
 MOVWF  ADCON1
 BSF    TRISA,0        ;Set RA0 to input
 BSF    ANSEL,0        ;Set RA0 to analog
 BCF    STATUS,RP0     ;Bank 0
 MOVLW  B’10000001’    ;Right, Vdd Vref, AN0
 MOVWF  ADCON0
 CALL   SampleTime     ;Wait min sample time
 BSF    ADCON0,GO      ;Start conversion
 BTFSC  ADCON0,GO      ;Is conversion done?
 GOTO   $-1            ;No, test again
 MOVF   ADRESH,W       ;Read upper 2 bits
 MOVWF  RESULTHI
 BSF    STATUS,RP0     ;Bank 1
 MOVF   ADRESL,W       ;Read lower 8 bits
 MOVWF  RESULTLO
```

12.2 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 12-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 12-4. **The maximum recommended impedance for analog sources is 10 kΩ**. As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 12-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

EQUATION 12-1: ACQUISITION TIME

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\mu s/°C)] \end{aligned}$$

Where CHOLD is charged to within 1/2 lsb:

$$V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \qquad ;[1]\ V_{CHOLD}\ \text{charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \qquad ;[2]\ V_{CHOLD}\ \text{charge response to}\ V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{2047}\right) \qquad ;\text{combining [1] and [2]}$$

Solving for TC:

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

Therefore:

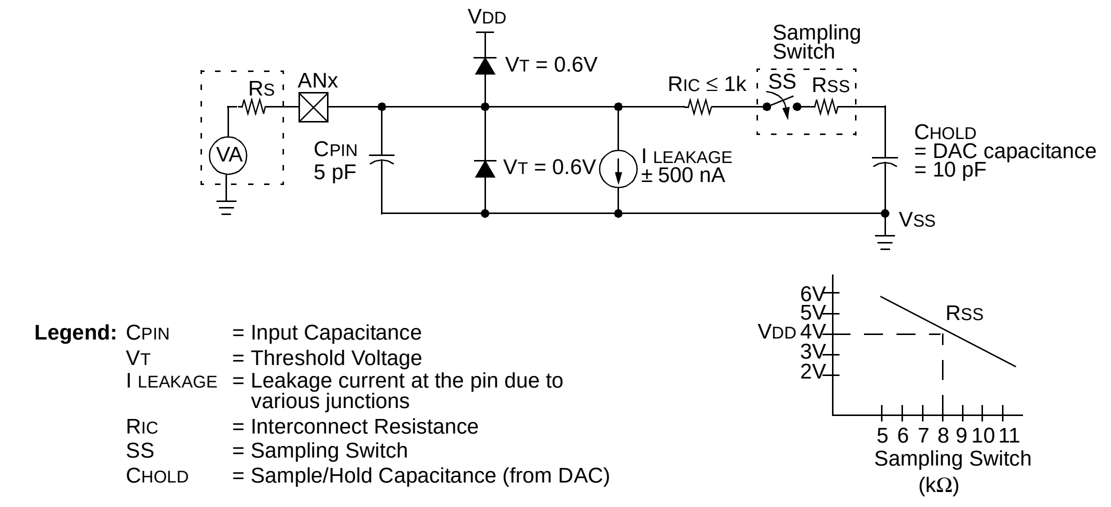
$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 4.67\mu s \end{aligned}$$

Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

FIGURE 12-4: ANALOG INPUT MODEL

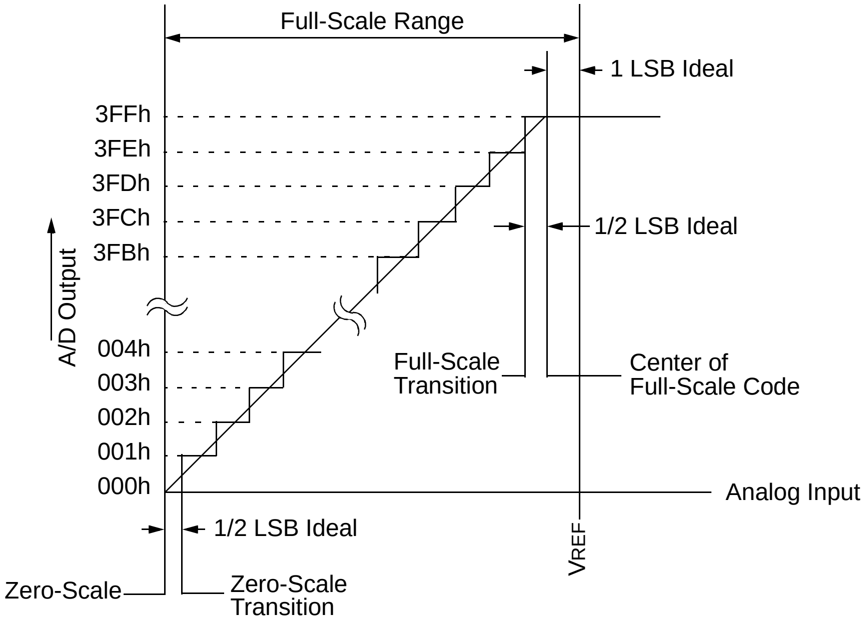

VDD
VT = 0.6V
Sampling Switch
RS
ANx
RIC ≤ 1k
SS
RSS
VA
CPIN 5 pF
VT = 0.6V
I LEAKAGE ± 500 nA
CHOLD = DAC capacitance = 10 pF
VSS
Legend: CPIN = Input Capacitance
VT = Threshold Voltage
I LEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance (from DAC)
6V
5V
VDD 4V
3V
2V
RSS
5 6 7 8 9 10 11
Sampling Switch (kΩ)

12.3 A/D Operation During Sleep

The A/D converter module can operate during Sleep. This requires the A/D clock source to be set to the internal oscillator. When the RC clock source is selected, the A/D waits one instruction before starting the conversion. This allows the SLEEP instruction to be executed, thus eliminating much of the switching noise from the conversion. When the conversion is complete, the GO/$\overline{\text{DONE}}$ bit is cleared and the result is loaded into the ADRESH:ADRESL registers. If the A/D interrupt is enabled, the device awakens from Sleep. If the GIE bit (INTCON<7>) is set, the program counter is set to the interrupt vector (0004h). If GIE is clear, the next instruction is executed. If the A/D interrupt is not enabled, the A/D module is turned off, although the ADON bit remains set.

When the A/D clock source is something other than RC, a SLEEP instruction causes the present conversion to be aborted, and the A/D module is turned off. The ADON bit remains set.

FIGURE 12-5: A/D TRANSFER FUNCTION



12.4 Effects of Reset

A device Reset forces all registers to their Reset state. Thus, the A/D module is turned off and any pending conversion is aborted. The ADRESH:ADRESL registers are unchanged.

TABLE 12-2: SUMMARY OF A/D REGISTERS

Addr	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
05h/ 105h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	`xxxx xxxx`	`uuuu uuuu`
09h	PORTE	—	—	—	—	RE3	RE2	RE1	RE0	`---- xxxx`	`---- uuuu`
0Bh/ 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000x`
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
1Eh	ADRESH	Most Significant 8 bits of the left justified A/D result or 2 bits of the right justified result								`xxxx xxxx`	`uuuu uuuu`
1Fh	ADCON0	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	`0000 0000`	`0000 0000`
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`1111 1111`	`1111 1111`
89h	TRISE	—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0	`---- 1111`	`---- 1111`
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	`0000 0000`	`0000 0000`
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	`1111 1111`	`1111 1111`
9Eh	ADRESL	Least Significant 2 bits of the left justified A/D result or 8 bits of the right justified result								`xxxx xxxx`	`uuuu uuuu`
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	`-000 ----`	`-000 ----`

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D module.

NOTES:

13.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

Data EEPROM memory is readable and writable and the Flash program memory is readable during normal operation (full VDD range). These memories are not directly mapped in the register file space. Instead, they are indirectly addressed through the Special Function Registers. There are six SFRs used to access these memories:

- EECON1
- EECON2
- EEDATL
- EEDATH
- EEADRL
- EEADRH

When interfacing the data memory block, EEDATL holds the 8-bit data for read/write, and EEADRL holds the address of the EE data location being accessed. This device has 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When interfacing the program memory block, the EEDATL and EEDATH registers form a 2-byte word that holds the 14-bit data for read, and the EEADRL and EEADRH registers form a 2-byte word that holds the 13-bit address of the EEPROM location being accessed. This device has 4k and 8k words of program EEPROM with an address range from 0h-0FFFh and 0h-1FFFh. The program memory allows one word reads.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory and read the program memory. When code-protected, the device programmer can no longer access data or program memory.

Additional information on the data EEPROM is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

13.1 EEADRL and EEADRH Registers

The EEADRL and EEADRH registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8k words of program EEPROM.

When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADRL register. When selecting a data address value, only the LSB of the address is written to the EEADRL register.

13.1.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for EE memory accesses.

Control bit EEPGD determines if the access will be a program or data memory access. When clear, as it is when reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory. Program memory can only be read.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to data EEPROM. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATL and EEADRL registers.

Interrupt flag bit EEIF (PIR1<7>), is set when write is complete. It must be cleared in the software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence.

REGISTER 13-1: EEDATL – EEPROM DATA LOW BYTE REGISTER (ADDRESS: 10Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0
bit 7							bit 0

bit 7-0 **EEDATL<7:0>**: Byte value to Write to or Read from data EEPROM bits or to Read from program memory

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

REGISTER 13-2: EEADRL – EEPROM ADDRESS LOW BYTE REGISTER (ADDRESS: 10Dh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0
bit 7							bit 0

bit 7-0 **EEADRL<7:0>**: Specifies one of 256 locations for EEPROM Read/Write Operation bits or low byte for program memory reads

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

REGISTER 13-3: EEDATH – EEPROM DATA HIGH BYTE REGISTER (ADDRESS: 10Eh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 0

bit 5-0 **EEDATH<5:0>**: Byte value to Write to or Read from data EEPROM bits or to Read from program memory

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

REGISTER 13-4: EEADRH – EEPROM ADDRESS HIGH BYTE REGISTER (ADDRESS: 10Fh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0
bit 7							bit 0

bit 4-0 **EEADRH<4:0>**: Specifies one of 256 locations for EEPROM Read/Write Operation bits or high bits for program memory reads

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

REGISTER 13-5: EECON1 – EEPROM CONTROL REGISTER 1 (ADDRESS: 18Ch)

R/W-0	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD:** Program/Data EEPROM Select bit

1 = Accesses program memory
0 = Accesses data memory

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **WRERR:** EEPROM Error Flag bit

1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset, any WDT Reset during normal operation or BOR Reset)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit

1 = Allows write cycles
0 = Inhibits write to the data EEPROM

bit 1 **WR**: Write Control bit

EEPGD = 1:
This bit is ignored
EEPGD = 0:
1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
0 = Write cycle to the data EEPROM is complete

bit 0 **RD**: Read Control bit

1 = Initiates a memory read (RD is cleared in hardware. The RD bit can only be set, not cleared, in software.)
0 = Does not initiate an memory read

Legend:			
S = Bit can only be set			
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

13.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADRL register, clear the EEPGD control bit (EECON1<7>), and then set control bit RD (EECON1<0>). The data is available in the very next cycle, in the EEDATL register; therefore, it can be read in the next instruction. EEDATL will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 13-1: DATA EEPROM READ

```
BSF     STATUS,RP1      ;
BCF     STATUS,RP0      ; Bank 2
MOVF    DATA_EE_ADDR,W  ; Data Memory
MOVWF   EEADR           ; Address to read
BSF     STATUS,RP0      ; Bank 3
BCF     EECON1,EEPGD    ; Point to Data
                        ; memory
BSF     EECON1,RD       ; EE Read
BCF     STATUS,RP0      ; Bank 2
MOVF    EEDATA,W        ; W = EEDATA
```

13.1.3 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADRL register and the data to the EEDATL register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the sequence described below is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
    - Write 55h to EECON2 in two steps (first to W, then to EECON2)
    - Write AAh to EECON2 in two steps (first to W, then to EECON2)
    - Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

EXAMPLE 13-2: DATA EEPROM WRITE

```
            BSF    STATUS,RP1   ;
            BSF    STATUS,RP0
            BTFSC  EECON1,WR    ;Wait for write
            GOTO   $-1          ;to complete
            BCF    STATUS,RP0   ;Bank 2
            MOVF   DATA_EE_ADDR,W;Data Memory
            MOVWF  EEADR        ;Address to write
            MOVF   DATA_EE_DATA,W;Data Memory Value
            MOVWF  EEDATA       ;to write
            BSF    STATUS,RP0   ;Bank 3
            BCF    EECON1,EEPGD ;Point to DATA
                                ;memory
            BSF    EECON1,WREN  ;Enable writes

            BCF    INTCON,GIE   ;Disable INTs.
            MOVLW  55h          ;
            MOVWF  EECON2       ;Write 55h
Required    MOVLW  AAh          ;
Sequence    MOVWF  EECON2       ;Write AAh
            BSF    EECON1,WR    ;Set WR bit to
                                ;begin write
            BSF    INTCON,GIE   ;Enable INTs.
            BCF    EECON1,WREN  ;Disable writes
```

13.1.4 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must write two bytes of the address to the EEADRL and EEADRH registers, set the EEPGD control bit (EECON1<7>), and then set control bit RD (EECON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "`BSF EECON1,RD`" instruction to be ignored. The data is available in the very next cycle, in the EEDATL and EEDATH registers; therefore, it can be read as two bytes in the following instructions. EEDATL and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

Note 1: The two instructions following a program memory read are required to be `NOP`'s. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

2: If the WR bit is set when EEPGD = `1`, it will be immediately reset to '`0`' and no operation will take place.

EXAMPLE 13-3: FLASH PROGRAM READ

```
         BSF     STATUS, RP1    ;
         BCF     STATUS, RP0    ; Bank 2
         MOVLW   MS_PROG_EE_ADDR;
         MOVWF   EEADRH         ; MS Byte of Program Address to read
         MOVLW   LS_PROG_EE_ADDR;
         MOVWF   EEADR          ; LS Byte of Program Address to read
         BSF     STATUS, RP0    ; Bank 3
         BSF     EECON1, EEPGD  ; Point to PROGRAM memory
         BSF     EECON1, RD     ; EE Read
     ;
         NOP
         NOP                    ; Any instructions here are ignored as program
                                ; memory is read in second cycle after BSF EECON1,RD
     ;
         BCF     STATUS, RP0    ; Bank 2
         MOVF    EEDATA, W      ; W = LS Byte of Program EEDATA
         MOVWF   DATAL          ;
         MOVF    EEDATH, W      ; W = MS Byte of Program EEDATA
         MOVWF   DATAH          ;
```

Required Sequence

FIGURE 13-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION

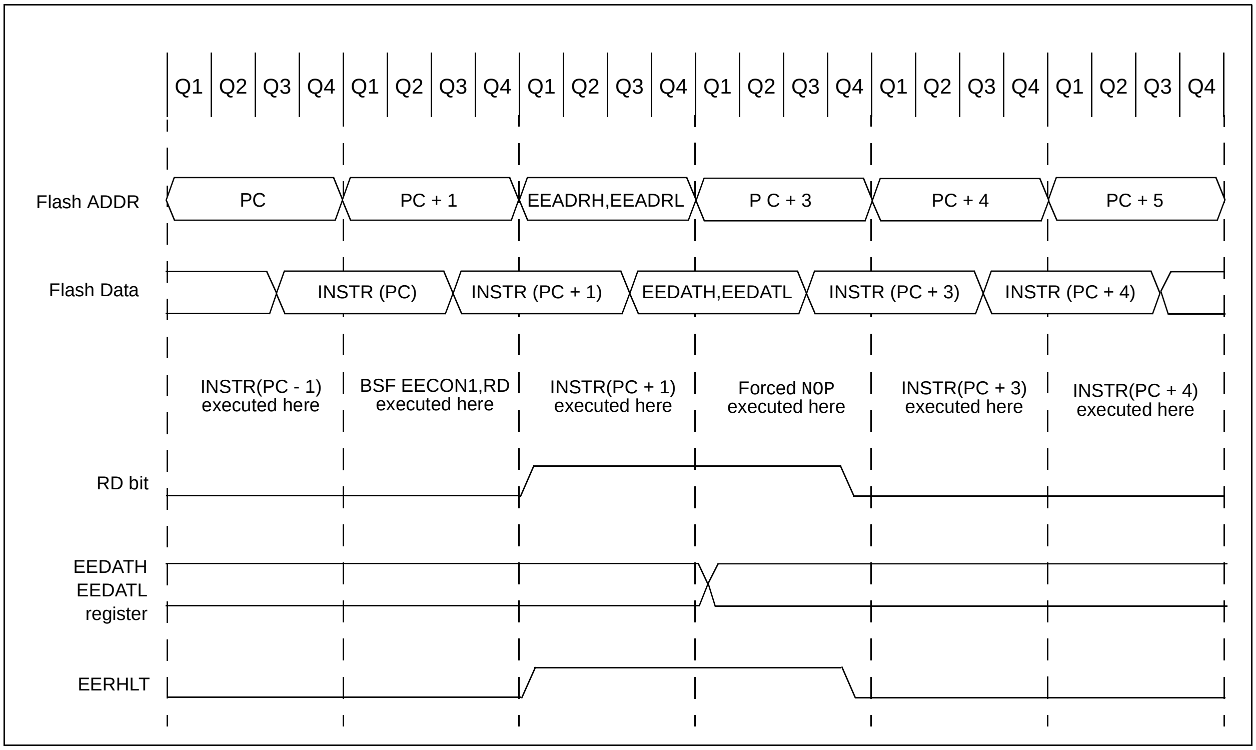

Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
Flash ADDR
PC
PC + 1
EEADRH,EEADRL
P C + 3
PC + 4
PC + 5
Flash Data
INSTR (PC)
INSTR (PC + 1)
EEDATH,EEDATL
INSTR (PC + 3)
INSTR (PC + 4)
INSTR(PC - 1) executed here
BSF EECON1,RD executed here
INSTR(PC + 1) executed here
Forced NOP executed here
INSTR(PC + 3) executed here
INSTR(PC + 4) executed here
RD bit
EEDATH
EEDATL
register
EERHLT

TABLE 13-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

Addr	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
0Bh/8Bh/ 10Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
10Ch	EEDATL	EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0	0000 0000	0000 0000
10Dh	EEADRL	EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0	0000 0000	0000 0000
10Eh	EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
10Fh	EEADRH	—	—	—	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---0 0000	---0 0000
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	---- ----

Legend: x = unknown, u = unchanged, - = unimplemented read as ‘0’, q = value depends upon condition.
Shaded cells are not used by data EEPROM module.

14.0 SSP MODULE OVERVIEW

The Synchronous Serial Port (SSP) module is a serial interface used to communicate with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI™)
- Inter-Integrated Circuit (I^{2}C™)

An overview of I^{2}C operations and additional information on the SSP module can be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

Refer to Application Note AN578, "*Use of the SSP Module in the Multi-Master Environment*" (DS00578).

14.1 SPI Mode

This section contains register definitions and operational characteristics of the SPI module. Additional information on the SPI module can be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

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

- Serial Data Out (SDO) – RC4/$\overline{\text{T1G}}$/SDO/SEG11
- Serial Data In (SDI) – RC7/RX/DT/SDI/SDA/SEG8
- Serial Clock (SCK) – RC6/TX/CK/SCK/SCL/SEG9

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

- Slave Select ($\overline{\text{SS}}$) – RA5/AN4/C2OUT/$\overline{\text{SS}}$/SEG5

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<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)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

REGISTER 14-1: SSPSTAT – SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)

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**: SPI™ Data Input Sample Phase bit

<u>SPI Master mode:</u>
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time (Microwire)
<u>SPI Slave mode:</u>
SMP must be cleared when SPI is used in Slave mode
<u>I^{2}C™ mode:</u>
This bit must be maintained clear

bit 6 **CKE**: SPI Clock Edge Select bit

<u>SPI mode, CKP = 0:</u>
1 = Data transmitted on rising edge of SCK (Microwire alternate)
0 = Data transmitted on falling edge of SCK
<u>SPI mode, CKP = 1:</u>
1 = Data transmitted on falling edge of SCK (Microwire default)
0 = Data transmitted on rising edge of SCK
<u>I^{2}C mode:</u>
This bit must be maintained clear

bit 5 **$D/\overline{A}$**: Data/$\overline{\text{Address}}$ bit (I^{2}C 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^{2}C mode only)
This bit is cleared when the SSP module is disabled, or when the Start bit is detected last. 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^{2}C mode only)
This bit is cleared when the SSP module is disabled, or when the Stop bit is detected last. 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/$\overline{\text{Write}}$ bit Information (I^{2}C mode only)
This bit holds the $R/\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or $\overline{\text{ACK}}$ bit.

1 = Read
0 = Write

bit 1 **UA**: Update Address bit (10-bit I^{2}C 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

<u>Receive (SPI and I^{2}C modes):</u>
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
<u>Transmit (I^{2}C mode only):</u>
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

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

REGISTER 14-2: SSPCON – SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL**: Write Collision Detect bit
`1` = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
`0` = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit
<u>In SPI™ mode:</u>
`1` = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
`0` = No overflow

<u>In I^{2}C™ mode:</u>
`1` = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a “don’t care” in Transmit mode. SSPOV must be cleared in software in either mode.
`0` = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
<u>In SPI mode:</u>
`1` = Enables serial port and configures SCK, SDO, and SDI as serial port pins
`0` = Disables serial port and configures these pins as I/O port pins

<u>In I^{2}C mode:</u>
`1` = Enables the serial port and configures the SDA and SCL pins as serial port pins
`0` = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

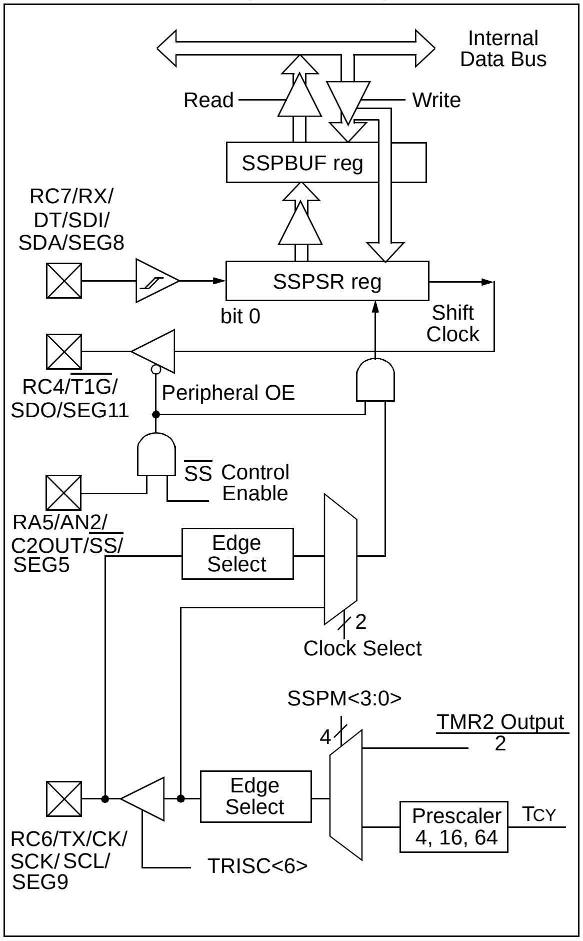
bit 4 **CKP**: Clock Polarity Select bit
<u>In SPI mode:</u>
`1` = Idle state for clock is a high level (Microwire default)
`0` = Idle state for clock is a low level (Microwire alternate)

<u>In I^{2}C mode:</u>
SCK release control
`1` = Enable clock
`0` = Holds clock low (clock stretch). (Used to ensure data setup time.)

bit 3-0 **SSPM<3:0>**: Synchronous Serial Port Mode Select bits
`0000` = SPI Master mode, clock = FOSC/4
`0001` = SPI Master mode, clock = FOSC/16
`0010` = SPI Master mode, clock = FOSC/64
`0011` = SPI Master mode, clock = TMR2 output/2
`0100` = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
`0101` = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
`0110` = I^{2}C Slave mode, 7-bit address
`0111` = I^{2}C Slave mode, 10-bit address
`1011` = I^{2}C Firmware Controlled Master mode (slave idle)
`1110` = I^{2}C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
`1111` = I^{2}C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

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

FIGURE 14-1: SSP BLOCK DIAGRAM (SPI MODE)



To enable the serial port, SSPEN bit (SSPCON<5>) must be set. To reset or reconfigure SPI mode:

- Clear bit SSPEN
- Re-initialize the SSPCON register
- Set SSPEN bit

This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave in a serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. This is:

- SDI must have TRISC<7> set
- SDO must have TRISC<4> cleared
- SCK (Master mode) must have TRISC<6> cleared
- SCK (Slave mode) must have TRISC<6> set
- SS must have TRISA<5> set.

.

Note 1: When the SPI is in Slave mode with $\overline{\text{SS}}$ pin control enabled (SSPCON<3:0> = `0100`), the SPI module will reset if the $\overline{\text{SS}}$ pin is set to VDD.

2: If the SPI is used in Slave mode with CKE = 1, then the $\overline{\text{SS}}$ pin control must be enabled.

3: When the SPI is in Slave mode with $\overline{\text{SS}}$ pin control enabled (SSPCON<3:0> = `0100`), the state of the $\overline{\text{SS}}$ pin can affect the state read back from the TRISC<4> bit. The peripheral OE signal from the SSP module into PORTC controls the state that is read back from the TRISC<4> bit (see **Section 19.4 "DC Characteristics: PIC16F91X-I (Industrial), PIC16F91X-E (Extended)"** for information on PORTC). If read-modify-write instructions, such as `BSF`, are performed on the TRISC register while the $\overline{\text{SS}}$ pin is high, this will cause the TRISC<4> bit to be set, thus disabling the SDO output.

14.2 Operation

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<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)

The SSP 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 eight 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, 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 (SSPCON<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.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the SSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 14-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 14-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP  BTFSS  SSPSTAT, BF    ;Has data been received(transmit complete)?
      BRA    LOOP           ;No
      MOVF   SSPBUF, W      ;WREG reg = contents of SSPBUF
      MOVWF  RXDATA         ;Save in user RAM, if data is meaningful
      MOVF   TXDATA, W      ;W reg = contents of TXDATA
      MOVWF  SSPBUF         ;New data to xmit
```

14.3 Enabling SPI I/O

To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$ must have TRISC<4> bit set

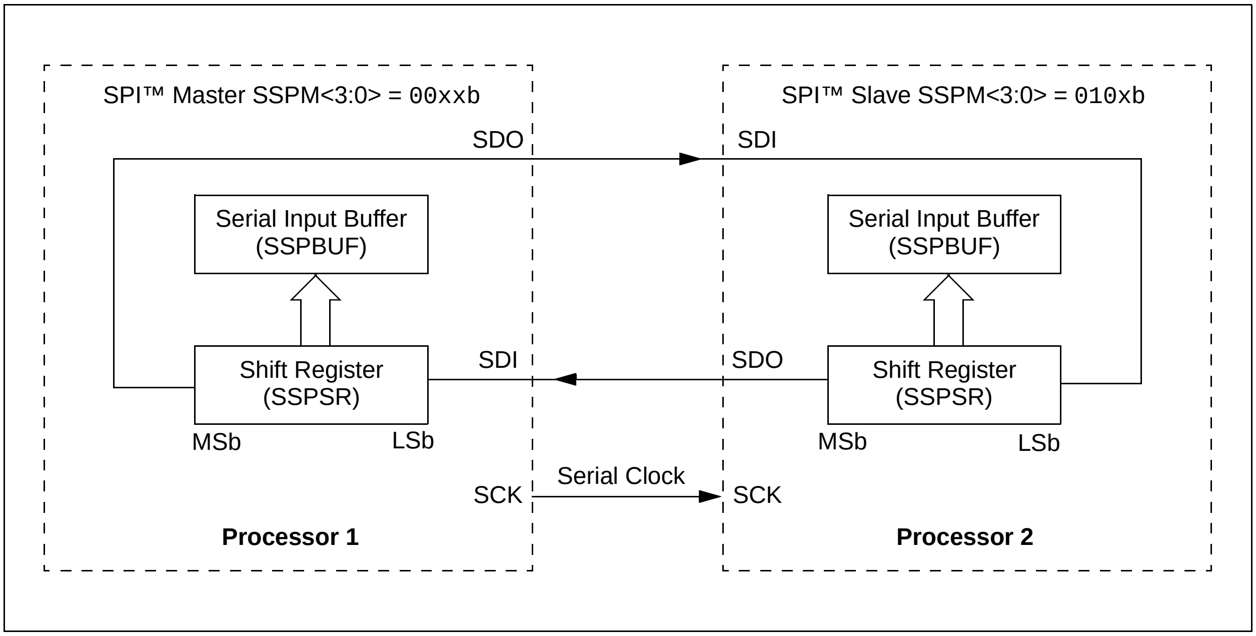
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

14.4 Typical Connection

Figure 14-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 14-2: SPI™ MASTER/SLAVE CONNECTION


SPI™ Master SSPM<3:0> = 00xxb
SDO
SDI
Serial Input Buffer (SSPBUF)
Shift Register (SSPSR)
MSb
LSb
SCK
Processor 1
SPI™ Slave SSPM<3:0> = 010xb
SDI
SDO
Serial Input Buffer (SSPBUF)
Shift Register (SSPSR)
MSb
LSb
SCK
Serial Clock
Processor 2

14.5 Master Mode

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 14-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and Status bits appropriately set). This could be useful in receiver applications as a “Line Activity Monitor” mode.

