



# **PIC16F193X/LF193X**

## **Data Sheet**

28/40/44-Pin Flash-Based, 8-Bit  
CMOS Microcontrollers with  
LCD Driver and 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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rPIC, SmartShunt and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, 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, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICTail, PIC<sup>32</sup> logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, WiperLock and ZENA 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.

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

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

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> 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.*



# MICROCHIP

# PIC16F193X/LF193X

## 28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers

### Devices Included In This Data Sheet:

#### PIC16F193X Devices:

- PIC16F1933
- PIC16F1936
- PIC16F1938
- PIC16F1934
- PIC16F1937
- PIC16F1939

#### PIC16LF193X Devices:

- PIC16LF1933
- PIC16LF1936
- PIC16LF1938
- PIC16LF1934
- PIC16LF1937
- PIC16LF1939

### High-Performance RISC CPU:

- Only 49 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 32 MHz oscillator/clock input
  - DC – 125 ns instruction cycle
- Up to 16K x 14 Words of Flash Program Memory
- Up to 1024 Bytes of Data Memory (RAM)
- Interrupt Capability
- 16-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes
- Processor Read Access to Program Memory
- Pinout Compatible to other 28/40-pin PIC16CXXX and PIC16FXXX Microcontrollers

### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to  $\pm 1\%$ , typical
  - Software selectable frequency range from 32 MHz to 31 kHz
- Power-Saving Sleep mode
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR)
  - Selectable between two trip points
  - Disable in Sleep option
- 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
- Wide Operating Voltage Range:
  - 1.8V-5.5V (PIC16F193X)
  - 1.8V-3.6V (PIC16LF193X)

### Low-Power Features:

- Standby Current:
  - 100 nA @ 2.0V, typical (PIC16LF193X)
- Operating Current:
  - 6.0  $\mu$ A @ 32 kHz, 2.0V, typical (PIC16LF193X)
  - 54  $\mu$ A @ 1 MHz, 2.0V, typical (PIC16LF193X)
- Low-Power Watchdog Timer Current:
  - 1.0  $\mu$ A @ 2.0V, typical (PIC16LF193X)

### Peripheral Features:

- Up to 35 I/O Pins and 1 Input-only pin:
  - High-current source/sink for direct LED drive
  - Individually programmable Interrupt-on-pin change pins
  - Individually programmable weak pull-ups
- A/D Converter:
  - 10-bit resolution and up to 14 channels
  - Can operate during Sleep
  - Selectable 1.024/2.048/4.096V voltage reference
- Timer0: 8-Bit Timer/Counter with 8-Bit Programmable Prescaler
- Enhanced Timer1
  - Dedicated low-power 32 kHz oscillator
  - 16-bit timer/counter with prescaler
  - External Gate Input mode with toggle and single shot modes
  - Interrupt-on-gate completion
- Timer2, 4, 6: 8-Bit Timer/Counter with 8-Bit Period Register, Prescaler and Postscaler
- Two Capture, Compare, PWM Modules (CCP)
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Three Enhanced Capture, Compare, PWM modules (ECCP)
  - Software selectable time-bases
  - Auto-shutdown and auto-restart
  - PWM steering
- Master Synchronous Serial Port (SSP) with SPI and I<sup>2</sup>C™ with:
  - 7-bit address masking
  - SMBUS/PMBUS™ compatibility
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)
  - RS-232, RS 485 and LIN compatible
  - Auto-Baud Detect