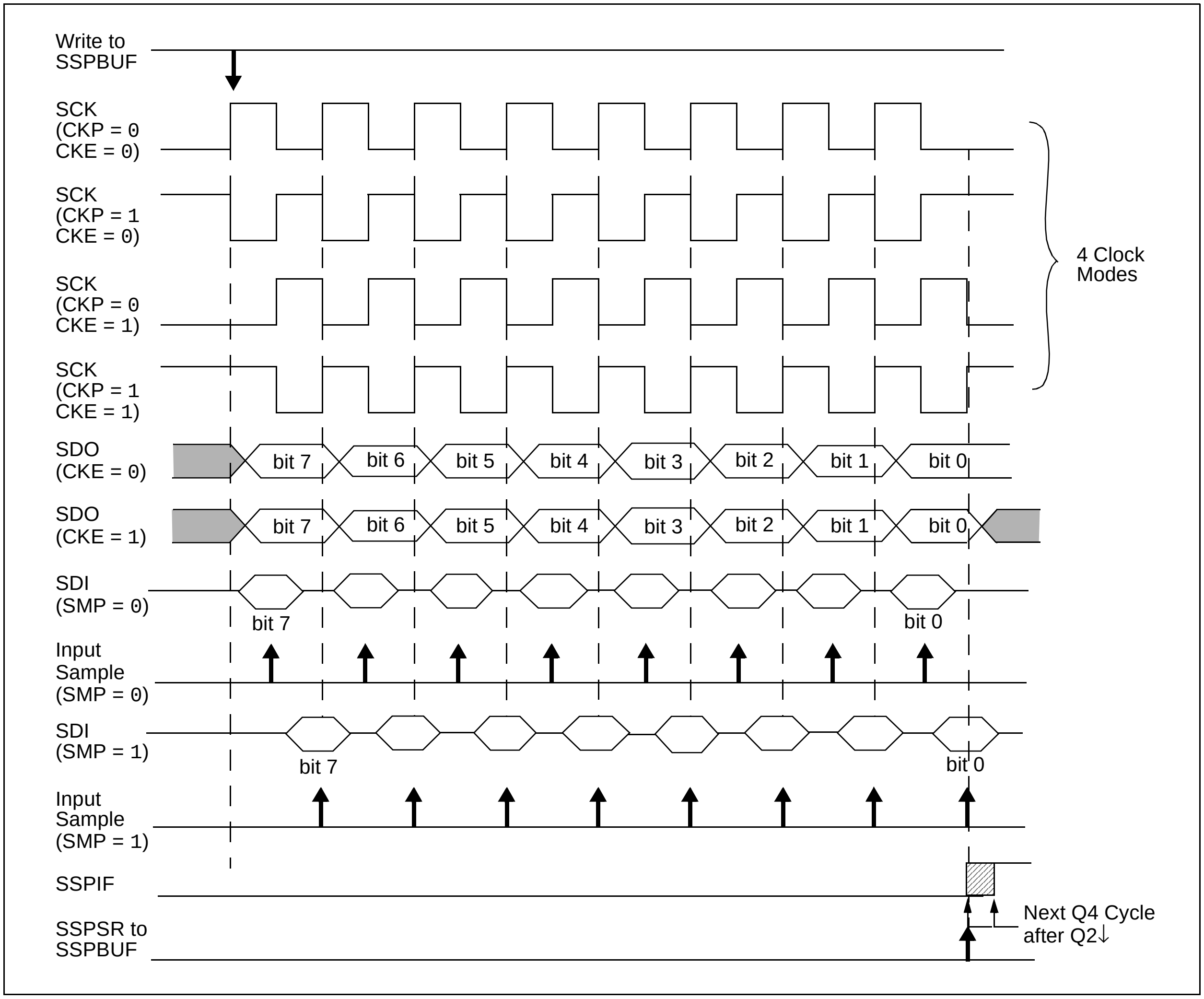
The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then, would give waveforms for SPI communication as shown in Figure 14-3, Figure 14-5 and Figure 14-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum data rate (at 40 MHz) of 10 Mbps.

Figure 14-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 14-3: SPI™ MODE WAVEFORM (MASTER MODE)



14.6 Slave Mode

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

14.7 Slave Select Synchronization

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = `0100`), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

2: If the SPI is used in Slave Mode with CKE set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '`0`'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 14-4: SLAVE SYNCHRONIZATION WAVEFORM

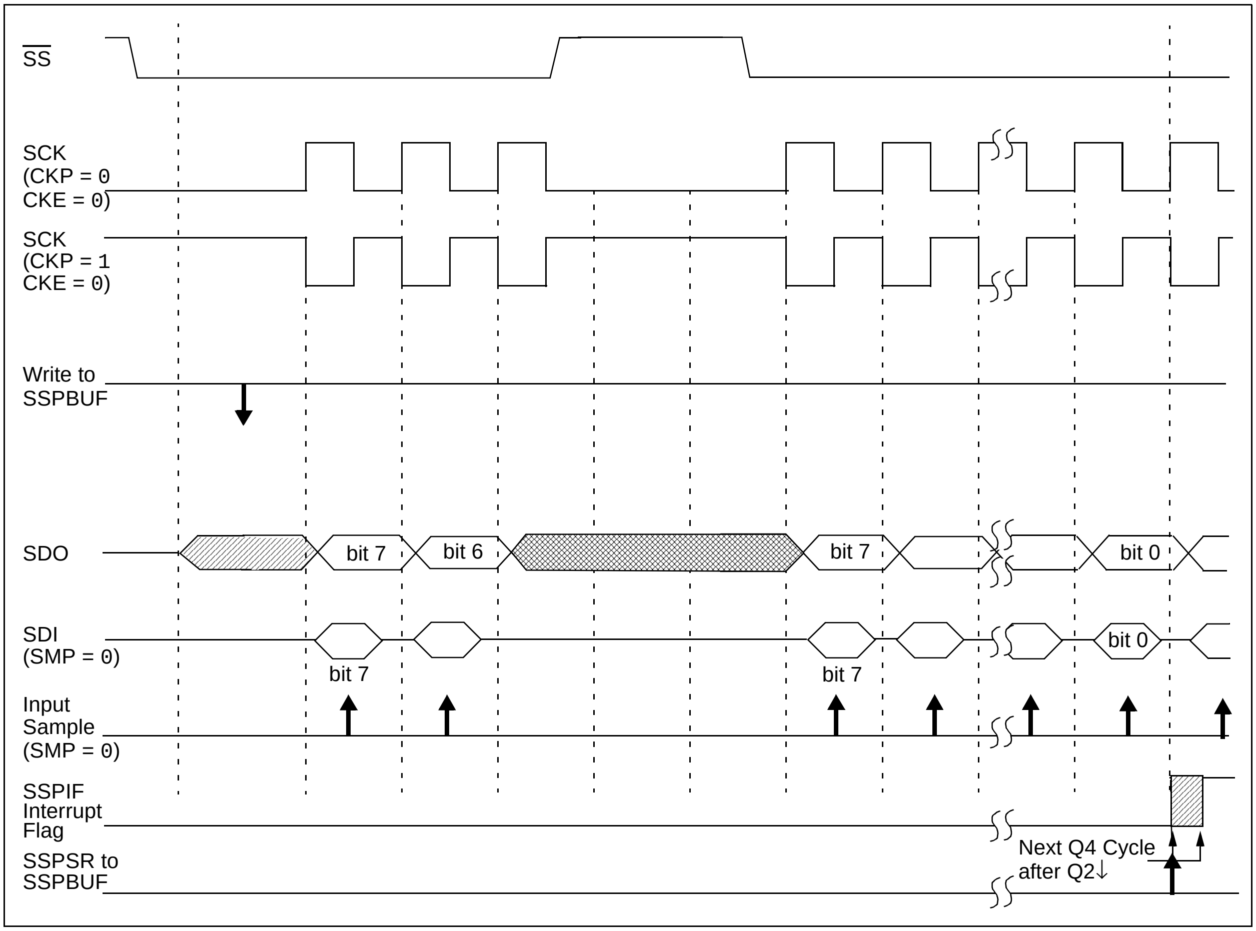


FIGURE 14-5: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

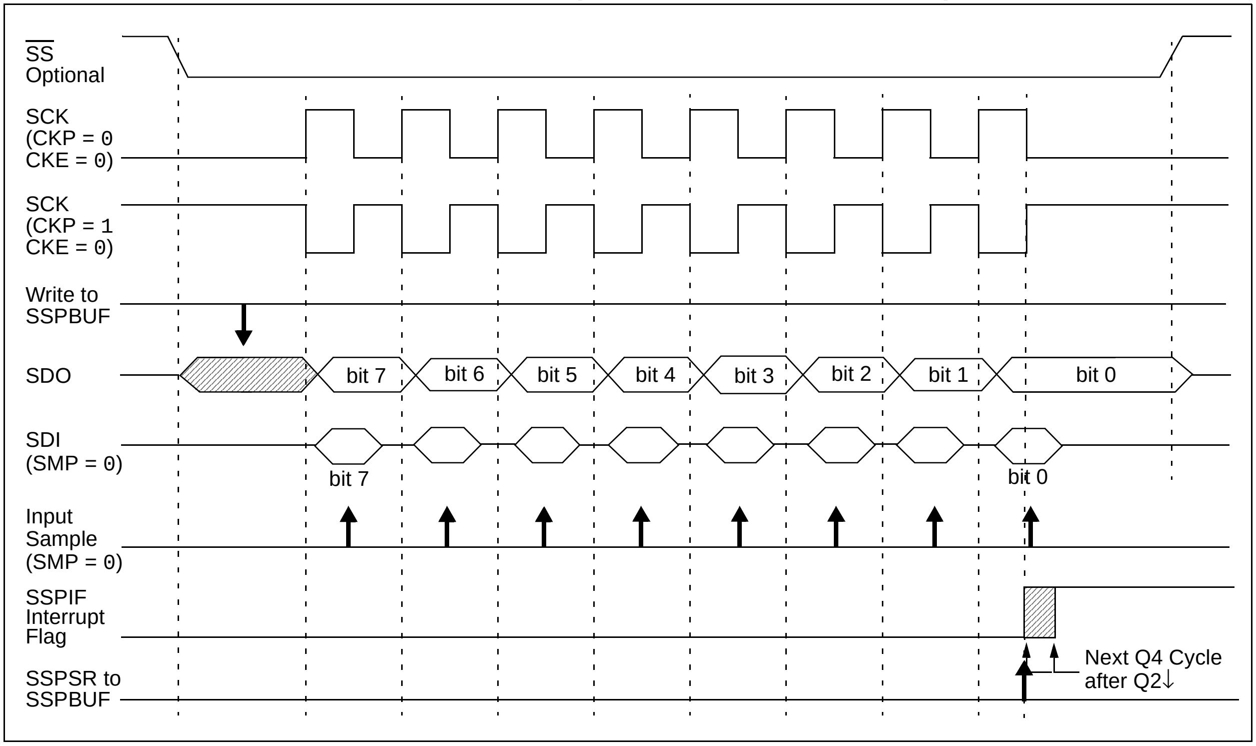
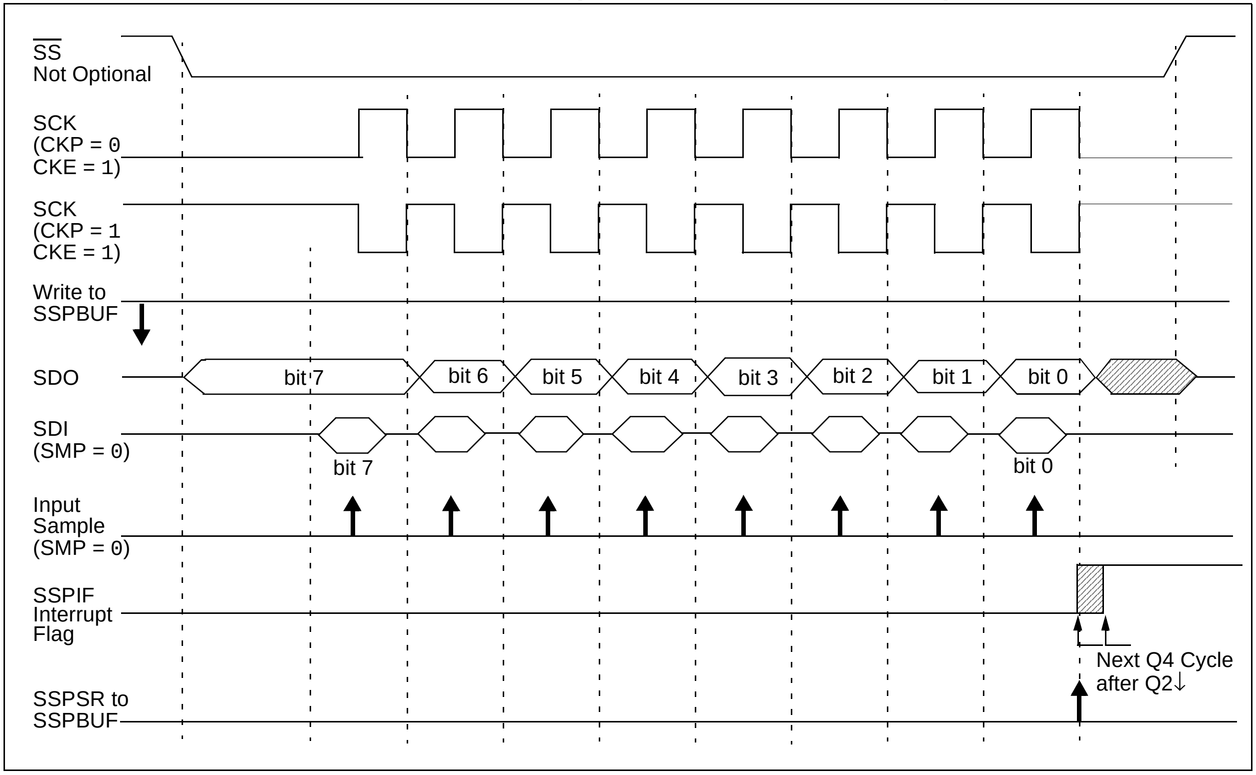


FIGURE 14-6: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



14.8 Sleep Operation

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the SSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

14.9 Effects of a Reset

A Reset disables the SSP module and terminates the current transfer.

14.10 Bus Mode Compatibility

Table 14-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 14-1: SPI™ BUS MODES

Standard SPI™ Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also a SMP bit which controls when the data is sampled.

TABLE 14-2: REGISTERS ASSOCIATED WITH SPI™ OPERATION

Address	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
0Bh,8Bh. 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000x`
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								`xxxx xxxx`	`uuuu uuuu`
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	`0000 0000`	`0000 0000`
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`1111 1111`	`1111 1111`
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	`0000 0000`	`0000 0000`
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`1111 1111`	`1111 1111`
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	`0000 0000`	`0000 0000`

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

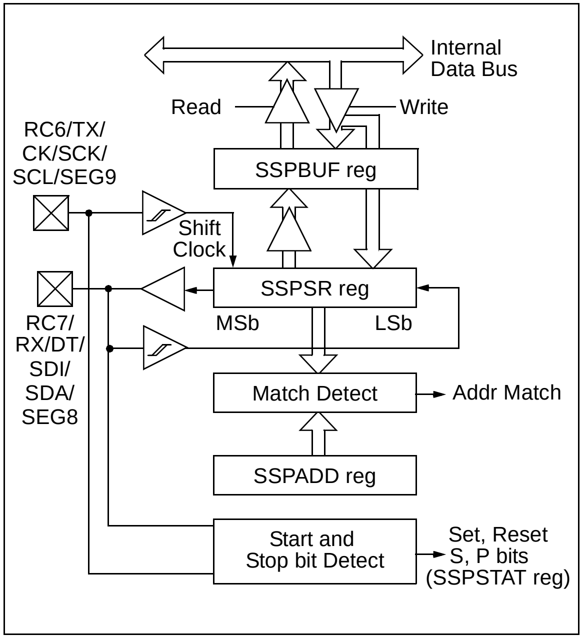
14.11 SSP I^{2}C Operation

The SSP module in I^{2}C mode, fully implements all slave functions, except general call support, and provides interrupts on Start and Stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RC6/TX/CK/SCK/SCL/SEG9 pin, which is the clock (SCL), and the RC7/RX/DT/SDI/SDA/SEG8 pin, which is the data (SDA).

The SSP module functions are enabled by setting SSP enable bit SSPEN (SSPCON<5>).

FIGURE 14-7: SSP BLOCK DIAGRAM (I^{2}C™ MODE)



The SSP module has five registers for the I^{2}C operation, which are listed below.

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I^{2}C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I^{2}C modes to be selected:

- I^{2}C Slave mode (7-bit address)
- I^{2}C Slave mode (10-bit address)
- I^{2}C Slave mode (7-bit address), with Start and Stop bit interrupts enabled to support Firmware Master mode
- I^{2}C Slave mode (10-bit address), with Start and Stop bit interrupts enabled to support Firmware Master mode
- I^{2}C Start and Stop bit interrupts enabled to support Firmware Master mode; Slave is idle

Selection of any I^{2}C mode with the SSPEN bit set forces the SCL and SDA pins to be open drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I^{2}C module.

Additional information on SSP I^{2}C operation can be found in the “*PICmicro® Mid-Range MCU Family Reference Manual*” (DS33023).

14.12 Slave Mode

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<7:6> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ($\overline{\text{ACK}}$) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the SSP module not to give this $\overline{\text{ACK}}$ pulse. They include (either or both):

a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.

b) The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 14-3 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. For high and low times of the I^{2}C specification, as well as the requirements of the SSP module, see **Section 19.0 “Electrical Specifications”**.

14.12.1 ADDRESSING

Once the SSP 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.

b) The buffer full bit, BF is set.

c) An $\overline{\text{ACK}}$ pulse is generated.

d) SSP interrupt flag bit, SSPIF (PIR1<3>) is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave (Figure 14-8). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{\text{W}}$ (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 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; if match releases SCL line, this will clear bit UA.
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.

TABLE 14-3: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		**SSPSR → SSPBUF**	**Generate $\overline{\text{ACK}}$ 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	No	No	Yes

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

14.12.2 RECEPTION

When the R/$\overline{\text{W}}$ bit of the address byte is clear and an address match occurs, the R/$\overline{\text{W}}$ 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 ($\overline{\text{ACK}}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set. This is an error condition due to the user's firmware.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

FIGURE 14-8: I²C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)

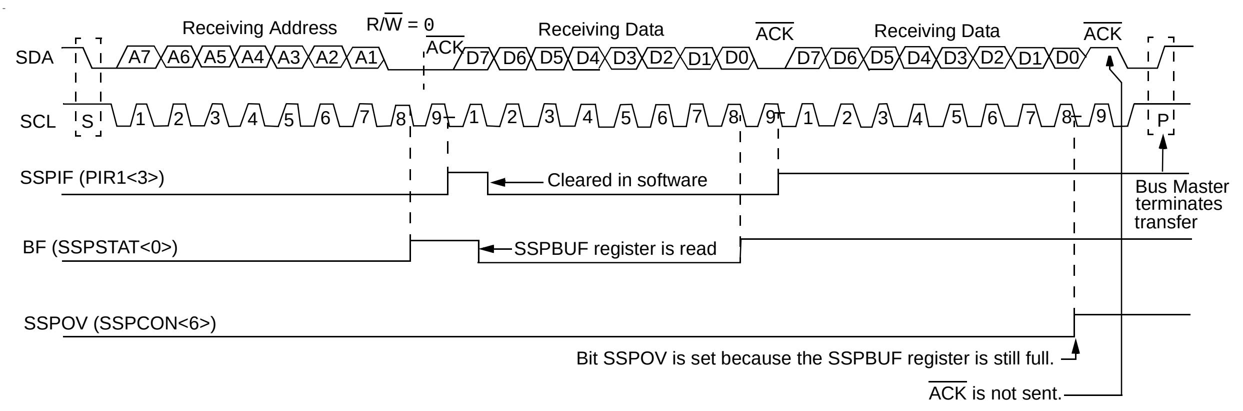
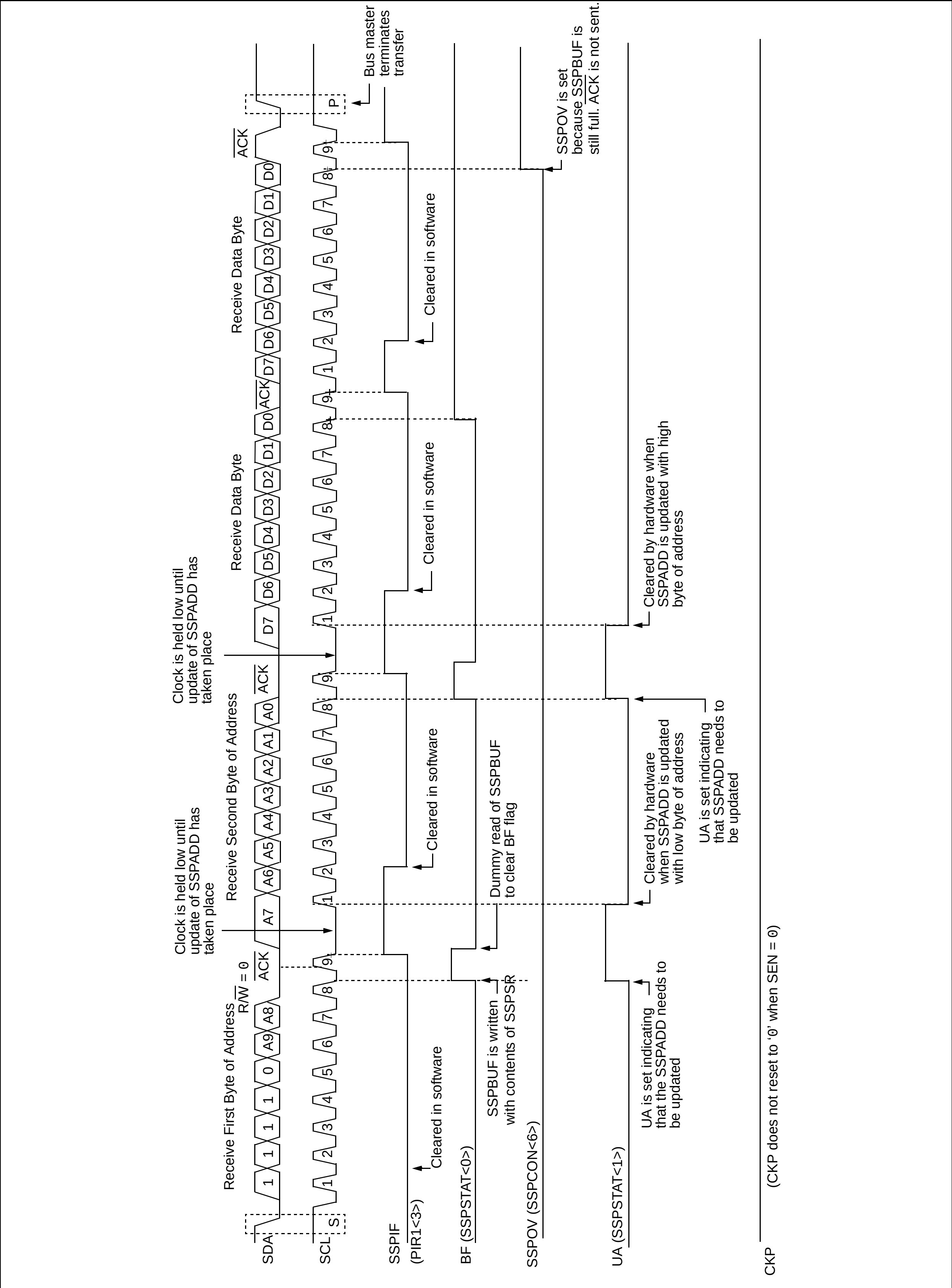


FIGURE 14-9: I²C™ SLAVE MODE TIMING (RECEPTION, 10-BIT ADDRESS)


SDA
SCL
S
P
1 1 1 1 0 A9 A8
$R/\overline{W}$ = 0
$\overline{ACK}$
A7 A6 A5 A4 A3 A2 A1 A0
D7 D6 D5 D4 D3 D2 D1 D0
1 2 3 4 5 6 7 8 9
Receive First Byte of Address
Receive Second Byte of Address
Receive Data Byte
Receive Data Byte
Clock is held low until update of SSPADD has taken place
Clock is held low until update of SSPADD has taken place
Bus master terminates transfer
SSPIF
(PIR1<3>)
Cleared in software
Cleared in software
Cleared in software
Cleared in software
BF (SSPSTAT<0>)
SSPBUF is written with contents of SSPSR
Dummy read of SSPBUF to clear BF flag
SSPOV (SSPCON<6>)
SSPOV is set because SSPBUF is still full. ACK is not sent.
UA (SSPSTAT<1>)
UA is set indicating that the SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with low byte of address
UA is set indicating that SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with high byte of address
CKP
(CKP does not reset to '0' when SEN = 0)

14.12.3 TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit, and pin RC6/TX/CK/SCK/SCL/SEG9 is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RC6/TX/CK/SCK/SCL/SEG9 should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 14-10).

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave then monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC6/TX/CK/SCK/SCL/SEG9 should be enabled by setting bit CKP.

FIGURE 14-10: I²C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)

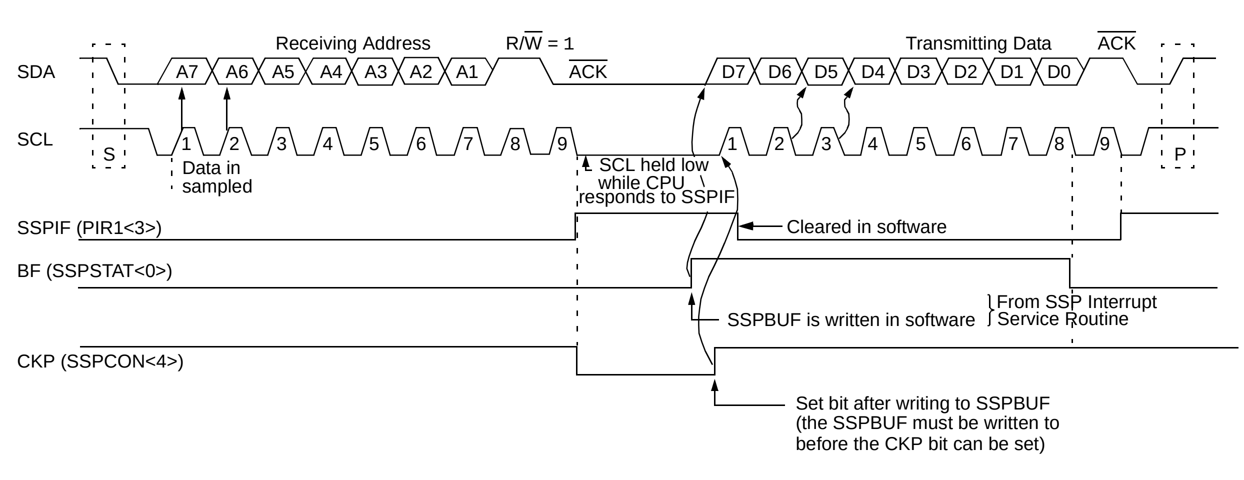
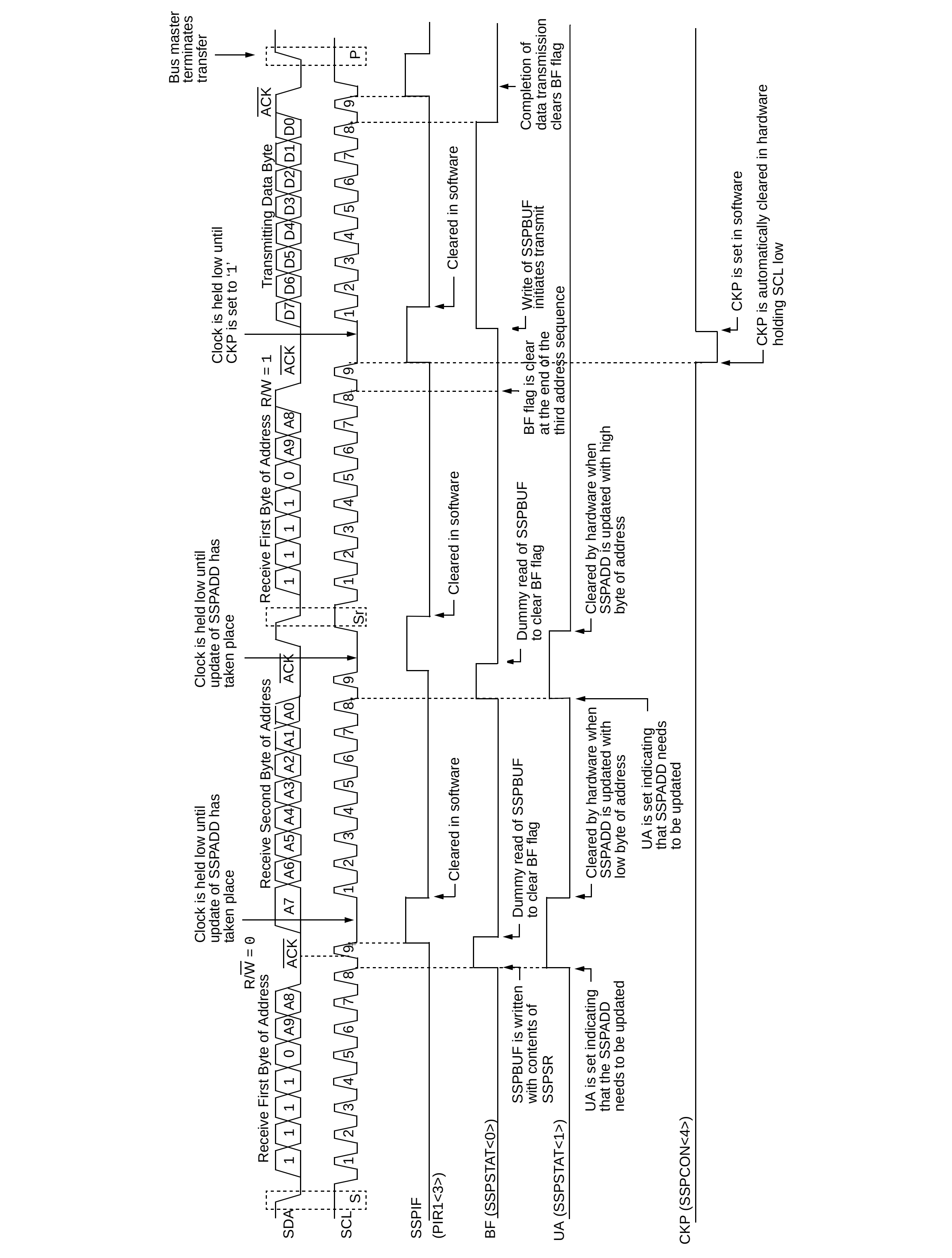

Receiving Address
R/W = 1
Transmitting Data
ACK
SDA
A7
A6
A5
A4
A3
A2
A1
ACK
D7
D6
D5
D4
D3
D2
D1
D0
SCL
S
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9
P
Data in sampled
SCL held low while CPU responds to SSPIF
SSPIF (PIR1<3>)
Cleared in software
BF (SSPSTAT<0>)
SSPBUF is written in software
From SSP Interrupt Service Routine
CKP (SSPCON<4>)
Set bit after writing to SSPBUF (the SSPBUF must be written to before the CKP bit can be set)

FIGURE 14-11: I²C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)


SDA
SCL
SSPIF
(PIR1<3>)
BF (SSPSTAT<0>)
UA (SSPSTAT<1>)
CKP (SSPCON<4>)
S
1
2
3
4
5
6
7
8
9
Receive First Byte of Address
1
1
1
1
0
A9
A8
R/W = 0
ACK
Clock is held low until update of SSPADD has taken place
Receive Second Byte of Address
A7
A6
A5
A4
A3
A2
A1
A0
ACK
Clock is held low until update of SSPADD has taken place
Sr
Receive First Byte of Address R/W = 1
1
1
1
1
0
A9
A8
ACK
Clock is held low until CKP is set to '1'
Transmitting Data Byte
D7
D6
D5
D4
D3
D2
D1
D0
ACK
Bus master terminates transfer
P
Cleared in software
Cleared in software
Cleared in software
Cleared in software
SSPBUF is written with contents of SSPSR
Dummy read of SSPBUF to clear BF flag
Dummy read of SSPBUF to clear BF flag
BF flag is clear at the end of the third address sequence
Write of SSPBUF initiates transmit
Completion of data transmission clears BF flag
UA is set indicating that the SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with low byte of address
UA is set indicating that SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with high byte of address
CKP is set in software
CKP is automatically cleared in hardware holding SCL low

14.13 Master Mode

Master mode of operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when the P bit is set or the bus is idle and both the S and P bits are clear.

In Master mode, the SCL and SDA lines are manipulated by clearing the corresponding TRISC<6:7> bit(s). The output level is always low, irrespective of the value(s) in PORTC<6:7>. So when transmitting data, a '1' data bit must have the TRISC<7> bit set (input) and a '0' data bit must have the TRISC<7> bit cleared (output). The same scenario is true for the SCL line with the TRISC<6> bit. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I^2C module.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt will occur if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Master mode of operation can be done with either the Slave mode idle (SSPM<3:0> = 1011), or with the Slave active. When both Master and Slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

14.14 Multi-master Mode

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions, allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I^2C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<6:7>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an $\overline{\text{ACK}}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

14.14.1 CLOCK SYNCHRONIZATION AND THE CKP BIT

When the CKP bit is cleared, the SCL output is forced to '0'; however, setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 14-12).

FIGURE 14-12: CLOCK SYNCHRONIZATION TIMING

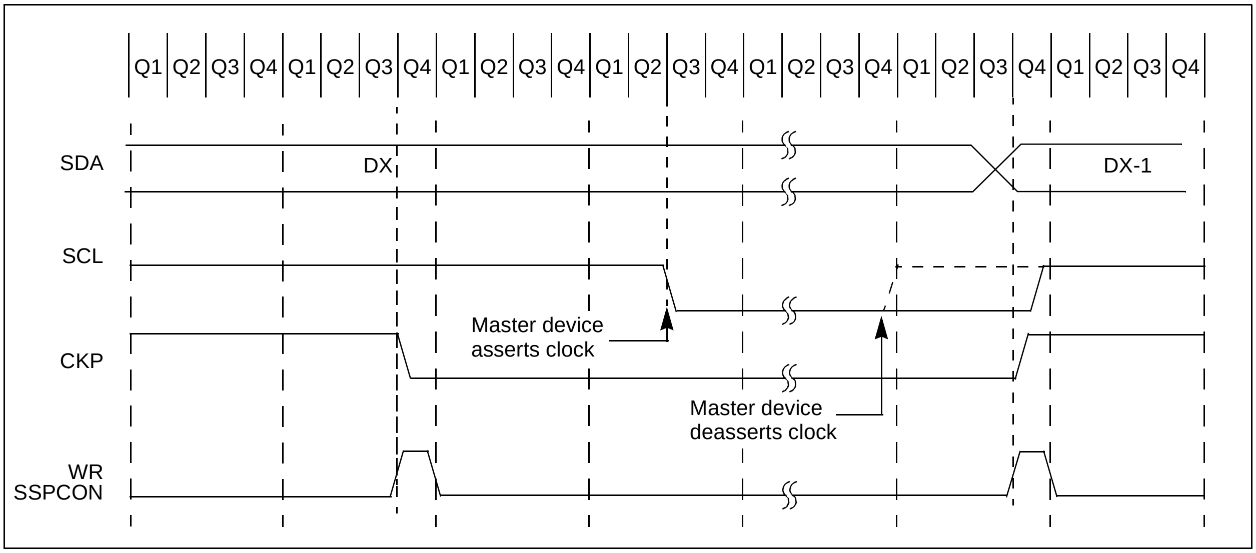


TABLE 14-4: REGISTERS ASSOCIATED WITH I^{2}C™ OPERATION

Address	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
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000x`
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	`0000 0000`	`0000 0000`
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								`xxxx xxxx`	`uuuu uuuu`
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	`0000 0000`	`0000 0000`
87h	TRISC	PORTC Data Direction Register								`1111 1111`	`1111 1111`
93h	SSPADD	Synchronous Serial Port (I^{2}C™ mode) Address Register								`0000 0000`	`0000 0000`
94h	SSPSTAT	SMP[(1)]	CKE[(1)]	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	`0000 0000`	`0000 0000`

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by SSP module in I^{2}C mode.

Note 1: Maintain these bits clear in I^{2}C mode.

15.0 CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Table 15-1 and Table 15-2 show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1, except where noted.

CCP1 Module:

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

CCP2 Module:

Capture/Compare/PWM Register2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023) and in Application Note AN594, *"Using the CCP Modules"* (DS00594).

TABLE 15-1: CCP MODE - TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 15-2: INTERACTION OF TWO CCP MODULES

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time base
Capture	Compare	The compare should be configured for the special event trigger, which clears TMR1
Compare	Compare	The compare(s) should be configured for the special event trigger, which clears TMR1
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt)
PWM	Capture	None
PWM	Compare	None

REGISTER 15-1: CCP1CON – CCP2CON[1] REGISTER (ADDRESS: 17h/1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

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

bit 5-4 **CCPxX:CCPxY**: PWM Least Significant bits

Capture mode:
Unused

Compare mode:
Unused

PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM<3:0>**: CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

Note 1: CCP2CON used for PIC16F914/917 only.

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

15.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC5/T1CKI/CCP1/SEG10. An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

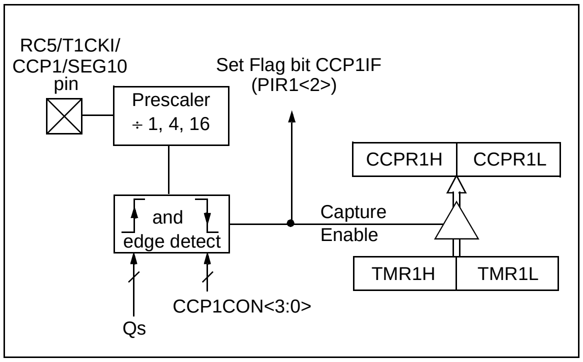
The type of event is configured by control bits CCP1M<3:0> (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new value.

15.1.1 CCP PIN CONFIGURATION

In Capture mode, the RC5/T1CKI/CCP1/SEG10 pin should be configured as an input by setting the TRISC<2> bit.

Note:	If the RC5/T1CKI/CCP1/SEG10 pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 15-3: CAPTURE MODE OPERATION BLOCK DIAGRAM



15.1.2 TIMER1 MODE SELECTION

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

15.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF, following any such change in operating mode.

15.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M<3:0>. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON     ; Turn CCP module off
MOVLW   NEW_CAPT_PS ; Load the W reg with
                    ; the new prescaler
                    ; move value and CCP ON
MOVWF   CCP1CON     ; Load CCP1CON with this
                    ; value
```

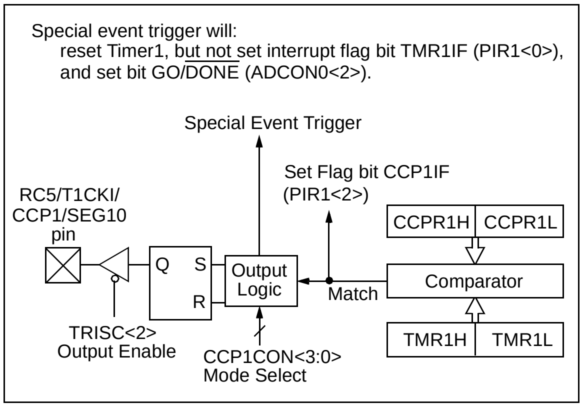
15.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC5/T1CKI/CCP1/SEG10 pin is:

- Driven high
- Driven low
- Remains unchanged

The action on the pin is based on the value of control bits CCP1M<3:0> (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

FIGURE 15-4: COMPARE MODE OPERATION BLOCK DIAGRAM



15.2.1 CCP PIN CONFIGURATION

The user must configure the RC5/T1CKI/CCP1/SEG10 pin as an output by clearing the TRISC<2> bit.

Note:	Clearing the CCP1CON register will force the RC5/T1CKI/CCP1/SEG10 compare output latch to the default low level. This is not the PORTC I/O data latch.

15.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

15.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the RC5/T1CKI/CCP1/SEG10 pin is not affected. The CCPIF bit is set, causing a CCP interrupt (if enabled).

15.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated, which may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of CCP2 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

Note:	The special event trigger from the CCP1and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>).

15.3 PWM Mode (PWM)

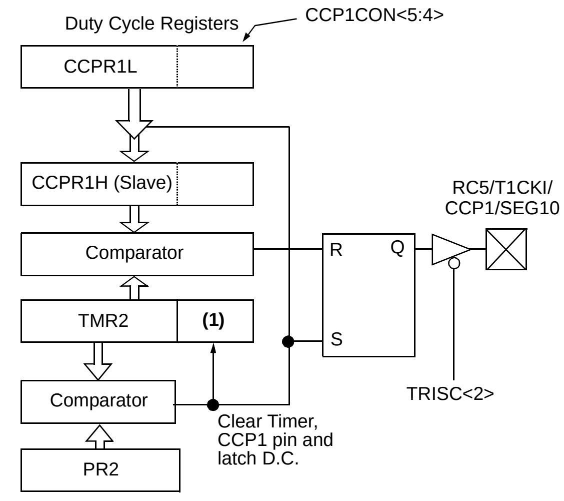
In Pulse Width Modulation mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the RC5/T1CKI/CCP1/SEG10 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the RC5/T1CKI/CCP1/SEG10 pin an output.

> **Note:** Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 15-5 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.3.3 "Setup for PWM Operation"**.

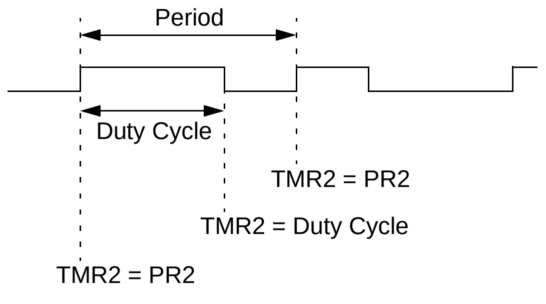
FIGURE 15-5: SIMPLIFIED PWM BLOCK DIAGRAM



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

A PWM output (Figure 15-6) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 15-6: PWM OUTPUT



15.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

PWM period = (PR2) + 1] • 4 • TOSC • (TMR2 prescale value)

PWM frequency is defined as 1 / [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The RC5/T1CKI/CCP1/SEG10 pin is set (exception: if PWM duty cycle = 0%, the RC5/T1CKI/CCP1/SEG10 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

> **Note:** The Timer2 postscaler (see **Section 7.0 "Timer2 Module"**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

15.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

PWM duty cycle =(CCPR1L:CCP1CON<5:4>) • TOSC • (TMR2 prescale value)

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitch-free PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock, or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

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

$$PWM\ Resolution = \frac{\log\left(\frac{F_{OSC}}{F_{PWM} \times TMR2\ Prescaler}\right)}{\log(2)} bits$$

> **Note:** If the PWM duty cycle value is longer than the PWM period, the RC5/T1CKI/CCP1/SEG10 pin will not be cleared.

15.3.3 SETUP FOR PWM OPERATION

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

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the RC5/T1CKI/CCP1/SEG10 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

TABLE 15-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 15-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1

Address	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
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	T1GINV	T1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.

TABLE 15-3: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Address	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
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module’s Register								0000 0000	0000 0000
92h	PR2	Timer2 Module’s Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as ‘0’. Shaded cells are not used by PWM and Timer2.

NOTES:

16.0 SPECIAL FEATURES OF THE CPU

The PIC16F91X has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator Selection
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming

The PIC16F91X has two timers that offer necessary delays on power-up. 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 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions-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-up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options (see Register 16-1).

16.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 16-1. These bits are mapped in program memory location 2007h.

Note: Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "PIC16F91X *Memory Programming Specification"* (DS41244) for more information.

REGISTER 16-1: CONFIG – CONFIGURATION WORD (ADDRESS: 2007h)

—	$\overline{\text{DEBUG}}$	FCMEN	IESO	BOREN1	BOREN0	$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0

bit 13 bit 0

bit 13 **Unimplemented:** Read as '1'

bit 12 **$\overline{\text{DEBUG}}$:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6/ICSPCLK/ICDCK/SEG14 and RB7/ICSPDAT/ICDDAT/SEG13 are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6/ICSPCLK/ICDCK/SEG14 and RB7/ICSPDAT/ICDDAT/SEG13 are dedicated to the debugger

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled
0 = Internal External Switchover mode is disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits[(1)]
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit (PCON<4>)
00 = BOR disabled

bit 7 **$\overline{\text{CPD}}$:** Data Code Protection bit[(2)]
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 6 **$\overline{\text{CP}}$:** Code Protection bit[(3)]
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** RB3/$\overline{\text{MCLR}}$/V_{PP} pin function select bit[(4)]
1 = RB3/$\overline{\text{MCLR}}$/V_{PP} pin function is $\overline{\text{MCLR}}$
0 = RB3/$\overline{\text{MCLR}}$/V_{PP} pin function is digital input, $\overline{\text{MCLR}}$ internally tied to V_{DD}

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKO function on RA6/OSC2/CLKO/T1OSO pin, RC on RA7/OSC1/CLKI/T1OSI
110 = RCIO oscillator: I/O function on RA6/OSC2/CLKO/T1OSO pin, RC on RA7/OSC1/CLKI/T1OSI
101 = INTOSC oscillator: CLKO function on RA6/OSC2/CLKO/T1OSO pin, I/O function on RA7/OSC1/CLKI/T1OSI
100 = INTOSCIO oscillator: I/O function on RA6/OSC2/CLKO/T1OSO pin, I/O function on RA7/OSC1/CLKI/T1OSI
011 = EC: I/O function on RA6/OSC2/CLKO/T1OSO pin, CLKI on RA7/OSC1/CLKI/T1OSI
010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKO/T1OSO and RA7/OSC1/CLKI/T1OSI
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKO/T1OSO and RA7/OSC1/CLKI/T1OSI
000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKO/T1OSO and RA7/OSC1/CLKI/T1OSI

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: The entire data EEPROM will be erased when the code protection is turned off.
3: The entire program memory will be erased when the code protection is turned off.
4: When $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

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

16.2 Reset

The PIC16F91X differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a “Reset state” on:

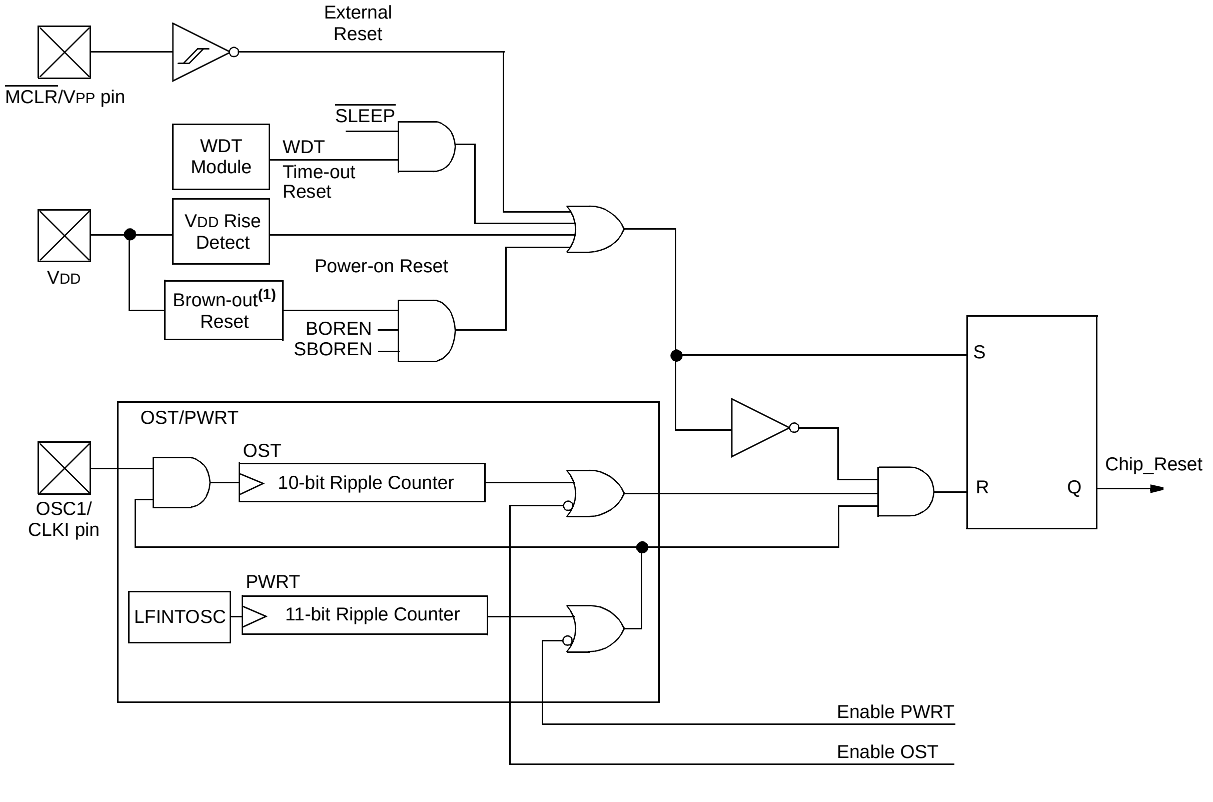
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

They are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 16-2. These bits are used in software to determine the nature of the Reset. See Table 16-5 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 16-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 19.0 “Electrical Specifications”** for pulse width specifications.

FIGURE 16-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: Refer to the Configuration Word register (Register 16-1).

16.3 Power-on Reset

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply connect the $\overline{\text{MCLR}}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 19.0 "Electrical Specifications"** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOR (see **Section 16.3.3 "Brown-Out Reset (BOR)"**).

Note:	The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 µs.

When the device starts normal operation (exits the Reset condition), device operating parameters
(i.e., voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

16.3.1 $\overline{\text{MCLR}}$

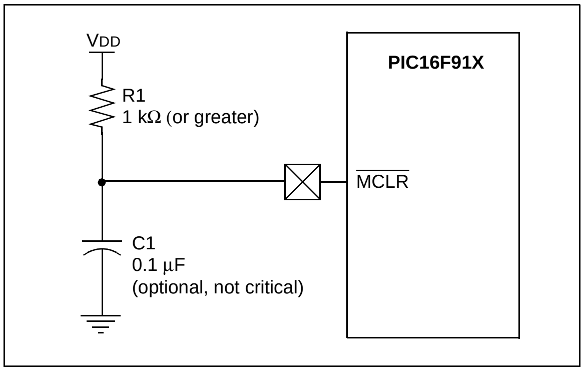
PIC16F91X has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{\text{MCLR}}$ pin low.

The behavior of the ESD protection on the $\overline{\text{MCLR}}$ pin has been altered from early devices of this family. Voltages applied to the pin that exceed its specification can result in both $\overline{\text{MCLR}}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{\text{MCLR}}$ pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 16-2, is suggested.

An internal $\overline{\text{MCLR}}$ option is enabled by clearing the MCLRE bit in the Configuration Word register. When cleared, $\overline{\text{MCLR}}$ is internally tied to VDD and an internal weak pull-up is enabled for the $\overline{\text{MCLR}}$ pin. In-Circuit Serial Programming is not affected by selecting the internal $\overline{\text{MCLR}}$ option.

FIGURE 16-2: RECOMMENDED $\overline{\text{MCLR}}$ CIRCUIT



16.3.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 3.4 "Internal Clock Modes"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, $\overline{\text{PWRTE}}$, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 19.0 "Electrical Specifications"**).

16.3.3 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register selects one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = `01`, the SBOREN bit (PCON<4>) enables/disables the BOR allowing it to be controlled in software. By selecting BOREN<1:0>, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 16-1 for the configuration word definition.

If VDD falls below VBOR for greater than parameter (TBOR) (see **Section 19.0 "Electrical Specifications"**), the Brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not insured to occur if VDD falls below VBOR for less than parameter (TBOR).

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 16-3). The Power-up Timer will now be invoked, if enabled and will keep the chip in Reset an additional 64 ms.

Note: The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word.

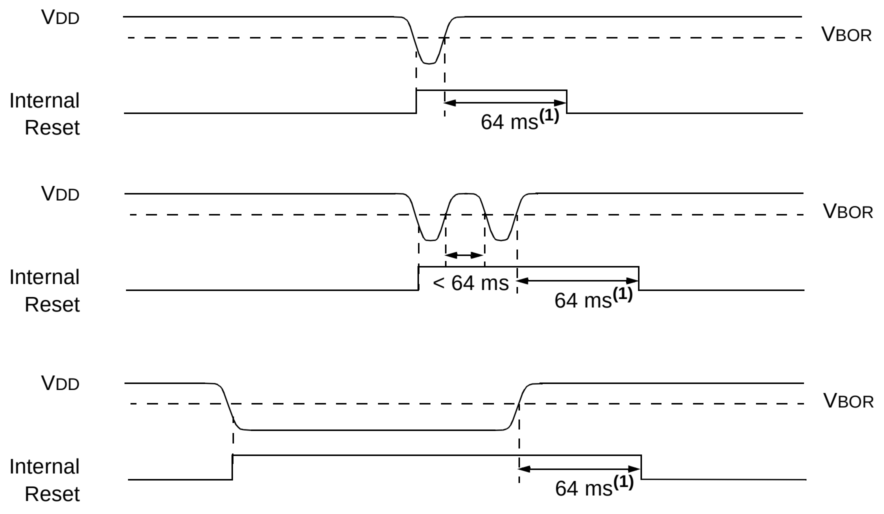
If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

16.3.4 BOR CALIBRATION

The PIC16F91X stores the BOR calibration values in fuses located in the Calibration Word (2008h). The Calibration Word is not erased when using the specified bulk erase sequence in the "PIC16F91X *Memory Programming Specification"* (DS41244) and thus, does not require reprogramming.

Address 2008h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "PIC16F91X *Memory Programming Specification"* (DS41244) for more information.

FIGURE 16-3: BROWN-OUT SITUATIONS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '`0`'.

16.3.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in EC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 16-4, Figure 16-5 and Figure 16-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active, by enabling Two-Speed Start-up or Fail-Safe Monitor (see **Section 3.6.2 "Two-Speed Start-up Sequence"** and **Section 3.7 "Fail-Safe Clock Monitor"**).

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 16-5). This is useful for testing purposes or to synchronize more than one PIC16F91X device operating in parallel.

Table 16-5 shows the Reset conditions for some special registers, while Table 16-5 shows the Reset conditions for all the registers.

16.3.6 POWER CONTROL (PCON) REGISTER

The Power Control (PCON) register (address 8Eh) has two Status bits to indicate what type of Reset that last occurred.

Bit 0 is $\overline{\text{BOR}}$ (Brown-out). $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOR}}$ = `0`, indicating that a Brown-out has occurred. The $\overline{\text{BOR}}$ Status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (BOREN<1:0> = `00` in the Configuration Word register).

Bit 1 is $\overline{\text{POR}}$ (Power-on Reset). It is a '`0`' on Power-on Reset and unaffected otherwise. The user must write a '`1`' to this bit following a Power-on Reset. On a subsequent Reset, if $\overline{\text{POR}}$ is '`0`', it will indicate that a Power-on Reset has occurred (i.e., VDD may have gone too low).

For more information, see **Section 16.3.3 "Brown-Out Reset (BOR)"**.

TABLE 16-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out Reset		Wake-up from Sleep
	$\overline{\text{PWRTE}}$ = 0	**$\overline{\text{PWRTE}}$ = 1**	**$\overline{\text{PWRTE}}$ = 0**	**$\overline{\text{PWRTE}}$ = 1**	
XT, HS, LP[(1)]	TPWRT + 1024 • TOSC	1024 • TOSC	TPWRT + 1024 • TOSC	1024 • TOSC	1024 • TOSC
RC, EC, INTOSC	TPWRT	—	TPWRT	—	—

Note 1: LP mode with T1OSC disabled.

TABLE 16-2: PCON BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	u	1	1	Power-on Reset
1	0	1	1	Brown-out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-up
u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep

Legend: u = unchanged, x = unknown

TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT

Address	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[(1)]
03h	STATUS	IRP	RP1	RPO	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`0001 1xxx`	`000q quuu`
8Eh	PCON	—	—	—	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	`--01 --qq`	`--0u --uu`

Legend: u = unchanged, x = unknown, — = unimplemented bit, reads as '`0`', q = value depends on condition. Shaded cells are not used by BOR.

Note 1: Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

FIGURE 16-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1

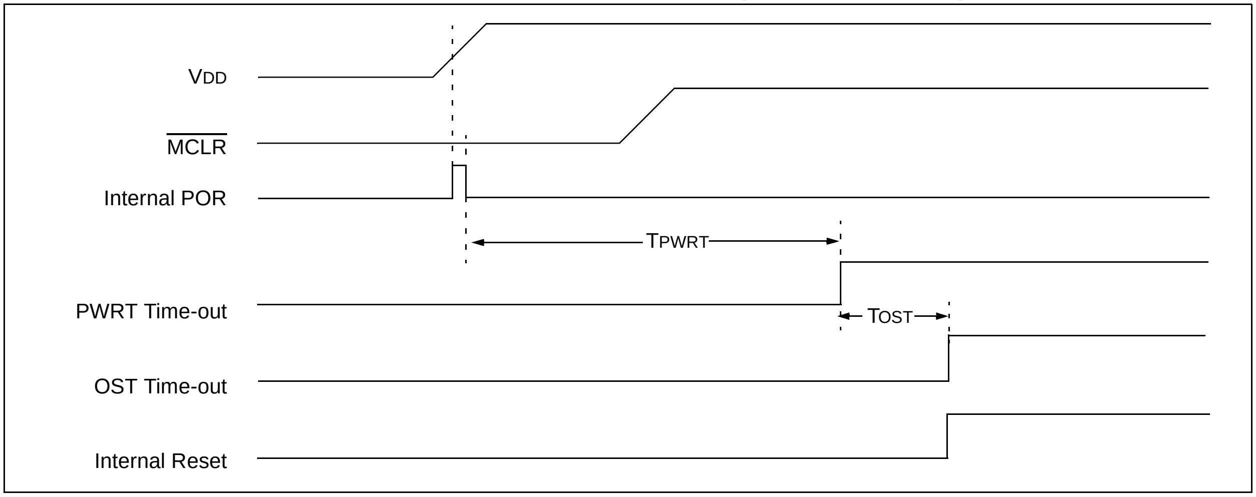


FIGURE 16-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2

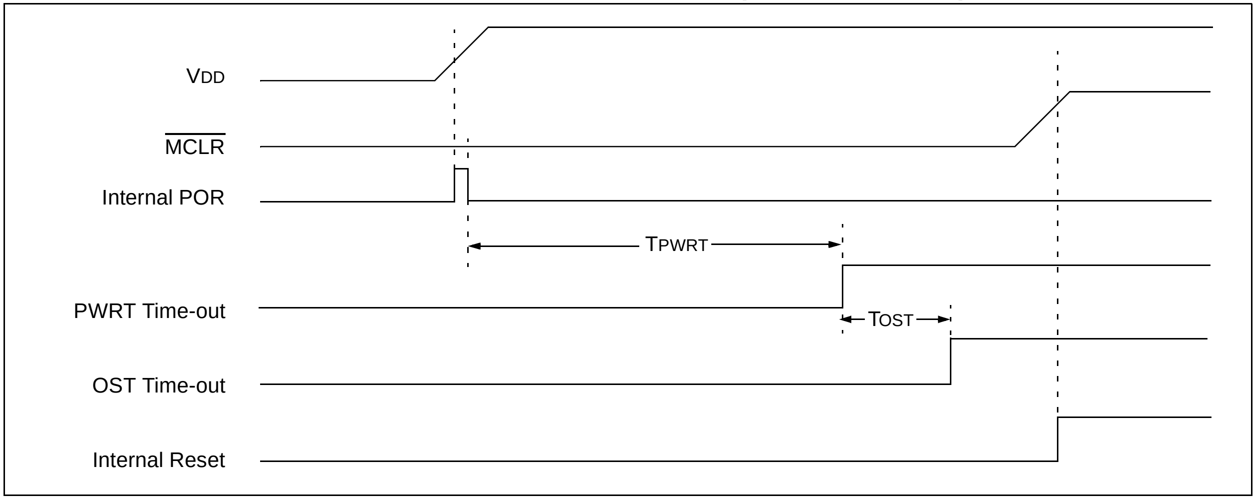


FIGURE 16-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD): CASE 3

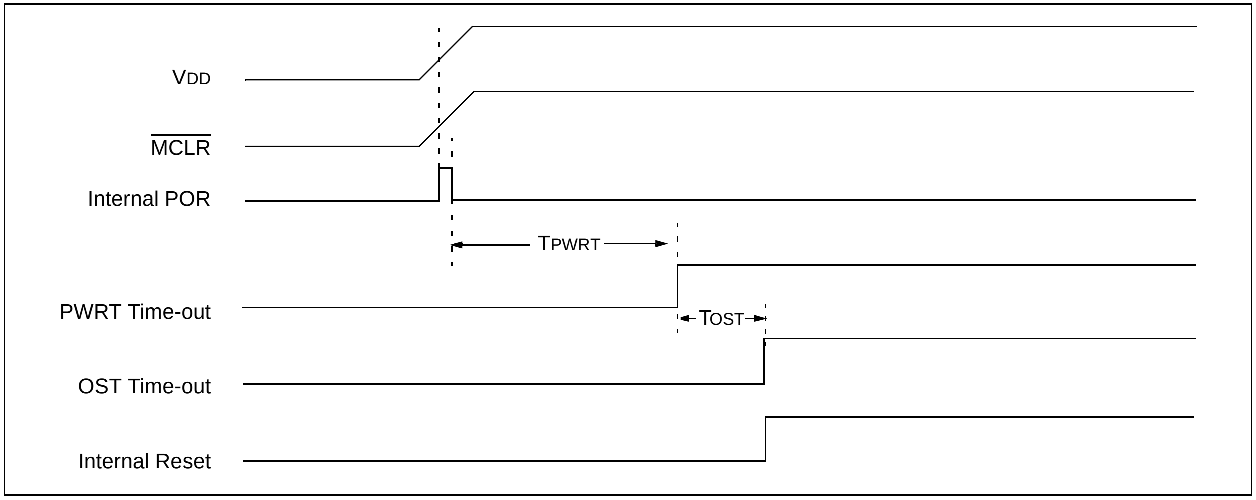


TABLE 16-4: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset	• $\overline{\text{MCLR}}$ Reset • WDT Reset • Brown-out Reset(1)	• Wake-up from Sleep through interrupt • Wake-up from Sleep through WDT time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/ 104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	xxxx xxxx	0000 0000	uuuu uuuu
PORTB	06h/106h	xxxx xxxx	0000 0000	uuuu uuuu
PORTC	07h	xxxx xxxx	0000 0000	uuuu uuuu
PORTD	08h	xxxx xxxx	0000 0000	uuuu uuuu
PORTE	09h	---- xxxx	---- 0000	---- uuuu
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 000x	0000 000x	uuuu uuuu(2)
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu(2)
PIR2	0Dh	0000 -0-0	0000 -0-0	uuuu -u-u
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	uuuu uuuu
TMR2	11h	01-0 0-00	01-0 0-00	uu-u u-uu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
SSPBUF	13h	xxxx xxxx	xxxx xxxx	uuuu uuuu
SSPCON	14h	0000 0000	0000 0000	uuuu uuuu
CCPR1L	15h	0000 0000	0000 0000	uuuu uuuu
CCPR1H	16h	0000 0010	0000 0010	uuuu uuuu
CCP1CON	17h	000x 000x	000x 000x	uuuu uuuu
RCSTA	18h	---0 1000	---0 1000	---u uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
CCP2CON	1Dh	--00 0000	--00 0000	--uu uuuu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’, q = value depends on condition.

Note 1: If V_{DD} goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 16-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 16-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

Register	Address	Power-on Reset	• $\overline{\text{MCLR}}$ Reset • WDT Reset • Brown-out Reset[1]	• Wake-up from Sleep through interrupt • Wake-up from Sleep through WDT time-out
ADCON0	1Fh	0000 0000	0000 0000	uuuu uuuu
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	1111 1111	1111 1111	uuuu uuuu
TRISB	86h/186h	1111 1111	1111 1111	uuuu uuuu
TRISC	87h	1111 1111	1111 1111	uuuu uuuu
TRISD	88h	1111 1111	1111 1111	uuuu uuuu
TRISE	89h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PIE2	8Dh	0000 0000	0000 0000	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uu[1,5]	--uu --uu
OSCCON	8Fh	-110 q000	-110 x000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
ANSEL	91h	1111 1111	1111 1111	uuuu uuuu
PR2	92h	1111 1111	1111 1111	1111 1111
SSPADD	93h	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	94h	0000 0000	0000 0000	uuuu uuuu
WPUB	95h	1111 1111	1111 1111	uuuu uuuu
IOCB	96h	0000 ----	0000 ----	uuuu ----
CMCON1	97h	---- --10	---- --10	---- --uu
TXSTA	98h	0000 -010	0000 -010	uuuu -uuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
CMCON0	9Ch	0000 0000	0000 0000	uuuu uuuu
VRCON	9Dh	0-0- 0000	0-0- 0000	u-u- uuuu
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	9Fh	-000 ----	-000 ----	-uuu ----
WDTCON	105h	---0 1000	---0 1000	---u uuuu
LCDCON	107h	0001 0011	0001 0011	uuuu uuuu
LCDPS	108h	0000 0000	0000 0000	uuuu uuuu
LVDCON	109h	--00 -100	--00 -100	--uu -uuu
EEDATL	10Ch	0000 0000	0000 0000	uuuu uuuu
EEADRL	10Dh	0000 0000	0000 0000	uuuu uuuu
EEDATH	10Eh	--00 0000	0000 0000	uuuu uuuu
EEADRH	10Fh	---0 0000	0000 0000	uuuu uuuu
LCDDATA0	110h	xxxx xxxx	uuuu uuuu	uuuu uuuu
LCDDATA1	111h	xxxx xxxx	uuuu uuuu	uuuu uuuu
LCDDATA2	112h	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’, q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 16-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 16-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

Register	Address	Power-on Reset	• $\overline{\text{MCLR}}$ Reset • WDT Reset • Brown-out Reset[1]	• Wake-up from Sleep through interrupt • Wake-up from Sleep through WDT time-out
LCDDATA3	113h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA4	114h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA5	115h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA6	116h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA7	117h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA8	118h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA9	119h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA10	11Ah	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDDATA11	11Bh	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
LCDSE0	11Ch	`0000 0000`	`uuuu uuuu`	`uuuu uuuu`
LCDSE1	11Dh	`0000 0000`	`uuuu uuuu`	`uuuu uuuu`
LCDSE2	11Eh	`0000 0000`	`uuuu uuuu`	`uuuu uuuu`
EECON1	18Ch	`x--- x000`	`u--- q000`	`u--- uuuu`

Legend: u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’, q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 16-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 16-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	Status Register	PCON Register
Power-on Reset	000h	`0001 1xxx`	`--01 --0x`
$\overline{\text{MCLR}}$ Reset during normal operation	000h	`000u uuuu`	`--0u --uu`
$\overline{\text{MCLR}}$ Reset during Sleep	000h	`0001 0uuu`	`--0u --uu`
WDT Reset	000h	`0000 uuuu`	`--0u --uu`
WDT Wake-up	PC + 1	`uuu0 0uuu`	`--uu --uu`
Brown-out Reset	000h	`0001 1uuu`	`--01 --10`
Interrupt Wake-up from Sleep	PC + 1[1]	`uuu1 0uuu`	`--uu --uu`

Legend: u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’.

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

16.4 Interrupts

The PIC16F91X has multiple sources of interrupt:

- External Interrupt RB0/INT/SEG0
- TMR0 Overflow Interrupt
- PORTB Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- LCD Interrupt
- PLVD Interrupt
- USART Receive and Transmit interrupts
- CCP1 and CCP2 Interrupts
- TMR2 Interrupt

The Interrupt Control (INTCON) register and Peripheral Interrupt Request 1 (PIR1) register record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>), enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- PORTB Change Interrupt
- TMR0 Overflow Interrupt

The peripheral interrupt flags are contained in the special registers, PIR1 and PIR2. The corresponding interrupt enable bit are contained in the special registers, PIE1 and PIE2.

The following interrupt flags are contained in the PIR1 register:

- EEPROM Data Write Interrupt
- A/D Interrupt
- USART Receive and Transmit Interrupts
- Timer1 Overflow Interrupt
- CCP1 Interrupt
- SSP Interrupt

The following interrupt flags are contained in the PIR2 register:

- Fail-Safe Clock Monitor Interrupt
- Comparator 1 and 2 Interrupts
- LCD Interrupt
- PLVD Interrupt
- CCP2 Interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 16-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

Note 1: Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again.

For additional information on Timer1, A/D or data EEPROM modules, refer to the respective peripheral section.

Note: The ANSEL (91h) and CMCON0 (9Ch) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘`0`’. Also, if a LCD output function is active on an external interrupt pin, that interrupt function will be disabled.

16.4.1 RB0/INT/SEG0 INTERRUPT

External interrupt on RB0/INT/SEG0 pin is edge-triggered; either rising if the INTEDG bit (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT/SEG0 pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The RB0/INT/SEG0 interrupt can wake-up the processor from Sleep if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up (0004h). See **Section 16.7 "Power-Down Mode (Sleep)"** for details on Sleep and Figure 16-10 for timing of wake-up from Sleep through RB0/INT/SEG0 interrupt.

16.4.2 TMR0 INTERRUPT

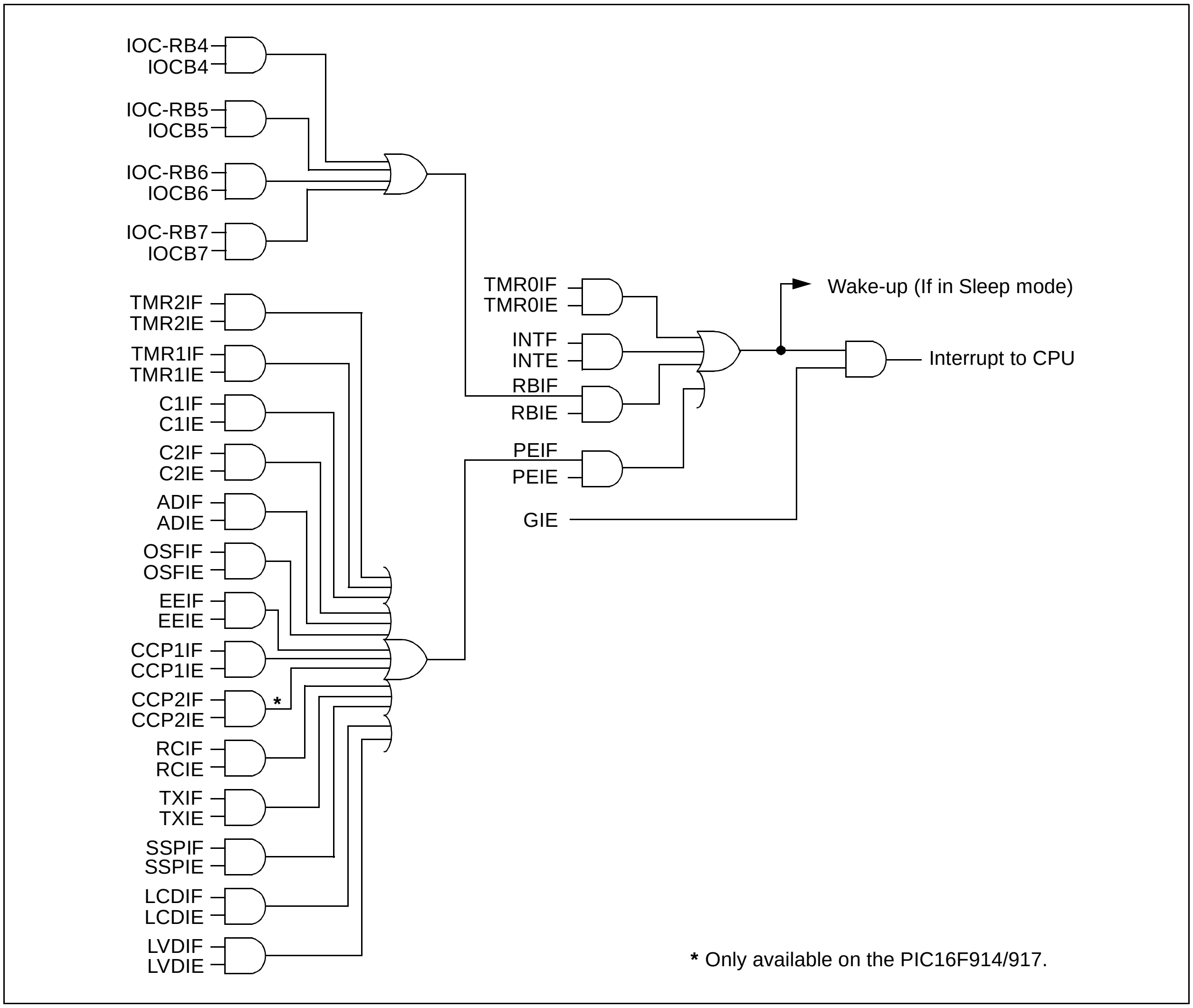
An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. See **Section 5.0 "Timer0 Module"** for operation of the Timer0 module.

16.4.3 PORTB INTERRUPT

An input change on PORTB change sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<3>) bit. Plus, individual pins can be configured through the IOCB register.

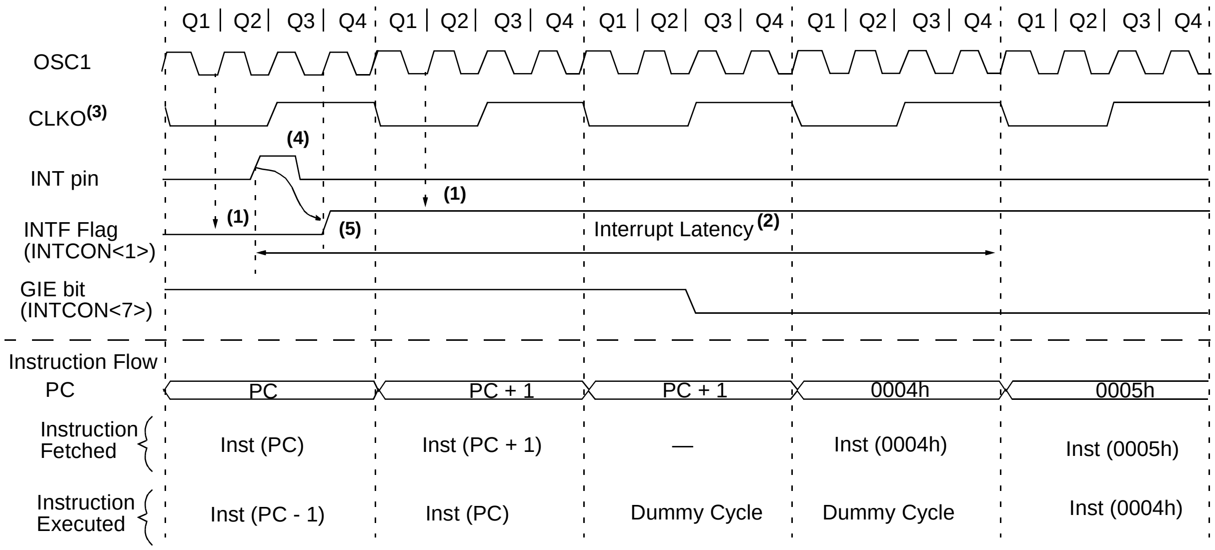
Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

FIGURE 16-7: INTERRUPT LOGIC



***** Only available on the PIC16F914/917.

FIGURE 16-8: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKO is available only in INTOSC and RC Oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 19.0 “Electrical Specifications”**.

5: INTF is enabled to be set any time during the Q4-Q1 cycles.

TABLE 16-6: SUMMARY OF INTERRUPT REGISTERS

Addr	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
0Bh, 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	OSFIF	C2IF	C1IF	LCDIF	—	LVDIF	—	CCP2IF	0000 -0-0	0000 -0-0
8Ch	PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	OSFIE	C2IE	C1IE	LCDIE	—	LVDIE	—	CCP2IE	0000 -0-0	0000 -0-0

Legend: x = unknown, u = unchanged, — = unimplemented read as ‘0’, q = value depends upon condition. Shaded cells are not used by the interrupt module.

16.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and Status registers). This must be implemented in software.

Since the lower 16 bytes of all banks are common in the PIC16F91X (see Figure 2-3), temporary holding registers, W_TEMP and STATUS_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. The same code shown in Example 16-1 can be used to:

- Store the W register
- Store the Status register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

> **Note:** The PIC16F91X normally does not require saving the PCLATH. However, if computed GOTO's are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

EXAMPLE 16-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP              ;Copy W to TEMP register
SWAPF   STATUS,W            ;Swap status to be saved into W
CLRF    STATUS              ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF   STATUS_TEMP         ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                      ;Insert user code here
:
SWAPF   STATUS_TEMP,W       ;Swap STATUS_TEMP register into W
                            ;(sets bank to original state)
MOVWF   STATUS              ;Move W into Status register
SWAPF   W_TEMP,F            ;Swap W_TEMP
SWAPF   W_TEMP,W            ;Swap W_TEMP into W
```

16.6 Watchdog Timer (WDT)

For PIC16F91X, the WDT has been modified from previous PIC16F devices. The new WDT is code and functionally compatible with previous PIC16F WDT modules and adds a 16-bit prescaler to the WDT. This allows the user to have a scaled value for the WDT and TMR0 at the same time. In addition, the WDT time-out value can be extended to 268 seconds. WDT is cleared under certain conditions described in Table 16-7.

16.6.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit does not reflect that the LFINTOSC is enabled.

The value of WDTCON is `'---0 1000'` on all Resets. This gives a nominal time base of 16 ms, which is compatible with the time base generated with previous PIC16F microcontroller versions.

Note:	When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

A new prescaler has been added to the path between the INTOSC and the multiplexers used to select the path for the WDT. This prescaler is 16 bits and can be programmed to divide the INTOSC by 32 to 65536, giving the WDT a nominal range of 1 ms to 268s.

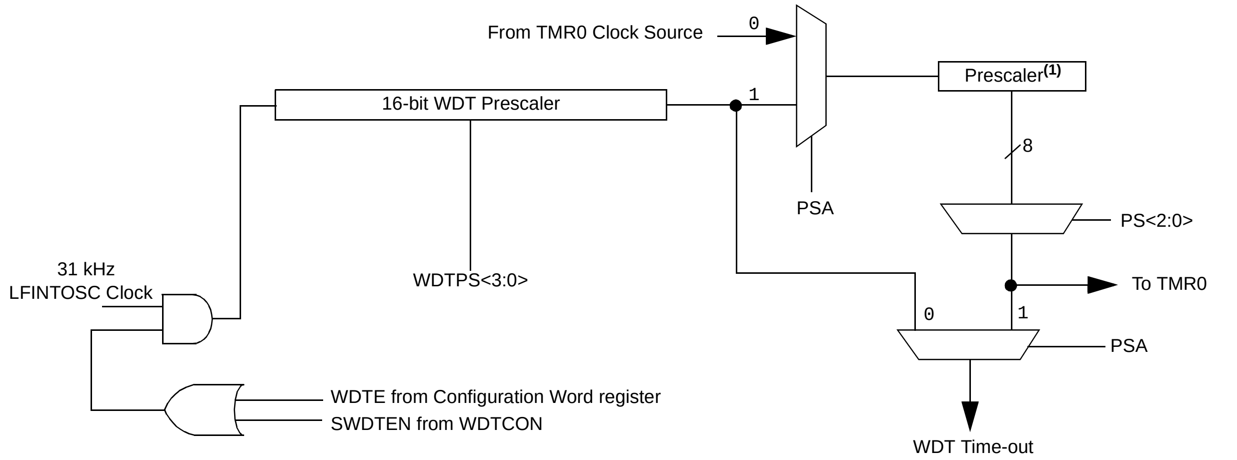
16.6.2 WDT CONTROL

The WDTE bit is located in the Configuration Word register. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word register is set, the SWDTEN bit (WDTCON<0>) has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits (OPTION_REG) have the same function as in previous versions of the PIC16F family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

FIGURE 16-9: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: This is the shared Timer0/WDT prescaler. See **Section 5.4 "Prescaler"** for more information.

TABLE 16-7: WDT STATUS

Conditions	WDT
WDTE = `0`	Cleared
`CLRWDT` Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST

REGISTER 16-2: WDTCON – WATCHDOG TIMER CONTROL REGISTER (ADDRESS: 105h)

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN

bit 7 bit 0

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 (Reset value)
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = reserved
1101 = reserved
1110 = reserved
1111 = reserved

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer bit[1]

1 = WDT is turned on
0 = WDT is turned off (Reset value)

Note 1: If WDTE configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

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

TABLE 16-8: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WSTPS1	WDTPS0	SWDTEN
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
2007h[1]	CONFIG	$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Register 16-1 for operation of all Configuration Word register bits.

16.7 Power-Down Mode (Sleep)

The Power-down mode is entered by executing a SLEEP instruction.

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- $\overline{PD}$ bit in the Status register is cleared.
- $\overline{TO}$ bit is set.
- Oscillator driver is turned off.
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at V_{DD} or V_{SS}, with no external circuitry drawing current from the I/O pin, and the comparators and CV_{REF} should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level.

> **Note:** It should be noted that a Reset generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

16.7.1 WAKE-UP FROM SLEEP

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

1. External Reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from RB0/INT/SEG0 pin, PORTB change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the Status register can be used to determine the cause of device Reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{TO}$ bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 Interrupt. Timer1 must be operating as an asynchronous counter.
2. EUSART Receive Interrupt
3. A/D conversion (when A/D clock source is RC)
4. EEPROM write operation completion
5. Comparator output changes state
6. Interrupt-on-change
7. External Interrupt from INT pin

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

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 GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction, then branches to the interrupt address (0004h). 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 interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The SLEEP instruction is completely executed.

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

16.7.2 WAKE-UP USING INTERRUPTS

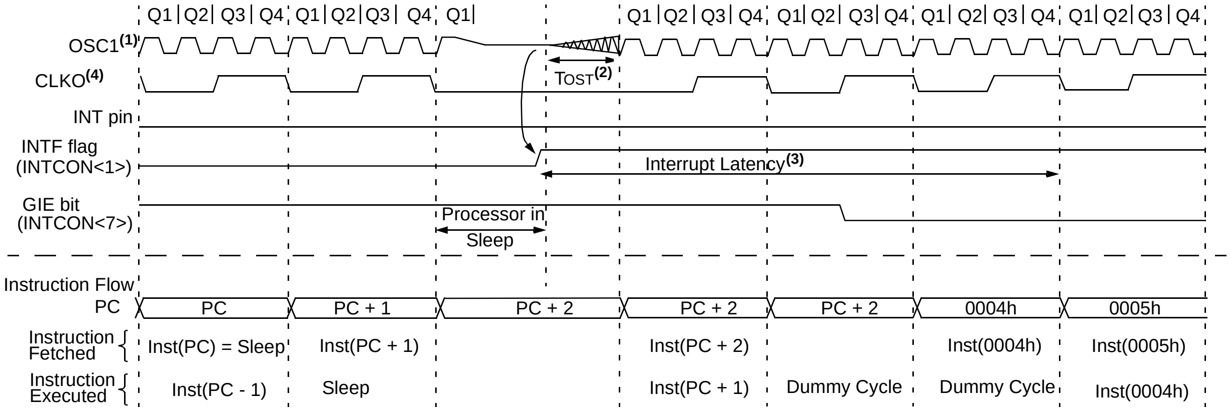
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

FIGURE 16-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS or LP Oscillator mode assumed.

2: TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.

3: GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.

4: CLKO is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

16.8 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out using ICSP for verification purposes.

Note:	The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. See the "PIC16F91X *Memory Programming Specification"* (DS41244) for more information.

16.9 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are used.

16.10 In-Circuit Serial Programming

The PIC16F91X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for:

- power
- ground
- programming voltage

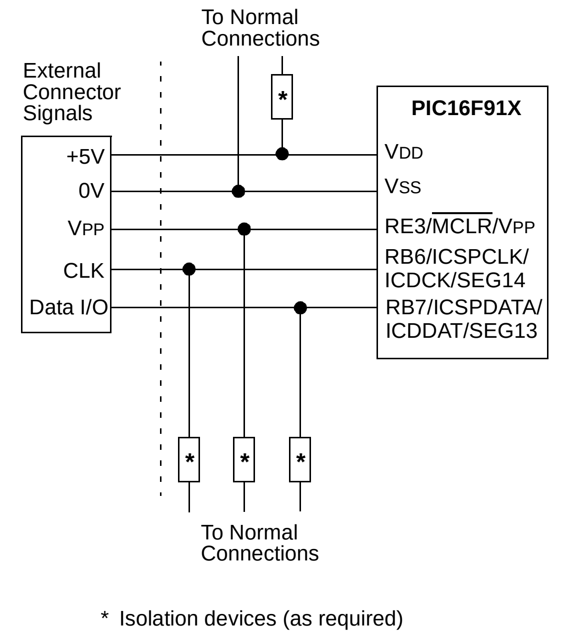
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB7/ICSPDAT/ICDDAT/SEG13 and RB1/SEG1 pins low, while raising the $\overline{\text{MCLR}}$ (V_{PP}) pin from V_{IL} to V_{IHH}. See "PIC16F91X *Memory Programming Specification"* (DS41244) for more information. RB7/ICSPDAT/ICDDAT/SEG13 becomes the programming data and RB6/ICSPCLK/ICDCK/SEG14 becomes the programming clock. Both RB7/ICSPDAT/ICDDAT/SEG13 and RB6/ICSPCLK/ICDCK/SEG14 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Program/Verify mode, the Program Counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a load or a read. For complete details of serial programming, please refer to the "PIC16F91X *Memory Programming Specification"* (DS41244).

A typical In-Circuit Serial Programming connection is shown in Figure 16-11.

FIGURE 16-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



16.11 In-Circuit Debugger

The PIC16F91X-ICD can be used in any of the package types. The device will be mounted on the target application board, which in turn has a 3 or 4 wire connection to the ICD tool.

When the debug bit in the Configuration Word (CONFIG<12>) is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. See Table 16-9 for more detail.

> **Note:** The user's application must have the circuitry required to support ICD functionality. Once the ICD circuitry is enabled, normal device pin functions on RB6/ICSPCLK/ICDCK/SEG14 and RB7/ICSPDAT/ICDDAT/SEG13 will not be usable. The ICD circuitry uses these pins for communication with the ICD2 external debugger.

For more information, see "Using *MPLAB® ICD 2"* (DS51265), available on Microchip's web site (www.microchip.com).

16.11.1 ICD PINOUT

The devices in the PIC16F91X family carry the circuitry for the In-Circuit Debugger on-chip and on existing device pins. This eliminates the need for a separate die or package for the ICD device. The pinout for the ICD device is the same as the devices (see **Section 1.0 "Device Overview"** for complete pinout and pin descriptions). Table 16-9 shows the location and function of the ICD related pins on the 28 and 40 pin devices.

TABLE 16-9: PIC16F91X-ICD PIN DESCRIPTIONS

Pin (PDIP)		Name	Type	Pull-up	Description
PIC16F914/917	PIC16F913/916				
40	28	ICDDATA	TTL	—	In Circuit Debugger Bidirectional data
39	27	ICDCLK	ST	—	In Circuit Debugger Bidirectional clock
1	1	$\overline{\text{MCLR}}$ / VPP	HV	—	Programming voltage
11,32	20	VDD	P	—	
12,31	8,19	VSS	P	—	

Legend: TTL = TTL input buffer, ST = Schmitt Trigger input buffer, P = Power, HV = High Voltage

17.0 INSTRUCTION SET SUMMARY

The PIC16F91X instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 17-1, while the various opcode fields are summarized in Table 17-1.

Table 17-2 lists the instructions recognized by the MPASM™ assembler. A complete description of each instruction is also available in the *"PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

Note:	To maintain upward compatibility with future products, do not use the OPTION and TRIS instructions.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

17.1 READ-MODIFY-WRITE OPERATIONS

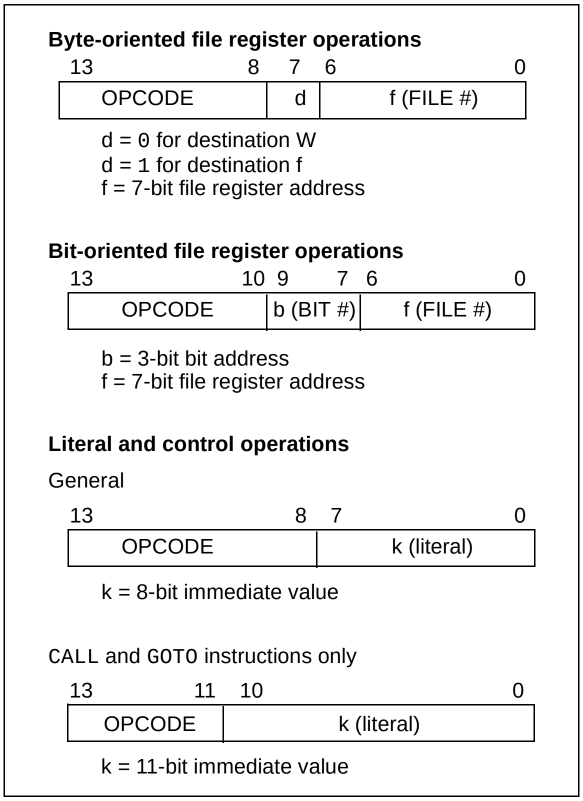
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF GPIO instruction will read GPIO, clear all the data bits, then write the result back to GPIO. This example would have the unintended result of clearing the condition that set the GPIF flag.

TABLE 17-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

FIGURE 17-1: GENERAL FORMAT FOR INSTRUCTIONS



Byte-oriented file register operations

d = `0` for destination W
d = `1` for destination f
f = 7-bit file register address

Bit-oriented file register operations

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

k = 11-bit immediate value

TABLE 17-2: PIC16F91X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode	Status Affected	Notes
				MSb LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
`ADDWF`	f, d	Add W and f	1	`00 0111 dfff ffff`	C,DC,Z	**1, 2**
`ANDWF`	f, d	AND W with f	1	`00 0101 dfff ffff`	Z	**1, 2**
`CLRF`	f	Clear f	1	`00 0001 lfff ffff`	Z	**2**
`CLRW`	-	Clear W	1	`00 0001 0xxx xxxx`	Z	
`COMF`	f, d	Complement f	1	`00 1001 dfff ffff`	Z	**1, 2**
`DECF`	f, d	Decrement f	1	`00 0011 dfff ffff`	Z	**1, 2**
`DECFSZ`	f, d	Decrement f, Skip if 0	1**(2)**	`00 1011 dfff ffff`		**1, 2, 3**
`INCF`	f, d	Increment f	1	`00 1010 dfff ffff`	Z	**1, 2**
`INCFSZ`	f, d	Increment f, Skip if 0	1**(2)**	`00 1111 dfff ffff`		**1, 2, 3**
`IORWF`	f, d	Inclusive OR W with f	1	`00 0100 dfff ffff`	Z	**1, 2**
`MOVF`	f, d	Move f	1	`00 1000 dfff ffff`	Z	**1, 2**
`MOVWF`	f	Move W to f	1	`00 0000 lfff ffff`		
`NOP`	-	No Operation	1	`00 0000 0xx0 0000`		
`RLF`	f, d	Rotate Left f through Carry	1	`00 1101 dfff ffff`	C	**1, 2**
`RRF`	f, d	Rotate Right f through Carry	1	`00 1100 dfff ffff`	C	**1, 2**
`SUBWF`	f, d	Subtract W from f	1	`00 0010 dfff ffff`	C,DC,Z	**1, 2**
`SWAPF`	f, d	Swap nibbles in f	1	`00 1110 dfff ffff`		**1, 2**
`XORWF`	f, d	Exclusive OR W with f	1	`00 0110 dfff ffff`	Z	**1, 2**
BIT-ORIENTED FILE REGISTER OPERATIONS						
`BCF`	f, b	Bit Clear f	1	`01 00bb bfff ffff`		**1, 2**
`BSF`	f, b	Bit Set f	1	`01 01bb bfff ffff`		**1, 2**
`BTFSC`	f, b	Bit Test f, Skip if Clear	1 **(2)**	`01 10bb bfff ffff`		**3**
`BTFSS`	f, b	Bit Test f, Skip if Set	1 **(2)**	`01 11bb bfff ffff`		**3**
LITERAL AND CONTROL OPERATIONS						
`ADDLW`	k	Add literal and W	1	`11 111x kkkk kkkk`	C,DC,Z	
`ANDLW`	k	AND literal with W	1	`11 1001 kkkk kkkk`	Z	
`CALL`	k	Call subroutine	2	`10 0kkk kkkk kkkk`		
`CLRWDT`	-	Clear Watchdog Timer	1	`00 0000 0110 0100`	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
`GOTO`	k	Go to address	2	`10 1kkk kkkk kkkk`		
`IORLW`	k	Inclusive OR literal with W	1	`11 1000 kkkk kkkk`	Z	
`MOVLW`	k	Move literal to W	1	`11 00xx kkkk kkkk`		
`RETFIE`	-	Return from interrupt	2	`00 0000 0000 1001`		
`RETLW`	k	Return with literal in W	2	`11 01xx kkkk kkkk`		
`RETURN`	-	Return from Subroutine	2	`00 0000 0000 1000`		
`SLEEP`	-	Go into Standby mode	1	`00 0000 0110 0011`	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
`SUBLW`	k	Subtract W from literal	1	`11 110x kkkk kkkk`	C,DC,Z	
`XORLW`	k	Exclusive OR literal with W	1	`11 1010 kkkk kkkk`	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF GPIO, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is ‘`1`’ for a pin configured as input and is driven low by an external device, the data will be written back with a ‘`0`’.

2: If this instruction is executed on the TMR0 register (and where applicable, d = `1`), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the *PICmicro® Mid-Range MCU Family Reference Manual”* (DS33023).

17.2 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND’ed with the eight-bit literal ‘k’. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is set.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed. If bit ‘b’, in register ‘f’, is ‘0’, the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \le f \le 127$ $0 \le b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is executed. If bit ‘b’ is ‘1’, then the next instruction is discarded and a NOP is executed instead, making this a two-cycle instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{\text{TO}}$ 1 → $\overline{\text{PD}}$
Status Affected:	$\overline{\text{TO}}$, $\overline{\text{PD}}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{\text{TO}}$ and $\overline{\text{PD}}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(\bar{f})$ → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are complemented. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are decremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. If the result is ‘1’, the next instruction is executed. If the result is ‘0’, then a NOP is executed instead, making it a two-cycle instruction.

GOTO	Go to Address
Syntax:	[*label*] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. If the result is ‘1’, the next instruction is executed. If the result is ‘0’, a NOP is executed instead, making it a two-cycle instruction.

IORLW	Inclusive OR Literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register ‘f’. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

MOVF — Move f

Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Encoding:	`00` | `1000` | `dfff` | `ffff`
Description:	The contents of register ‘f’ is moved to a destination dependent upon the status of ‘d’. If ‘d’ = `0`, destination is W register. If ‘d’ = `1`, the destination is file register ‘f’ itself. ‘d’ = `1` is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example	`MOVF FSR,` 0

After Instruction

W = value in FSR register

Z = 1

MOVLW — Move Literal to W

Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Encoding:	`11` | `00xx` | `kkkk` | `kkkk`
Description:	The eight bit literal ‘k’ is loaded into W register. The “don’t cares” will assemble as ‘`0`’s.
Words:	1
Cycles:	1
Example	`MOVLW 0x5A`

After Instruction

W = 0x5A

MOVWF — Move W to f

Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Encoding:	`00` | `0000` | `1fff` | `ffff`
Description:	Move data from W register to register ‘f’.
Words:	1
Cycles:	1
Example	`MOVWF OPTION`

Before Instruction

OPTION = 0xFF

W = 0x4F

After Instruction

OPTION = 0x4F

W = 0x4F

NOP — No Operation

Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	`00` | `0000` | `0xx0` | `0000`
Description:	No operation.
Words:	1
Cycles:	1
Example	`NOP`

RETFIE — Return from Interrupt

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

00	0000	0000	1001

Description: Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

RETLW — Return with Literal in W

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

11	01xx	kkkk	kkkk

Description: The W register is loaded with the eight bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
        CALL TABLE ;W contains table
                   ;offset value
        •          ;W now has table
TABLE   value
        •
        •
        ADDWF PC   ;W = offset
        RETLW k1   ;Begin table
        RETLW k2   ;
        •
        •
        •
        RETLW kn  ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN — Return from Subroutine

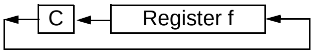
Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	00 1101 dfff 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 the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’. 
Words:	1
Cycles:	1

Example

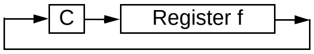
```
RLF     REG1,0
```

Before Instruction

```
        REG1    =    1110 0110
        C       =    0
```

After Instruction

```
        REG1    =    1110 0110
        W       =    1100 1100
        C       =    1
```

RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
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 the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. 

SLEEP	
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped.

SUBLW	Subtract W from Literal
Syntax:	[*label*] SUBLW k
Operands:	$0 \leq k \leq 255$
Operation:	k - (W) → (W)
Status Affected:	C, DC, Z
Description:	The W register is subtracted (2’s complement method) from the eight-bit literal ‘k’. The result is placed in the W register.

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - (W) → (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2’s complement method) W register from register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

SWAPF	**Swap Nibbles in f**
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f<3:0>) \rightarrow (destination<7:4>)$, $(f<7:4>) \rightarrow (destination<3:0>)$
Status Affected:	None
Description:	The upper and lower nibbles of register ‘f’ are exchanged. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed in register ‘f’.

XORLW	**Exclusive OR Literal with W**
Syntax:	[*label*] XORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

XORWF	**Exclusive OR W with f**
Syntax:	[*label*] XORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

18.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/
MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELOQ® Security ICs
  - PICDEM MSC
  - microID® RFID
  - CAN
  - PowerSmart® Battery Management
  - Analog

18.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® based application that contains:

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor with color coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files (assembly or C)
  - mixed assembly and C
  - machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increasing flexibility and power.

18.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro MCUs.

The MPASM assembler generates relocatable object files for the MPLINK object linker, Intel® standard hex files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects
- User defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

18.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

18.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB object librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

18.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, timekeeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

18.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

18.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

18.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

18.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

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

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

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

18.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

18.11 MPLAB ICD 2 In-Circuit Debugger

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

18.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

18.13 MPLAB PM3 Device Programmer

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

18.14 PICSTART Plus Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus development programmer supports most PICmicro devices up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

18.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

18.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/ Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *“TCP/IP Lean, Web Servers for Embedded Systems,”* by Jeremy Bentham

18.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

18.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

18.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User’s Guide.

18.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

18.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

18.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

18.23 PICkit™ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips ‘n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

18.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

18.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/ calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

NOTES:

19.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings[†]

Ambient temperature under bias	-40° to +125°C
Storage temperature	-65°C to +150°C
Voltage on V_{DD} with respect to V_{SS}	-0.3V to +6.5V
Voltage on $\overline{MCLR}$ with respect to Vss	-0.3V to +13.5V
Voltage on all other pins with respect to V_{SS}	-0.3V to (V_{DD} + 0.3V)
Total power dissipation[1]	800 mW
Maximum current out of V_{SS} pin	300 mA
Maximum current into V_{DD} pin	250 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD})	±20 mA
Output clamp current, I_{OK} (Vo < 0 or Vo >V_{DD})	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB and PORTE[2] (combined)	200 mA
Maximum current sourced by PORTA, PORTB and PORTE[2] (combined)	200 mA
Maximum current sourced PORTA and PORTC (combined)	200 mA
Maximum current sunk by PORTA and PORTC (combined)	200 mA

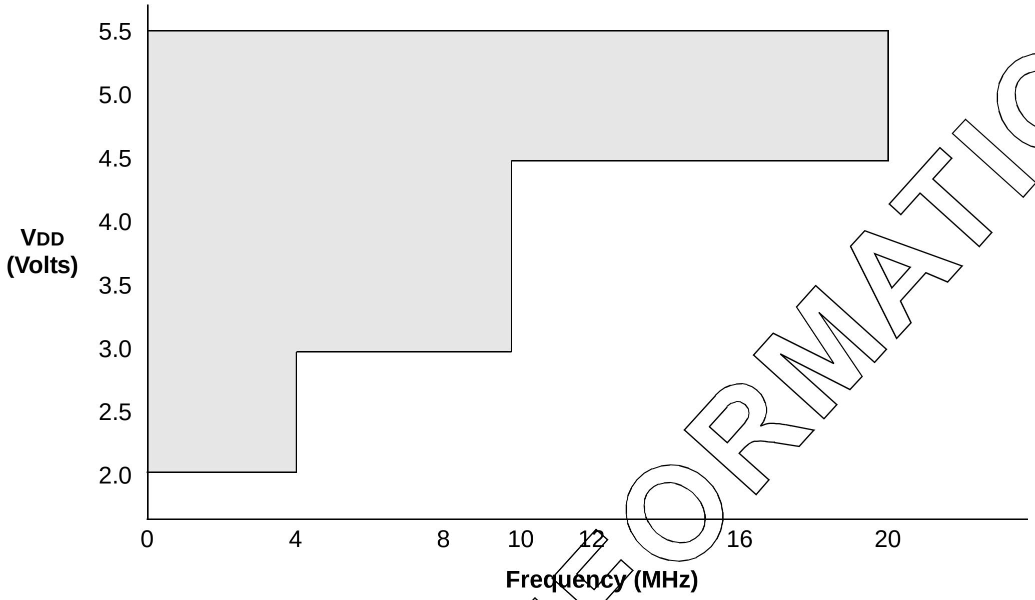
Note 1: Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$.

2: PORTD and PORTE are not implemented in PIC16F913/916 devices.

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Note: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 Ω should be used when applying a “low” level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to V_{SS}.

FIGURE 19-1: PIC16F91X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

19.1 DC Characteristics: PIC16F91X -I (Industrial) PIC16F91X -E (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VDD	**Supply Voltage**					FOSC < = 4 MHz:
D001			2.0	—	5.5	V	
D001C			3.0	—	5.5	V	FOSC < = 10 MHz
D001D			4.5	—	5.5	V	FOSC < = 20 MHz
D002	VDR	**RAM Data Retention Voltage**[(1)]	1.5*	—	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	—	VSS	—	V	See **Section 16.3 “Power-on Reset”** for details.
D004	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05*	—	—	V/ms	See **Section 16.3 “Power-on Reset”** for details.
D005	VBOR	**Brown-out Reset**	—	2.1	—	V	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

19.2 DC Characteristics: PIC16F91X-I (Industrial)

DC CHARACTERISTICS		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device Characteristics**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**	
						VDD	**Note**
D010	**Supply Current (IDD)[1, 2]**	—	12	TBD	μA	2.0	FOSC = 32 kHz LP Oscillator mode
		—	18	TBD	μA	3.0	
		—	35	TBD	μA	5.0	
D011		—	110	TBD	μA	2.0	FOSC = 1 MHz XT Oscillator mode
		—	190	TBD	μA	3.0	
		—	330	TBD	μA	5.0	
D012		—	220	TBD	μA	2.0	FOSC = 4 MHz XT Oscillator mode
		—	370	TBD	μA	3.0	
		—	0.6	TBD	mA	5.0	
D013		—	70	TBD	μA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	140	TBD	μA	3.0	
		—	260	TBD	μA	5.0	
D014		—	180	TBD	μA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	320	TBD	μA	3.0	
		—	500	TBD	μA	5.0	
D015		—	TBD	TBD	μA	2.0	FOSC = 31 kHz INTOSC mode
		—	TBD	TBD	μA	3.0	
		—	TBD	TBD	mA	5.0	
D016		—	340	TBD	μA	2.0	FOSC = 4 MHz INTOSC mode
		—	500	TBD	μA	3.0	
		—	0.8	TBD	mA	5.0	
D017		—	180	TBD	μA	2.0	FOSC = 4 MHz EXTRC mode
		—	320	TBD	μA	3.0	
		—	580	TBD	μA	5.0	
D018		—	2.1	TBD	mA	4.5	FOSC = 20 MHz HS Oscillator mode
		—	3.0	TBD	mA	5.0	
D020	**Power-down Base Current (IPD)[4]**	—	1.0	TBD	μA	2.0	WDT, BOR, Comparators, VREF and T1OSC disabled
		—	0.4	TBD	μA	3.0	
		—	2.9	TBD	μA	5.0	

Legend: TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

19.2 DC Characteristics: PIC16F91X-I (Industrial) (Continued)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D021		—	0.3	TBD	µA	2.0	WDT Current
		—	1.8	TBD	µA	3.0	
		—	8.4	TBD	µA	5.0	
D022		—	58	TBD	µA	3.0	BOR Current
		—	75	TBD	µA	5.0	
D023		—	3.3	TBD	µA	2.0	Comparator Current(3)
		—	6.1	TBD	µA	3.0	
		—	11.5	TBD	µA	5.0	
D024		—	58	TBD	µA	2.0	CVREF Current
		—	85	TBD	µA	3.0	
		—	138	TBD	µA	5.0	
D025		—	4.0	TBD	µA	2.0	T1OSC Current
		—	4.6	TBD	µA	3.0	
		—	6.0	TBD	µA	5.0	
D026		—	1.2	TBD	nA	3.0	A/D Current
		—	0.0022	TBD	µA	5.0	
D024		—	58	TBD	µA	2.0	CVREF Current
		—	85	TBD	µA	3.0	
		—	138	TBD	µA	5.0	
D025		—	4.0	TBD	µA	2.0	T1OSC Current
		—	4.6	TBD	µA	3.0	
		—	6.0	TBD	µA	5.0	
D026		—	1.2	TBD	nA	3.0	A/D Current
		—	0.0022	TBD	µA	5.0	

Legend: TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

19.3 DC Characteristics: PIC16F91X-E (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D010E	Supply Current (IDD)(1, 2)	—	9	TBD	µA	2.0	FOSC = 32 kHz LP Oscillator mode
		—	18	TBD	µA	3.0	
		—	35	TBD	µA	5.0	
D011E		—	110	TBD	µA	2.0	FOSC = 1 MHz XT Oscillator mode
		—	190	TBD	µA	3.0	
		—	330	TBD	µA	5.0	
D012E		—	220	TBD	µA	2.0	FOSC = 4 MHz XT Oscillator mode
		—	370	TBD	µA	3.0	
		—	0.6	TBD	mA	5.0	
D013E		—	70	TBD	µA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	140	TBD	µA	3.0	
		—	260	TBD	µA	5.0	
D014E		—	180	TBD	µA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D015E		—	TBD	TBD	µA	2.0	FOSC = 31 kHz INTOSC mode
		—	TBD	TBD	µA	3.0	
		—	TBD	TBD	mA	5.0	
D016E		—	340	TBD	µA	2.0	FOSC = 4 MHz INTOSC mode
		—	500	TBD	µA	3.0	
		—	0.8	TBD	mA	5.0	
D017E		—	180	TBD	µA	2.0	FOSC = 4 MHz EXTRC mode
		—	320	TBD	µA	3.0	
		—	580	TBD	µA	5.0	
D018E		—	2.1	TBD	mA	4.5	FOSC = 20 MHz HS Oscillator mode
		—	2.4	TBD	mA	5.0	

Legend: TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

19.3 DC Characteristics: PIC16F91X-E (Extended) (Continued)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020E	**Power-down Base Current (IPD)[4]**	—	0.00099	TBD	µA	2.0	WDT, BOR, Comparators, VREF and T1OSC disabled
		—	0.0012	TBD	µA	3.0	
		—	0.0029	TBD	µA	5.0	
D021E		—	0.3	TBD	µA	2.0	WDT Current
		—	1.8	TBD	µA	3.0	
		—	8.4	TBD	µA	5.0	
D022E		—	58	TBD	µA	3.0	BOR Current
		—	109	TBD	µA	5.0	
D023E		—	3.3	TBD	µA	2.0	Comparator Current[3]
		—	6.1	TBD	µA	3.0	
		—	11.5	TBD	µA	5.0	
D024E		—	58	TBD	µA	2.0	CVREF Current
		—	85	TBD	µA	3.0	
		—	138	TBD	µA	5.0	
D025E		—	4.0	TBD	µA	2.0	T1OSC Current
		—	4.6	TBD	µA	3.0	
		—	6.0	TBD	µA	5.0	
D026E		—	0.0012	TBD	µA	3.0	A/D Current[3]
		—	0.0022	TBD	µA	5.0	

Legend: TBD = To Be Determined

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

4: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

19.4 DC Characteristics: PIC16F91X-I (Industrial), PIC16F91X-E (Extended)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O port:					
D030		with TTL buffer	VSS	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			VSS	—	0.15 VDD	V	Otherwise
D031		with Schmitt Trigger buffer	VSS	—	0.2 VDD	V	Entire range
D032		$\overline{\text{MCLR}}$, OSC1 (RC mode)	VSS	—	0.2 VDD	V	
D033		OSC1 (XT and LP modes)(1)	VSS	—	0.3	V	
D033A		OSC1 (HS mode)(1)	VSS	—	0.3 VDD	V	
D034		I^{2}C™ mode	VSS	—	0.3VDD	V	Entire VDD Range
	VIH	**Input High Voltage**					
		I/O port:		—			
D040		with TTL buffer	2.0	—	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			(0.25 VDD + 0.8)	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.8 VDD	—	VDD	V	Entire range
D042		$\overline{\text{MCLR}}$	0.8 VDD	—	VDD	V	
D043		OSC1 (XT and LP modes)	1.6	—	VDD	V	**(Note 1)**
D043A		OSC1 (HS mode)	0.7 VDD	—	VDD	V	**(Note 1)**
D043B		OSC1 (RC mode)	0.9 VDD	—	VDD	V	
D044		I^{2}C mode	0.7VDD	—	VDD	V	Entire VDD Range
D070	IPUR	PORTB Weak Pull-up Current	50*	250	400*	μA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current(2)**					
D060		I/O port	—	± 0.1	± 1	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
D061		$\overline{\text{MCLR}}$(3)	—	± 0.1	± 5	μA	VSS ≤ VPIN ≤ VDD
D063		OSC1	—	± 0.1	± 5	μA	VSS ≤ VPIN ≤ VDD, XT, HS and LP OSC configuration
	VOL	**Output Low Voltage**					
D080		I/O port	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V (Ind.)
D083		OSC2/CLKO (RC mode)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V (Ind.) IOL = 1.2 mA, VDD = 4.5V (Ext.)
	VOH	**Output High Voltage**					
D090		I/O port	VDD – 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V (Ind.)
D092		OSC2/CLKO (RC mode)	VDD – 0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V (Ind.) IOH = -1.0 mA, VDD = 4.5V (Ext.)

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

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

3: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

19.4 DC Characteristics: PIC16F91X-I (Industrial), PIC16F91X-E (Extended) (Continued)

DC CHARACTERISTICS	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin	—	—	15*	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101	CIO	All I/O pins	—	—	50*	pF	
		Data EEPROM Memory					
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D120A	ED	Byte Endurance	10K	100K	—	E/W	+85°C ≤ TA ≤ +125°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON1 to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	5	6	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(2)	1M	10M	—	E/W	-40°C ≤ TA ≤ +85°C
		Program Flash Memory					
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	ED	Cell Endurance	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VPEW	VDD for Erase/Write	4.5	—	5.5	V	
D133	TPEW	Erase/Write cycle time	—	2	2.5	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

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

3: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

19.5 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

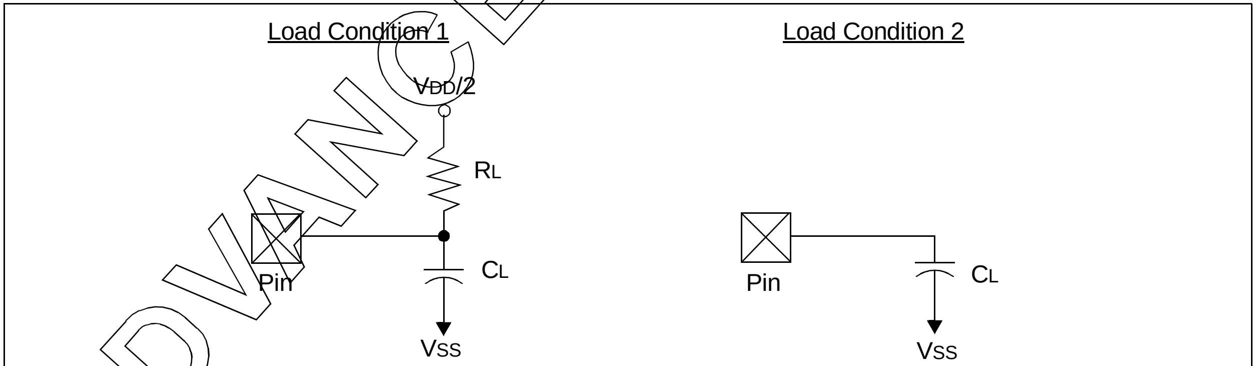
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKO	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 19-2: LOAD CONDITIONS



Legend: RL = 464Ω
CL = 50 pF for all pins
15 pF for OSC2 output

19.6 AC Characteristics: PIC16F91X (Industrial, Extended)

FIGURE 19-3: EXTERNAL CLOCK TIMING

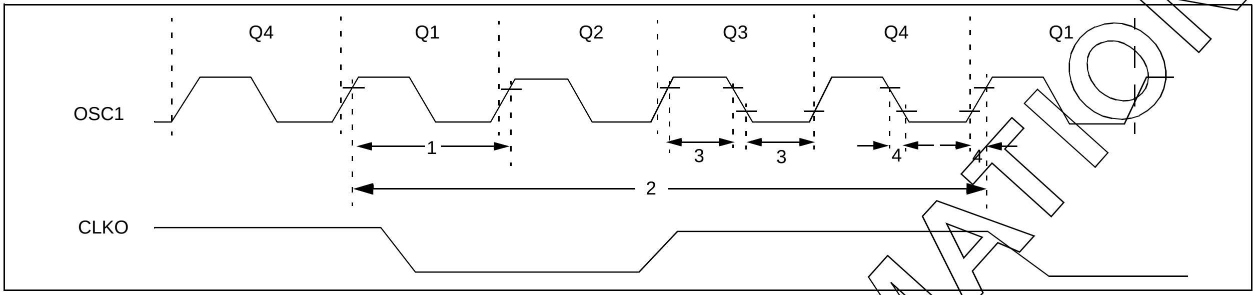


TABLE 19-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	External CLKI Frequency(1)	DC	—	37	kHz	LP Oscillator mode
			DC	—	4	MHz	XT Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	20	MHz	EC Oscillator mode
		Oscillator Frequency(1)	5	—	37	kHz	LP Oscillator mode
			—	4	—	MHz	INTOSC mode
			DC	—	4	MHz	RC Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	20	MHz	HS Oscillator mode
1	TOSC	External CLKI Period(1)	27	—	∞	μs	LP Oscillator mode
			50	—	∞	ns	HS Oscillator mode
			50	—	∞	ns	EC Oscillator mode
			250	—	∞	ns	XT Oscillator mode
		Oscillator Period(1)	27		200	μs	LP Oscillator mode
			—	250	—	ns	INTOSC mode
			250	—	—	ns	RC Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode
2	TCY	Instruction Cycle Time(1)	200	TCY	DC	ns	TCY = 4/FOSC
3	TosL, TosH	External CLKI (OSC1) High External CLKI Low	2*	—	—	μs	LP oscillator, TOSC L/H duty cycle
			20*	—	—	ns	HS oscillator, TOSC L/H duty cycle
			100 *	—	—	ns	XT oscillator, TOSC L/H duty cycle
4	TosR, TosF	External CLKI Rise External CLKI Fall	—	—	50*	ns	LP oscillator
			—	—	25*	ns	XT oscillator
			—	—	15*	ns	HS oscillator

* These parameters are characterized but not tested.

† 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at 'min' values with an external clock applied to OSC1 pin. When an external clock input is used, the 'max' cycle time limit is 'DC' (no clock) for all devices.

TABLE 19-2: PRECISION INTERNAL OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Freq. Tolerance	Min	Typ†	Max	Units	Conditions
F10	FOSC	Internal Calibrated INTOSC Frequency[1]	±1% ±2% ±5%	— — —	8.00 8.00 8.00	TBD TBD TBD	MHz MHz MHz	VDD and Temperature TBD 2.5V ≤ VDD ≤ 5.5V 0°C ≤ TA ≤ +85°C 2.0V ≤ VDD ≤ 5.5V -40°C ≤ TA ≤ +85°C (Ind.) -40°C ≤ TA ≤ +125°C (Ext.)
F14	TIOSCST	Oscillator Wake-up from Sleep Start-up Time*	— — —	— — —	TBD TBD TBD	TBD TBD TBD	μs μs μs	VDD = 2.0V, -40°C to +85°C VDD = 3.0V, -40°C to +85°C VDD = 5.0V, -40°C to +85°C

Legend: TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 uF and 0.01 uF values in parallel are recommended.

FIGURE 19-4: CLKO AND I/O TIMING

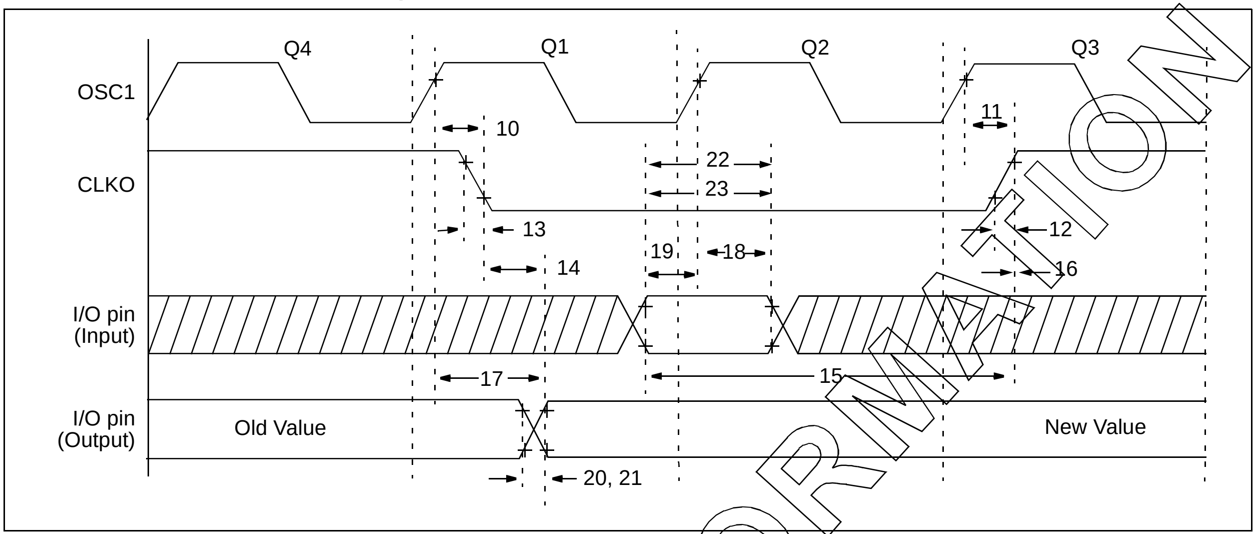


TABLE 19-3: CLKO AND I/O TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLOUT↓		—	75	200	ns	**(Note 1)**
11*	TosH2ckH	OSC1↑ to CLOUT↑		—	75	200	ns	**(Note 1)**
12*	TckR	CLKO Rise Time		—	35	100	ns	**(Note 1)**
13*	TckF	CLKO Fall Time		—	35	100	ns	**(Note 1)**
14*	TckL2ioV	CLKO↓ to Port Out Valid		—	—	0.5 TCY + 20	ns	**(Note 1)**
15*	TioV2ckH	Port In Valid before CLKO↑		TOSC + 200 ns	—	—	ns	**(Note 1)**
16*	TckH2ioI	Port In Hold after CLKO↑		0	—	—	ns	**(Note 1)**
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port Out Valid		—	50	150*	ns	
				—	—	300	ns	
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	3.0-5.5V	100	—	—	ns	
			2.0-5.5V	200	—	—	ns	
19*	TioV2osH	Port Input Valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20*	TioR	Port Output Rise Time	3.0-5.5V	—	10	40	ns	
			2.0-5.5V	—	—	145		
21*	TioF	Port Output Fall Time	3.0-5.5V	—	10	40	ns	
			2.0-5.5V	—	—	145		
22*	Tinp	INT Pin High or Low Time		25	—	—	ns	
23*	Trbp	PORTA change INT high or low time		TCY	—	—	ns	

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in RC mode where CLKO output is 4 x TOSC.

FIGURE 19-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

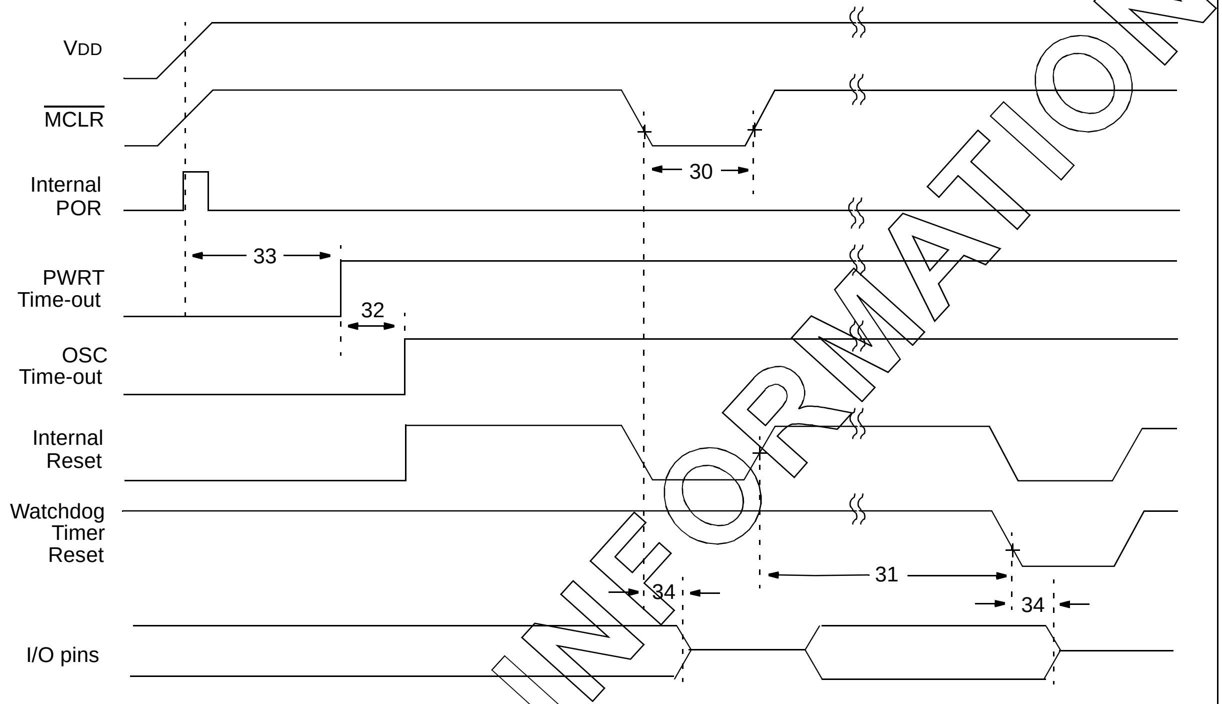
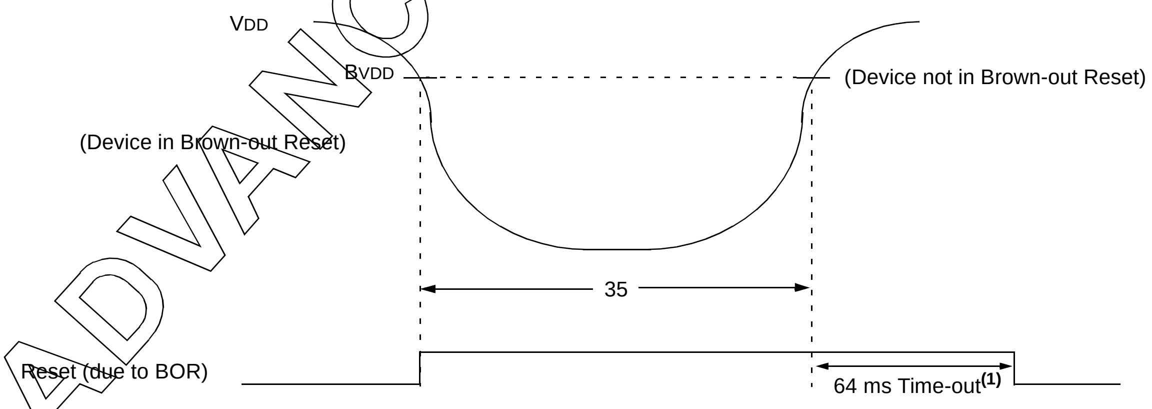


FIGURE 19-6: BROWN-OUT RESET TIMING AND CHARACTERISTICS



Note 1: 64 ms delay only if $\overline{\text{PWRTE}}$ bit in the Configuration Word is programmed to '0'.

TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	$\overline{\text{MCLR}}$ Pulse Width (low)	2 11	— 18	— 24	µs ms	VDD = 5V, -40°C to +85°C Extended temperature
31	TWDT	Watchdog Timer Time-out Period (No Prescaler)	10 10	17 17	25 30	ms ms	VDD = 5V, -40°C to +85°C Extended temperature
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	28* TBD	64 TBD	132* TBD	ms ms	VDD = 5V, -40°C to +85°C Extended Temperature
34	TIOZ	I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	2.0	µs	
	BVDD	Brown-out Reset Voltage	2.025	—	2.175	V	
35	TBOR	Brown-out Reset Pulse Width	100*	—	—	µs	VDD ≤ BVDD (D005)

Legend: TBD = To Be Determined

* These parameters are characterized but not tested.

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

FIGURE 19-7: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

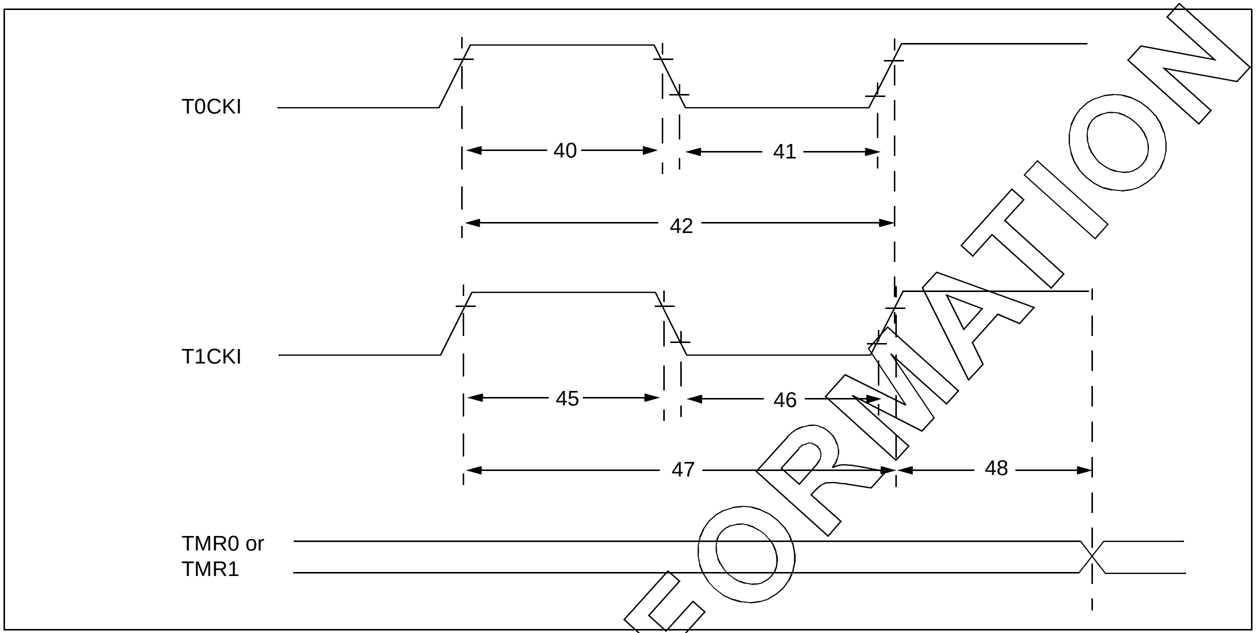


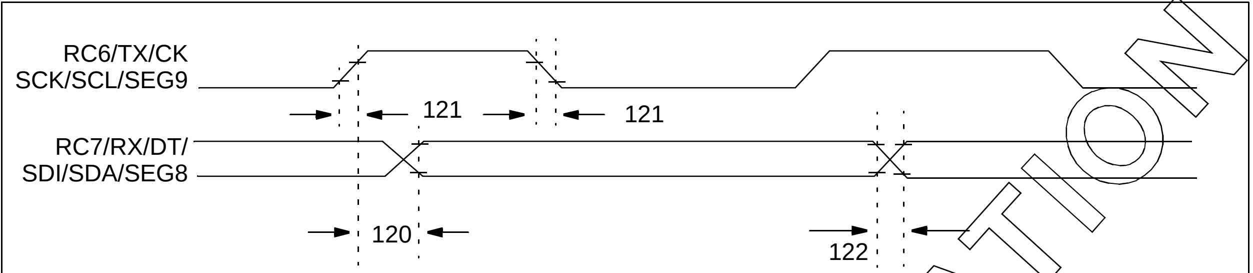
TABLE 19-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic			Min	Typ†	Max	Units	Conditions
40*	TT0H	T0CKI High Pulse Width		No Prescaler	0.5 TCY + 20	—	—	ns	
				With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width		No Prescaler	0.5 TCY + 20	—	—	ns	
				With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period			Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler		0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	3.0-5.5V	15	—	—	ns	
				2.0-5.5V	25	—	—	ns	
			Asynchronous	3.0-5.5V	30	—	—	ns	
				2.0-5.5V	50	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler		0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	3.0-5.5V	15	—	—	ns	
				2.0-5.5V	25	—	—	ns	
			Asynchronous	3.0-5.5V	30	—	—	ns	
				2.0-5.5V	50	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	3.0-5.5V	Greater of: 30 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				2.0-5.5V	50 or $\frac{T_{CY} + 40}{N}$	—	—	ns	
			Asynchronous	3.0-5.5V	60	—	—	ns	
				2.0-5.5V	100	—	—	ns	
	FT1	Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)			DC	—	200*	kHz	
48	TCKEZTMR1	Delay from external clock edge to timer increment			2 TOSC*	—	7 TOSC*	—	

* These parameters are characterized but not tested.

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

FIGURE 19-8: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



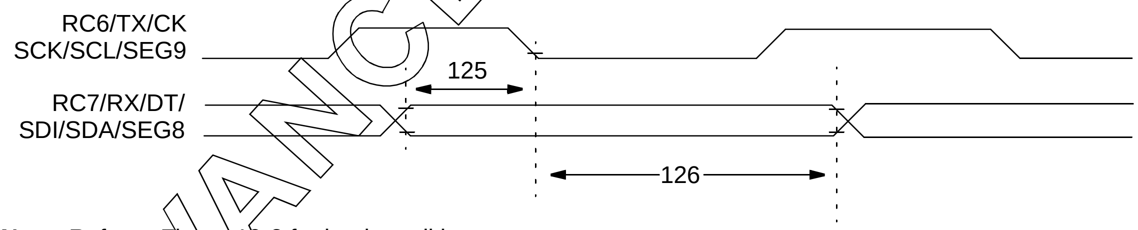
Note: Refer to Figure 19-2 for load conditions.

TABLE 19-6: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	3.0-5.5V	—	80	ns	
			2.0-5.5V	—	100	ns	
121	TCKRF	Clock out rise time and fall time (Master mode)	3.0-5.5V	—	45	ns	
			2.0-5.5V	—	50	ns	
122	TDTRF	Data-out rise time and fall time	3.0-5.5V	—	45	ns	
			2.0-5.5V	—	50	ns	

FIGURE 19-9: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



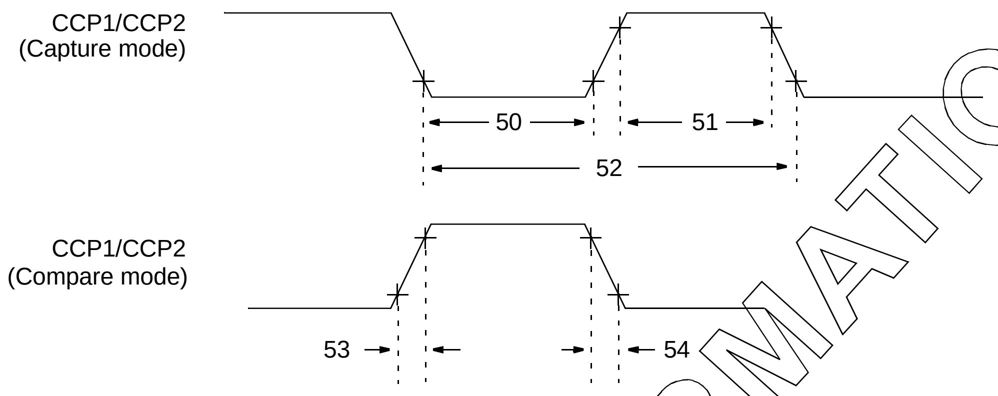
Note: Refer to Figure 19-2 for load conditions.

TABLE 19-7: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (Master and Slave) Data-hold before CK ↓ (DT hold time)	10	—	ns	
126	TCKL2DTL	Data-hold after CK ↓ (DT hold time)	15	—	ns	

FIGURE 19-10: CAPTURE/COMPARE/PWM TIMINGS



Note: Refer to Figure 19-2 for load conditions.

TABLE 19-8: CAPTURE/COMPARE/PWM REQUIREMENTS

Param. No.	Sym	Characteristic			Min	Typ†	Max	Units	Conditions
50*	TCCL	CCP1 input low time	No Prescaler		0.5TCY + 5	—	—	ns	
			With Prescaler	3.0-5.5V	10	—	—	ns	
				2.0-5.5V	20	—	—	ns	
51*	TCCH	CCP1 input high time	No Prescaler		0.5TCY + 5	—	—	ns	
			With Prescaler	3.0-5.5V	10	—	—	ns	
				2.0-5.5V	20	—	—	ns	
52*	TCCP	CCP1 input period			$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TCCR	CCP1 output fall time		3.0-5.5V	—	10	25	ns	
				2.0-5.5V	—	25	50	ns	
54*	TCCF	CCP1 output fall time		3.0-5.5V	—	10	25	ns	
				2.0-5.5V	—	25	45	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 19-9: COMPARATOR SPECIFICATIONS

Comparator Specifications		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +125°C				
Sym	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
V_{OS}	Input Offset Voltage	—	±5.0	±10	mV	
V_{CM}	Input Common Mode Voltage	0	—	V_{DD} – 1.5	V	
C_{MRR}	Common Mode Rejection Ratio	+55*	—	—	db	
T_{RT}	Response Time(1)	—	150	400*	ns	
T_{MC2COV}	Comparator Mode Change to Output Valid	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (V_{DD} – 1.5)/2 while the other input transitions from V_{SS} to V_{DD} – 1.5V.

TABLE 19-10: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

Voltage Reference Specifications		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +125°C				
Sym.	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	Resolution	— —	V_{DD}/24* V_{DD}/32	— —	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
	Absolute Accuracy	— —	— —	±1/4* ±1/2*	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
	Unit Resistor Value (R)	—	2K*	—	Ω	
	Settling Time(1)	—	—	10*	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while VRR = 1 and VR<3:0> transitions from '0000' to '1111'.

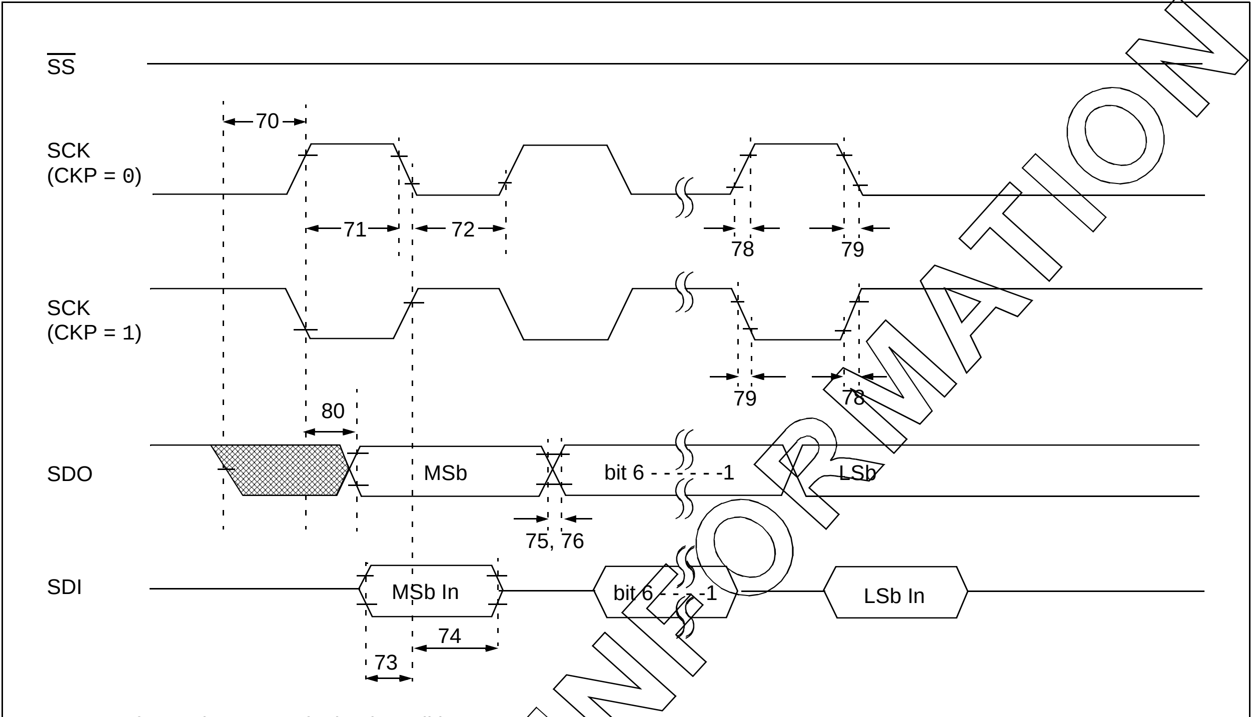
TABLE 19-11: PIC16F91X PLVD CHARACTERISTICS:

DC Characteristics			Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ T_A ≤ +125°C Operating Voltage V_{DD} Range 2.0V-5.5V				
Sym.	**Characteristic**		**Min**	**Typ†**	**Max**	**Units**	**Conditions**
V_{PLVD}	PLVD Voltage	LVDL<2:0> = 000	TBD	1.9	TBD	V	
		TBD	TBD	2.0	TBD	V	
		TBD	TBD	2.1	TBD	V	
		TBD	TBD	2.2	TBD	V	
		TBD	TBD	2.3	TBD	V	
		TBD	TBD	4.0	TBD	V	
		TBD	TBD	4.2	TBD	V	
		TBD	TBD	4.5	TBD	V	

Legend: TBD = To Be Determined

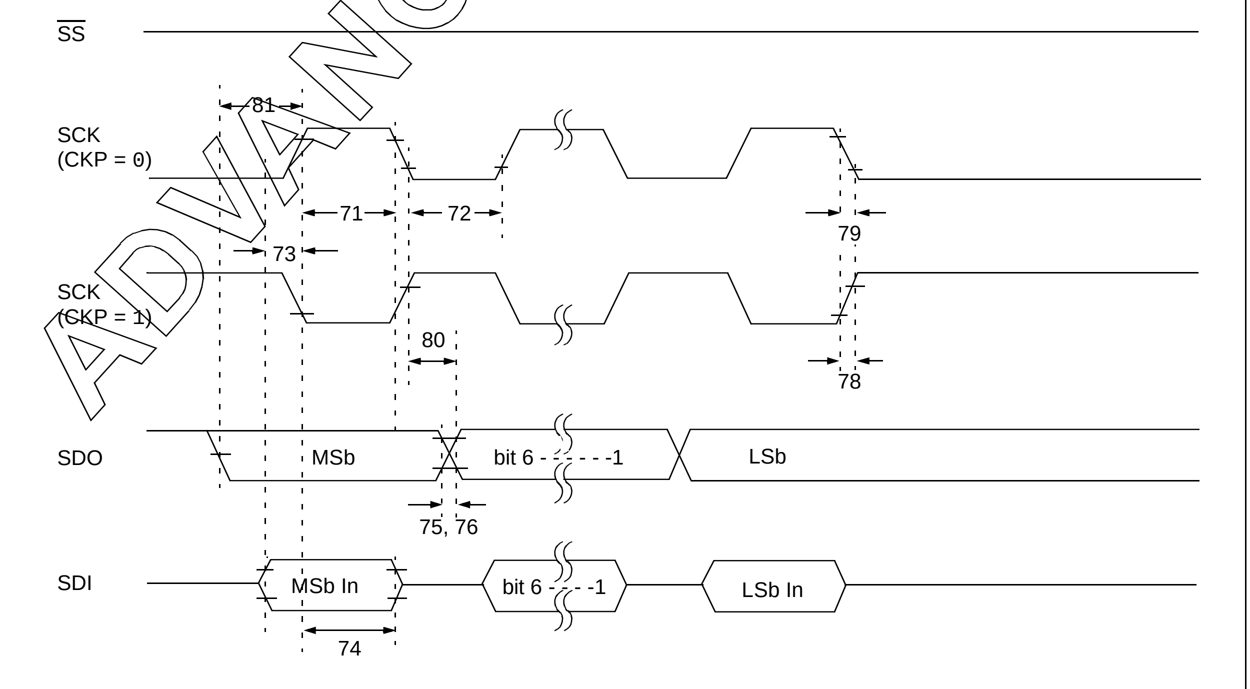
† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-11: SPI™ MASTER MODE TIMING (CKE = 0, SMP = 0)



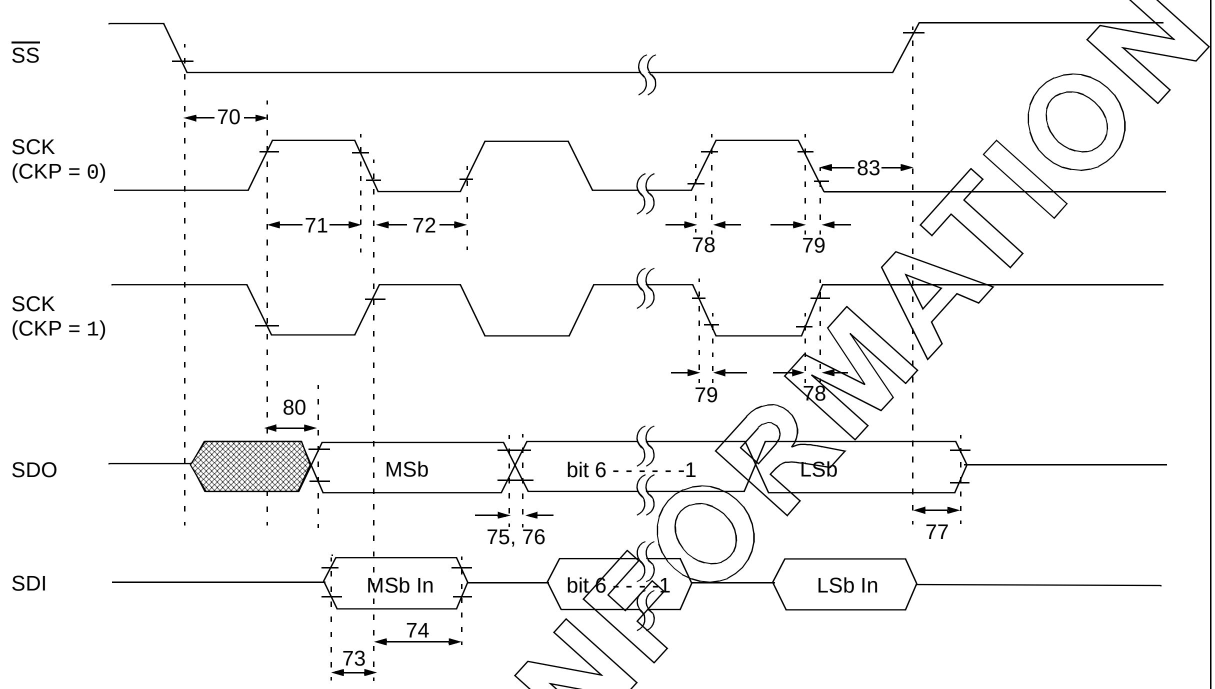
Note: Refer to Figure 19-2 for load conditions.

FIGURE 19-12: SPI™ MASTER MODE TIMING (CKE = 1, SMP = 1)



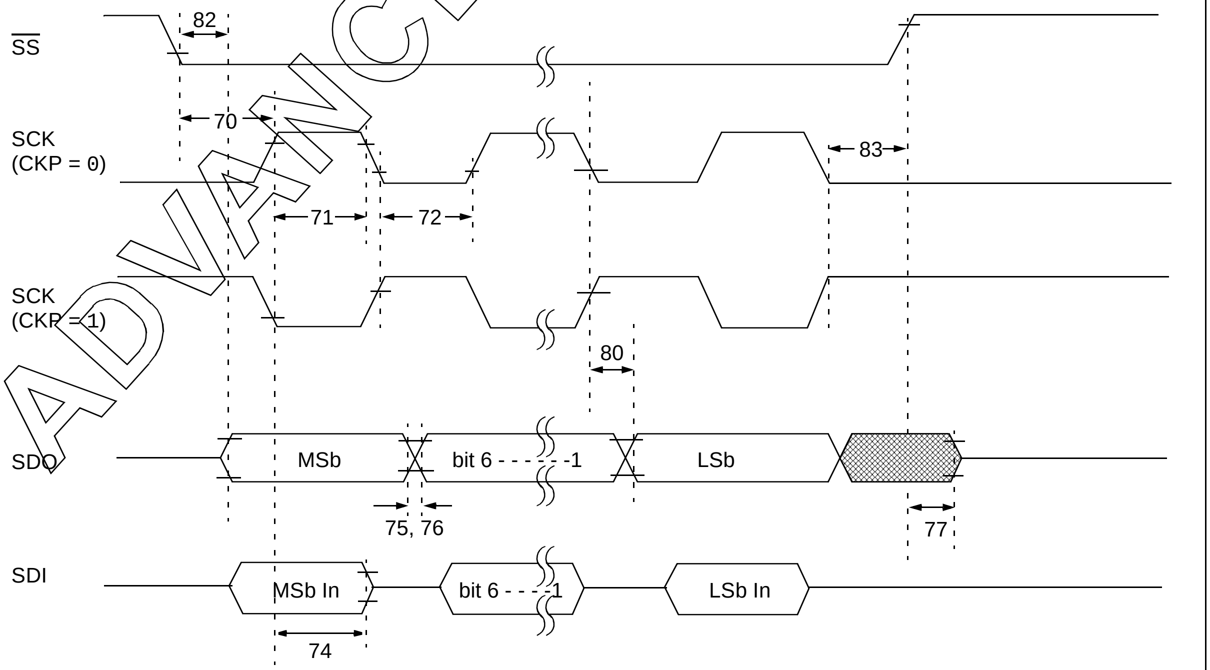
Note: Refer to Figure 19-2 for load conditions.

FIGURE 19-13: SPI™ SLAVE MODE TIMING (CKE = 0)



Note: Refer to Figure 19-2 for load conditions.

FIGURE 19-14: SPI™ SLAVE MODE TIMING (CKE = 1)



Note: Refer to Figure 19-2 for load conditions.

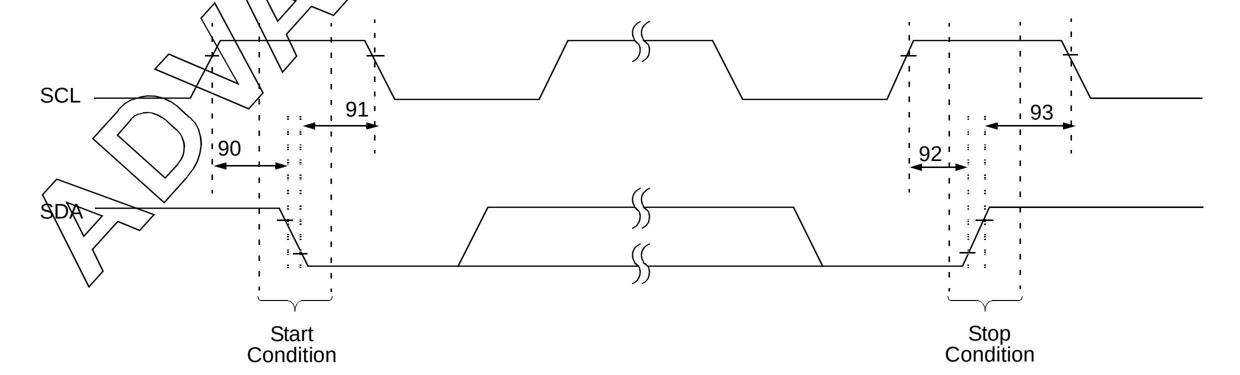
TABLE 19-12: SPI™ MODE REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}$↓ to SCK↓ or SCK↑ input		Tcy	—	—	ns	
71*	TscH	SCK input high time (Slave mode)		Tcy + 20	—	—	ns	
72*	TscL	SCK input low time (Slave mode)		Tcy + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge		100	—	—	ns	
74*	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	—	ns	
75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
76*	TdoF	SDO data output fall time		—	10	25	ns	
77*	TssH2doZ	$\overline{SS}$↑ to SDO output high-impedance		10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
79*	TscF	SCK output fall time (Master mode)		—	10	25	ns	
80*	TscH2doV, TscL2doV	SDO data output valid after SCK edge	3.0-5.5V	—	—	50	ns	
			2.0-5.5V	—	—	145	ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge		Tcy	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}$↓ edge		—	—	50	ns	
83*	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK edge		1.5Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-15: I^{2}C™ BUS START/STOP BITS TIMING



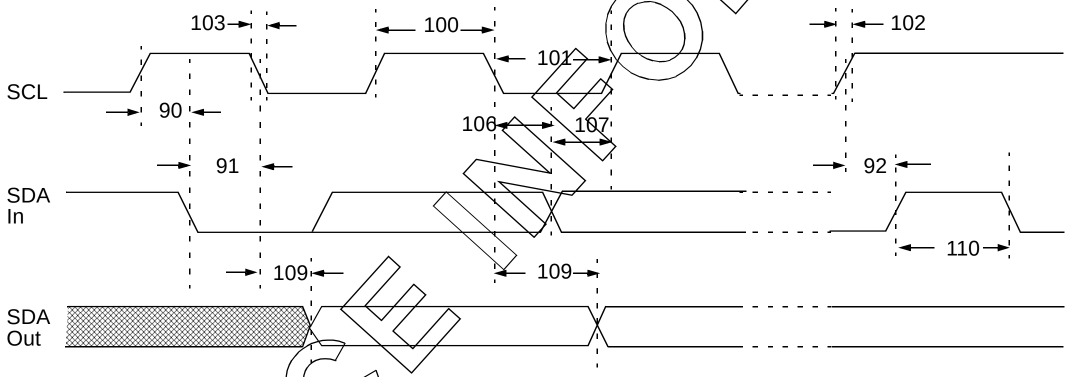
Note: Refer to Figure 19-2 for load conditions.

TABLE 19-13: I^{2}C™ BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
90*	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
91*	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
92*	TSU:STO	Stop condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
93	THD:STO	Stop condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 19-16: I^{2}C™ BUS DATA TIMING



Note: Refer to Figure 19-2 for load conditions.

TABLE 19-14: I^{2}C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
90*	TSU:STA	Start condition setup time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91*	THD:STA	Start condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92*	TSU:STO	Stop condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I^{2}C bus device can be used in a Standard mode (100 kHz) I^{2}C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I^{2}C bus specification), before the SCL line is released.

TABLE 19-15: PIC16F91X A/D CONVERTER CHARACTERISTICS:

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	10 bits	bit	
A02	EABS	Total Absolute Error*(1)	—	—	±1	LSb	VREF = 5.0V
A03	EIL	Integral Error	—	—	±1	LSb	VREF = 5.0V
A04	EDL	Differential Error	—	—	±1	LSb	No missing codes to 10 bits VREF = 5.0V
A05	EFS	Full-scale Range	2.2*	—	5.5*	V	
A06	EOFF	Offset Error	—	—	±1	LSb	VREF = 5.0V
A07	EGN	Gain Error	—	—	±1	LSb	VREF = 5.0V
A10	—	Monotonicity	—	assured(2)	—	—	VSS ≤ VAIN ≤ VREF+
A20	VREF	Reference Voltage (VREF+ – VREF-)	2.0	—	VDD + 0.3	V	
A21	VREF+	Reference Voltage High	AVDD – 2.5V	—	AVDD + 0.3V	V	
A22	VREF-	Reference Voltage Low	AVSS – 0.3V	—	VREF+ – 2.0V	V	
A25	VAIN	Analog Input Voltage	VSS	—	VREF	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	
A50	IREF	VREF Input Current **(Note 3)**	— —	— —	±5 ±150	μA μA	During VAIN acquisition. During A/D conversion cycle.

* These parameters are characterized but not tested.

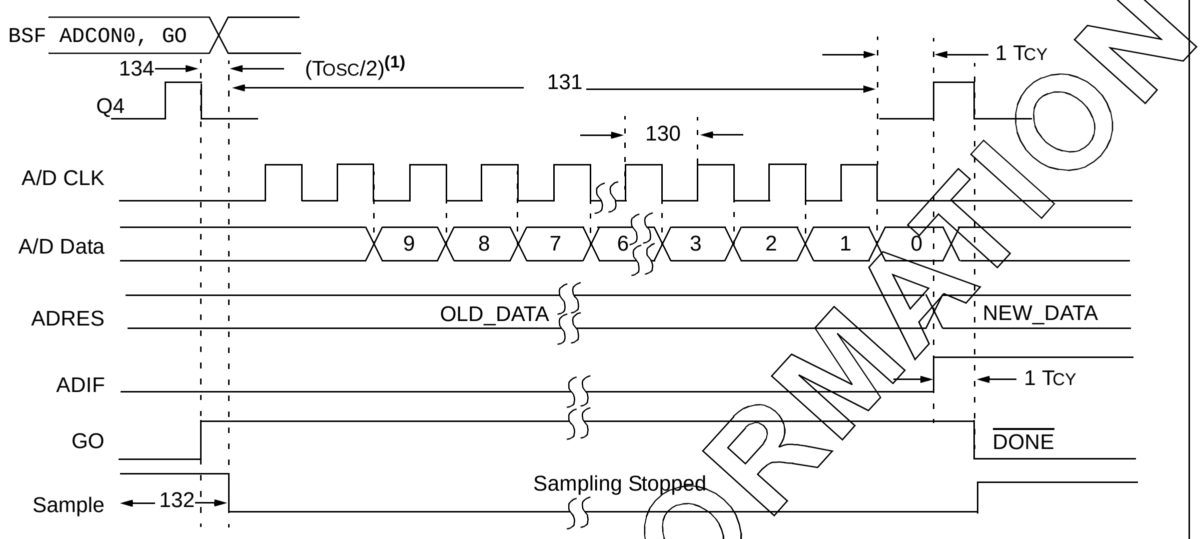
† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: VREF+ current is from RA3/AN3/C1+/VREF+/SEG15 pin or AVDD, whichever is selected as the VREF+ source. VREF- current is from RA2/AN2/C2+/VREF-/COM2 pin or AVSS, whichever is selected as the VREF- source.

FIGURE 19-17: PIC16F91X A/D CONVERSION TIMING (NORMAL MODE)



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

TABLE 19-16: PIC16F91X A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130	TAD	A/D Clock Period(2)	1.6	—	—	µs	TOSC-based, VREF ≥ 3.0V
			3.0*	—	—	µs	TOSC-based, VREF full range
130	TAD	A/D Internal RC Oscillator Period	3.0*	6.0	9.0*	µs	ADCS<1:0> = 11 (RC mode) At VDD = 2.5V
			2.0*	4.0	6.0*	µs	At VDD = 5.0V
131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	11	—	TAD	Set GO/DONE bit to new data in A/D Result register
132	TACQ	Acquisition Time		11.5	—	µs	
			5*	—	—	µs	The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD).
134	TGO	Q4 to A/D Clock Start	—	TOSC/2	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: ADRESH and ADRESL registers may be read on the following TCY cycle.

2: See Table 12-1 for minimum conditions.

20.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

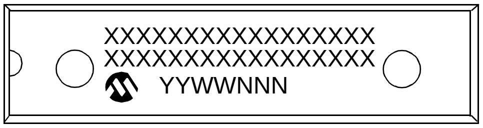
Graphs are not available at this time.

NOTES:

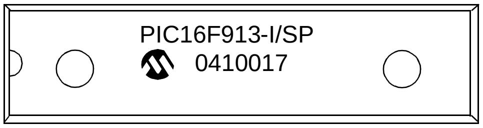
21.0 PACKAGING INFORMATION

21.1 Package Marking Information

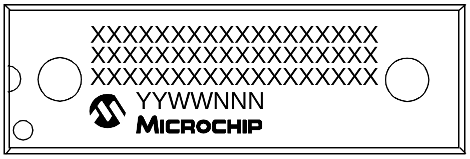
28-Lead PDIP (Skinny DIP)



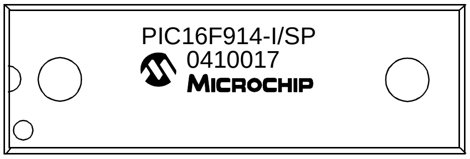
Example



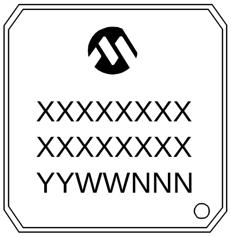
40-Lead PDIP



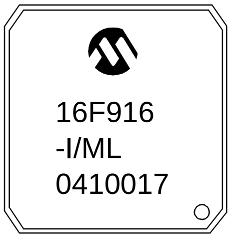
Example



28-Lead QFN



Example



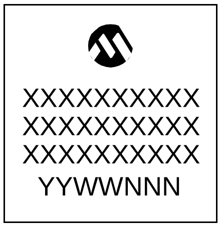
Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

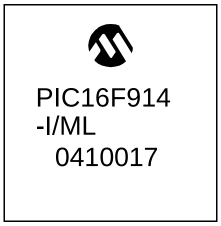
* Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

Package Marking Information (Continued)

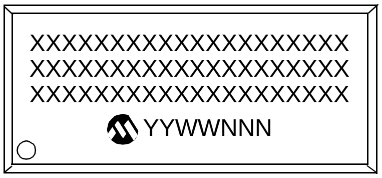
44-Lead QFN



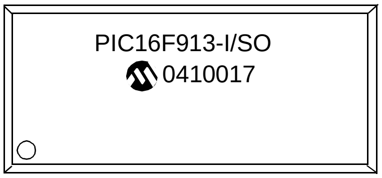
Example



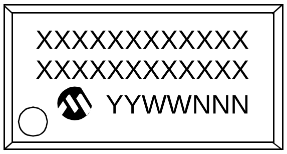
28-Lead SOIC



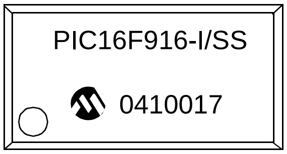
Example



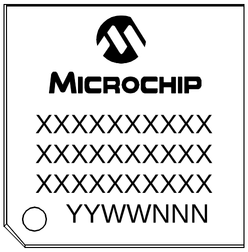
28-Lead SSOP



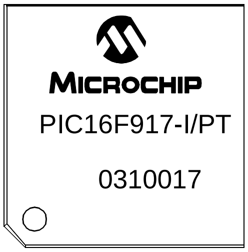
Example



44-Lead TQFP



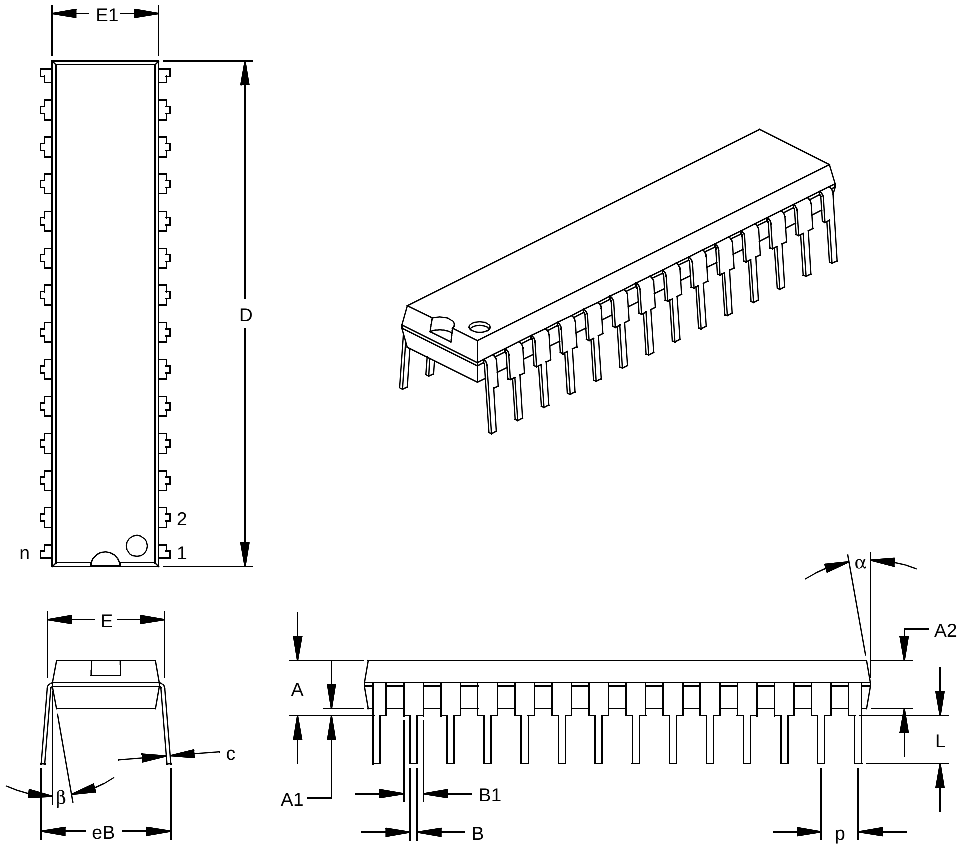
Example



21.2 Package Details

The following sections give the technical details of the packages.

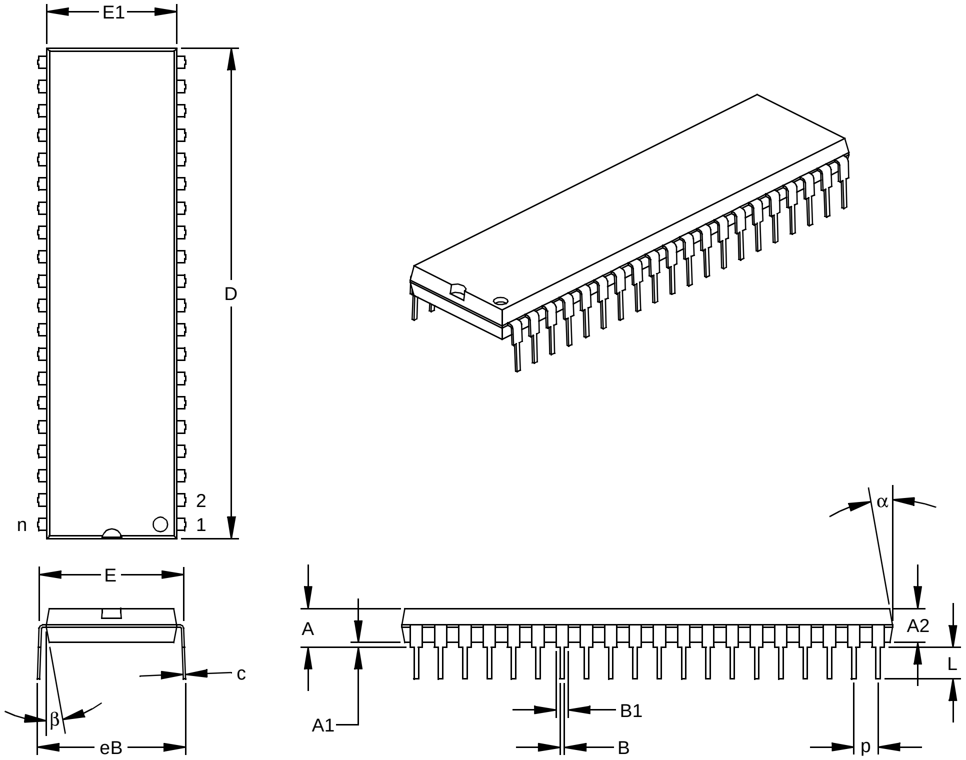
28-Lead Skinny Plastic Dual In-line (SP) – 300 mil Body (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.150	.160	3.56	3.81	4.06
Molded Package Thickness	A2	.125	.130	.135	3.18	3.30	3.43
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.310	.325	7.62	7.87	8.26
Molded Package Width	E1	.275	.285	.295	6.99	7.24	7.49
Overall Length	D	1.345	1.365	1.385	34.16	34.67	35.18
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.040	.053	.065	1.02	1.33	1.65
Lower Lead Width	B	.016	.019	.022	0.41	0.48	0.56
Overall Row Spacing §	eB	.320	.350	.430	8.13	8.89	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MO-095
Drawing No. C04-070

40-Lead Plastic Dual In-line (P) – 600 mil Body (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	40			40		
Pitch	p		.100			2.54	
Top to Seating Plane	A	.160	.175	.190	4.06	4.45	4.83
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.620	.650	.680	15.75	16.51	17.27
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter

§ Significant Characteristic

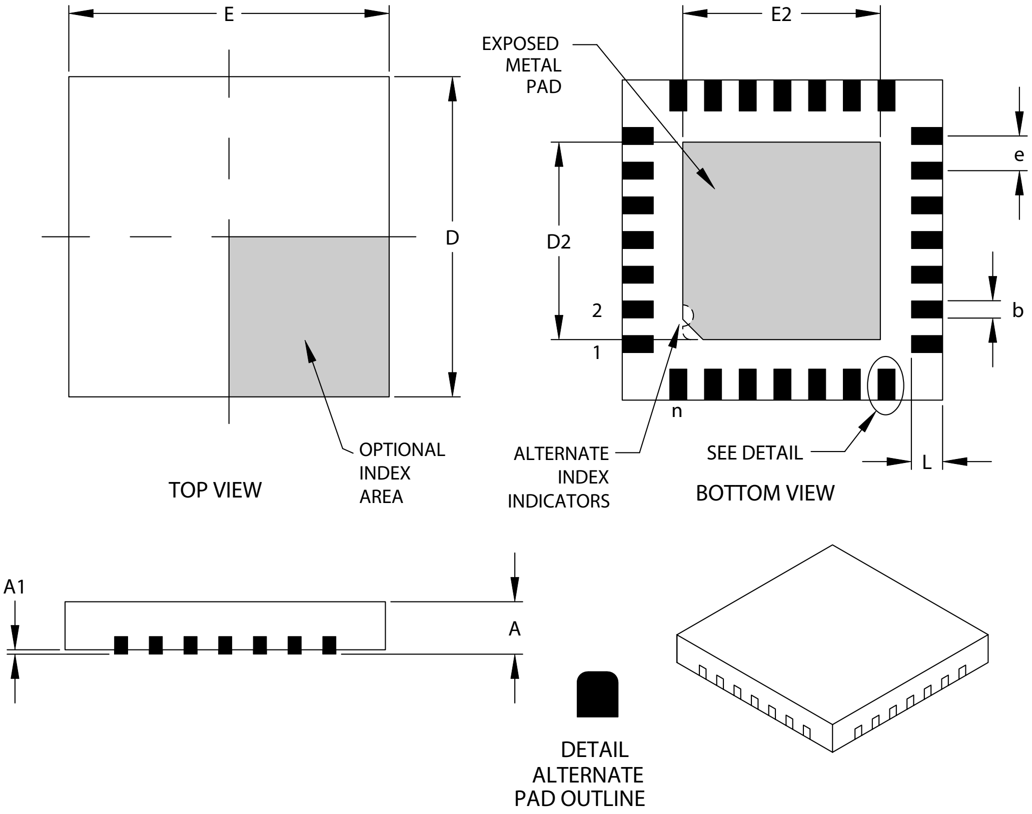
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.

JEDEC Equivalent: MO-011

Drawing No. C04-016

28-Lead Plastic Quad Flat No Lead Package (ML) 6x6 mm Body (QFN) – With 0.55 mm Contact Length (Saw Singulated)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	28			28		
Pitch	e	.026 BSC			0.65 BSC		
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0.00	0.02	0.05
Contact Thickness	A3	.008 REF			0.20 REF		
Overall Width	E	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Width	E2	.140	.146	.152	3.55	3.70	3.85
Overall Length	D	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Length	D2	.140	.146	.152	3.55	3.70	3.85
Contact Width	b	.009	.011	.013	0.23	0.28	0.33
Contact Length	L	.020	.024	.028	0.50	0.60	0.70

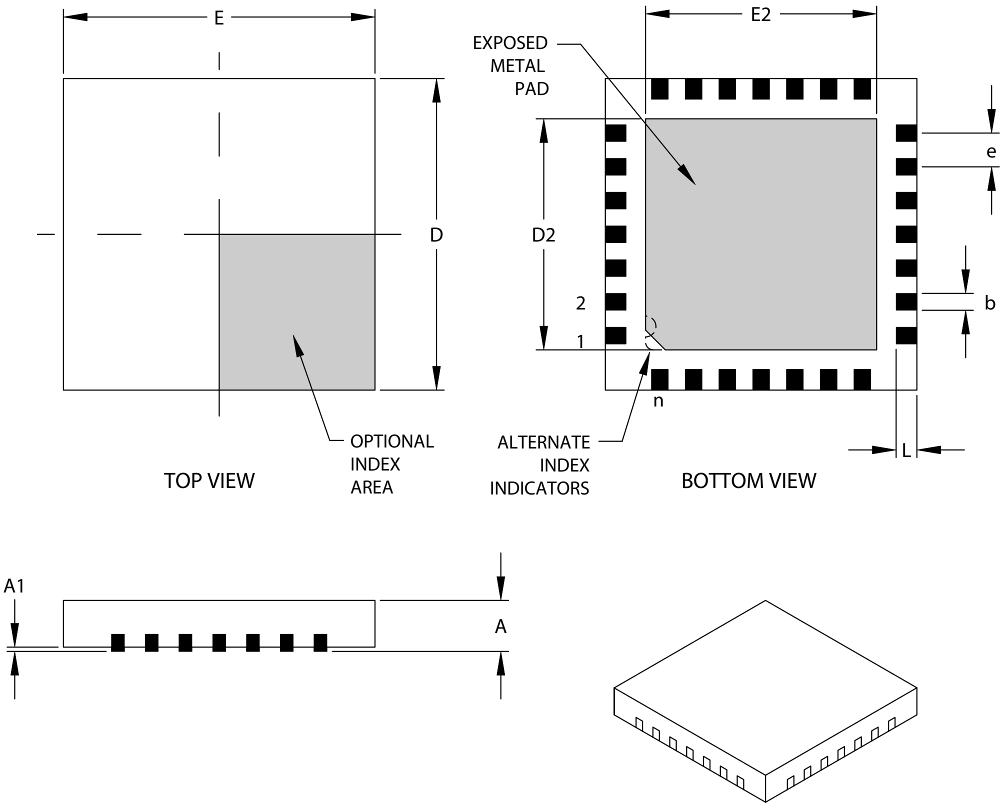
*Controlling Parameter

Notes:

JEDEC equivalent: MO-220

Drawing No. C04-105 Revised 05-24-04

28-Lead Plastic Quad Flat No Lead Package (ML) 6x6x0.9 mm Body (QFN-S) – With 0.40 mm Contact Length (Saw Singulated)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	e		.026 BSC			0.65 BSC	
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0.00	0.02	0.05
Overall Width	E	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Width	E2	.169	.175	.177	4.30	4.45	4.50
Overall Length	D	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Length	D2	.169	.175	.177	4.30	4.45	4.50
Lead Width	b	.013	.015	.017	0.33	0.38	0.43
Lead Length	L	.012	.016	.020	0.30	0.40	0.50

*Controlling Parameter

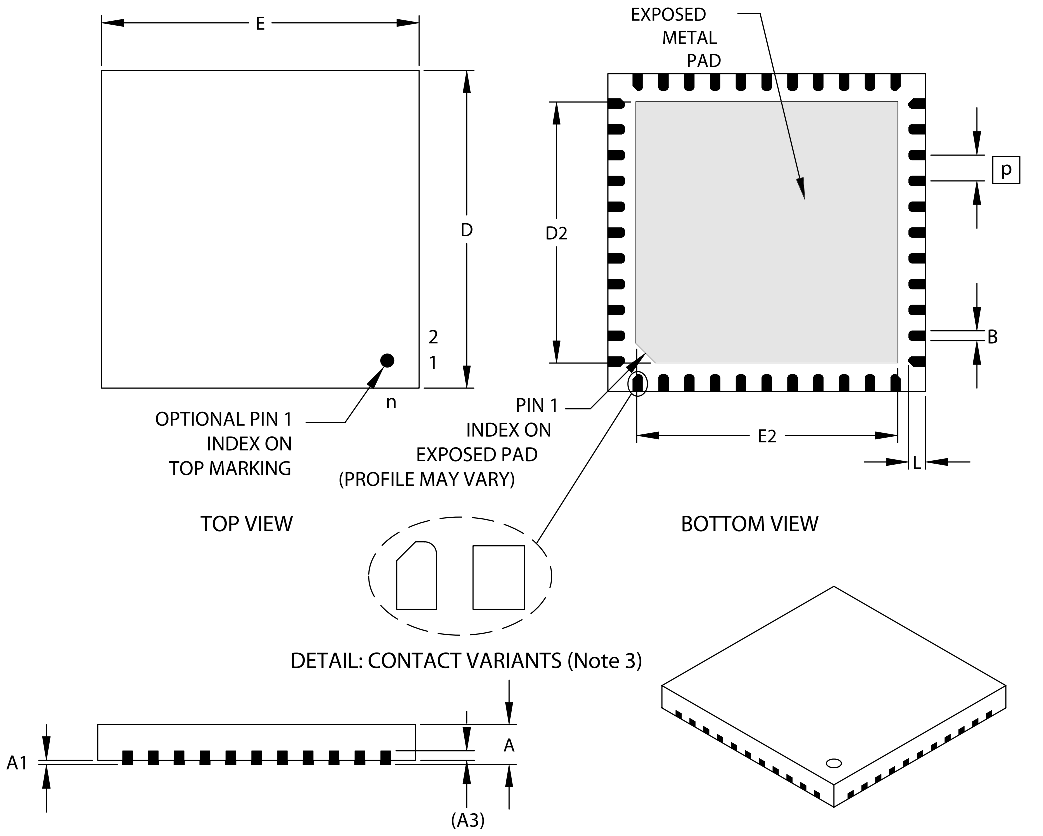
Notes:

JEDEC equivalent: MO-220

Drawing No. C04-124

Revised 05/24/04

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



	Units	INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Contacts	n		44			44	
Pitch	p		.026 BSC [1]			0.65 BSC [1]	
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0	0.02	0.05
Base Thickness	(A3)		.010 REF [2]			0.25 REF [2]	
Overall Width	E	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Width	E2	.246	.268	.274	6.25	6.80	6.95
Overall Length	D	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Length	D2	.246	.268	.274	6.25	6.80	6.95
Contact Width	B	.008	.013	.013	0.20	0.33	0.35
Contact Length	L	.014	.016	.019	0.35	0.40	0.48

*Controlling Parameter

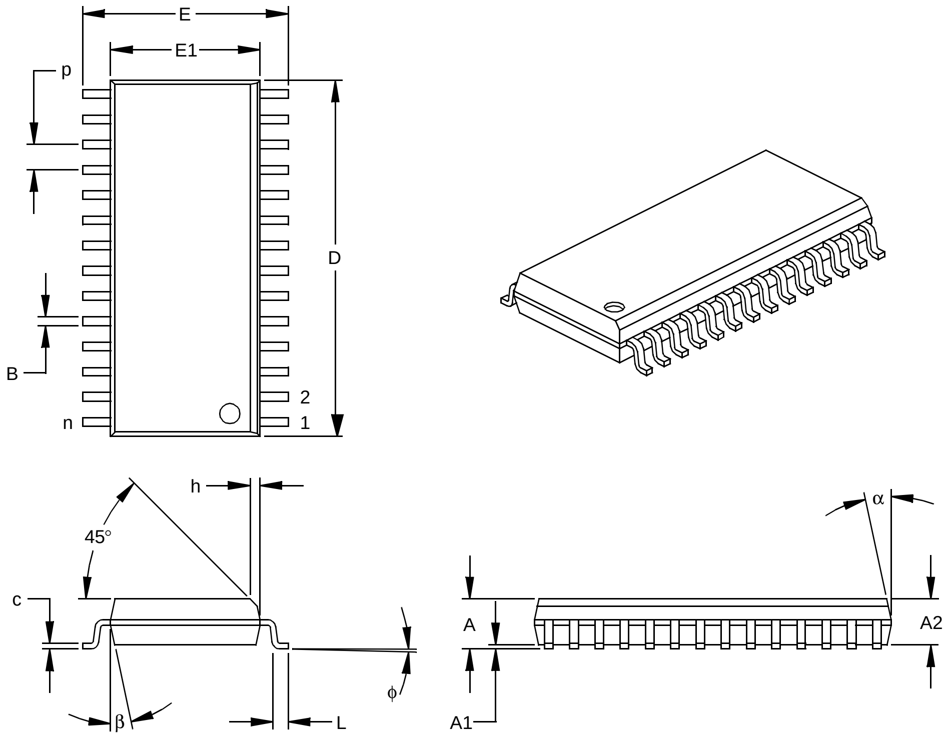
Notes:

1. BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   See ASME Y14.5M
2. REF: Reference Dimension, usually without tolerance, for information purposes only.
   See ASME Y14.5M
3. Contact profiles may vary.

JEDEC equivalent: M0-220

Drawing No. C04-103

28-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)

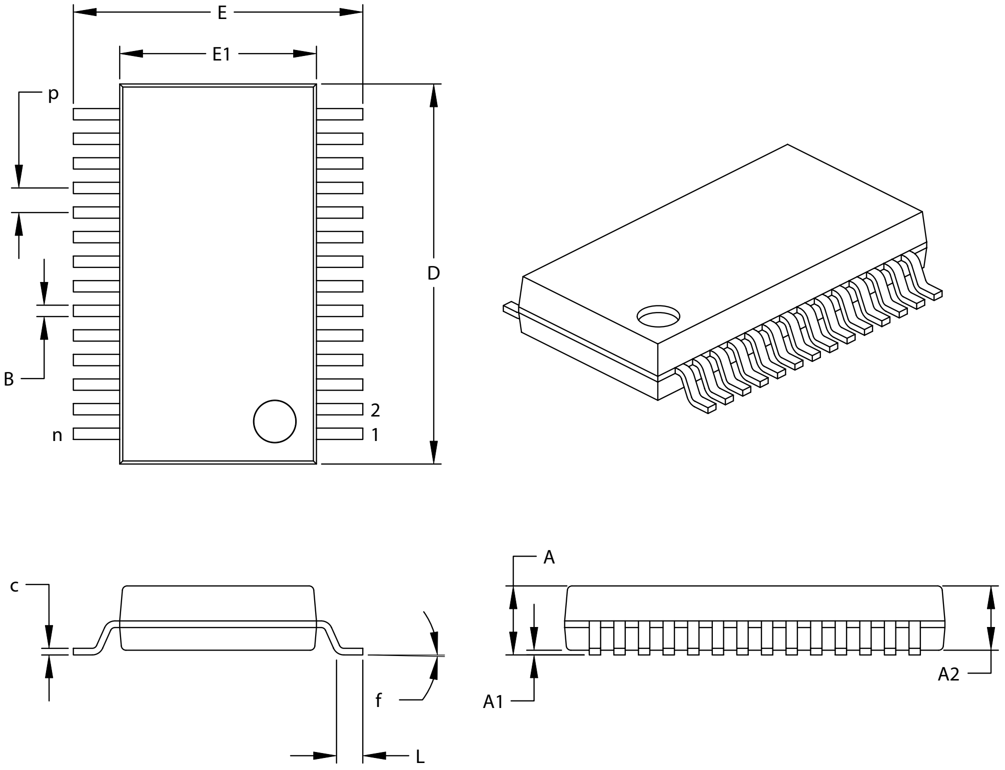


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.288	.295	.299	7.32	7.49	7.59
Overall Length	D	.695	.704	.712	17.65	17.87	18.08
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle Top	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.013	0.23	0.28	0.33
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-052

28-Lead Plastic Shrink Small Outline (SS) – 209 mil Body, 5.30 mm (SSOP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.026			0.65	
Overall Height	A	-	-	.079	-	-	2.0
Molded Package Thickness	A2	.065	.069	.073	1.65	1.75	1.85
Standoff	A1	.002	-	-	0.05	-	-
Overall Width	E	.295	.307	.323	7.49	7.80	8.20
Molded Package Width	E1	.009	.209	.220	5.00	5.30	5.60
Overall Length	D	.390	.402	.413	9.90	10.20	10.50
Foot Length	L	.022	.030	.037	0.55	0.75	0.95
Lead Thickness	c	.004	-	.010	0.09	-	0.25
Foot Angle	f	0°	4°	8°	0°	4°	8°
Lead Width	B	.009	-	.015	0.22	-	0.38

*Controlling Parameter

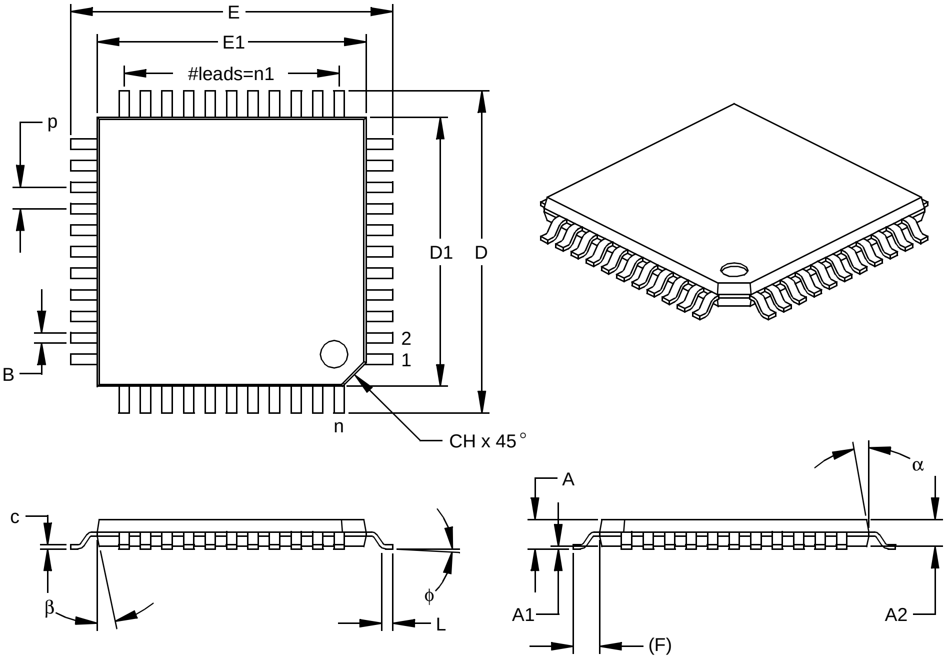
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-150

Drawing No. C04-073

44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	44			44		
Pitch	p		.031			0.80	
Pins per Side	n1		11			11	
Overall Height	A	.039	.043	.047	1.00	1.10	1.20
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Foot Length	L	.018	.024	.030	0.45	0.60	0.75
Footprint (Reference)	(F)		.039		1.00		
Foot Angle	φ	0	3.5	7	0	3.5	7
Overall Width	E	.463	.472	.482	11.75	12.00	12.25
Overall Length	D	.463	.472	.482	11.75	12.00	12.25
Molded Package Width	E1	.390	.394	.398	9.90	10.00	10.10
Molded Package Length	D1	.390	.394	.398	9.90	10.00	10.10
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.012	.015	.017	0.30	0.38	0.44
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-026
Drawing No. C04-076

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A

This is a new data sheet.

APPENDIX B: MIGRATING FROM OTHER PICmicro® DEVICES

This discusses some of the issues in migrating from other PICmicro devices to the PIC16F91X family of devices.

B.1 PIC16F676 to PIC16F91X

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F676	PIC16F91X
Max Operating Speed	20 MHz	20 MHz
Max Program Memory (Words)	1K	8K
Max SRAM (Bytes)	64	352
A/D Resolution	10-bit	10-bit
Data EEPROM (bytes)	128	256
Timers (8/16-bit)	1/1	2/1
Oscillator Modes	8	8
Brown-out Reset	Y	Y
Internal Pull-ups	RB0/1/2/4/5	RB<7:0>
Interrupt-on-change	RB0/1/2/3/4/5	RB<7:4>
Comparator	1	2
USART	N	Y
Extended WDT	N	Y
Software Control Option of WDT/BOR	N	Y
INTOSC Frequencies	4 MHz	32 kHz - 8 MHz
Clock Switching	N	Y

APPENDIX C: CONVERSION CONSIDERATIONS

Considerations for converting from previous versions of devices to the ones listed in this data sheet are listed in Table C-1.

TABLE C-1: CONVERSION CONSIDERATIONS

Characteristic	PIC16F91X	PIC16F87X	PIC16F87XA
Pins	28/40	28/40	28/40
Timers	3	3	3
Interrupts	11 or 12	13 or 14	14 or 15
Communication	USART, SSP (SPI™, I^{2}C™ Slave)	PSP, USART, SSP (SPI, I^{2}C Master/Slave)	PSP, USART, SSP (SPI, I^{2}C Master/Slave)
Frequency	20 MHz	20 MHz	20 MHz
Voltage	2.0V-5.5V	2.2V-5.5V	2.0V-5.5V
A/D	10-bit, 7 conversion clock selects	10-bit, 4 conversion clock selects	10-bit, 7 conversion clock selects
CCP	2	2	2
Comparator	2	—	2
Comparator Voltage Reference	Yes	—	Yes
Program Memory	4K, 8K EPROM	4K, 8K Flash (Erase/Write on single-word)	4K, 8K Flash (Erase/Write on four-word blocks)
RAM	256, 352 bytes	192, 368 bytes	192, 368 bytes
EEPROM Data	256 bytes	128, 256 bytes	128, 256 bytes
Code Protection	On/Off	Segmented, starting at end of program memory	On/Off
Program Memory Write Protection	—	On/Off	Segmented, starting at beginning of program memory
LCD Module	16, 24 segment drivers, 4 commons	—	—
Other	In-Circuit Debugger, Low-Voltage Programming	In-Circuit Debugger, Low-Voltage Programming	In-Circuit Debugger, Low-Voltage Programming

INDEX

NOTES:

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

042003

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager   Total Pages Sent ________

RE: Reader Response

From: Name ____

Company ____

Address ____

City / State / ZIP / Country ______________________________

Telephone: (_______) _________ - _________   FAX: (______) _________ - _________

Application (optional):

Would you like a reply?____Y ____N

Device: PIC16F91X   Literature Number: DS41250A

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this document easy to follow? If not, why?

____

____

4. What additions to the document do you think would enhance the structure and subject?

____

____

5. What deletions from the document could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	**Temperature Range**	**Package**	**Pattern**

Device	PIC16F91X: Standard V_{DD} range PIC16F91XT: (Tape and Reel)

Temperature Range	I = -40°C to +85°C
	E = -40°C to +125°C

Package	P = PDIP
	SL = SOIC (Gull wing, 150 mil body)
	ST = TSSOP(4.4 mm)

Pattern	3-Digit Pattern Code for QTP (blank otherwise)

Examples:

a) PIC16F913-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC16F913-I/SO = Industrial Temp., SOIC package, 20 MHz

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
480-792-7627
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westford, MA
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-750-3506
Fax: 86-591-750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Tel: 86-755-8290-1380
Fax: 86-755-8295-1393

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Qingdao
Tel: 86-532-502-7355
Fax: 86-532-502-7205

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

India - New Delhi
Tel: 91-11-5160-8632
Fax: 91-11-5160-8632

Japan - Kanagawa
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Kaohsiung
Tel: 886-7-536-4816
Fax: 886-7-536-4817

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Taiwan - Hsinchu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

EUROPE

Austria - Weis
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark - Ballerup
Tel: 45-4420-9895
Fax: 45-4420-9910

France - Massy
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Ismaning
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

England - Berkshire
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/24/04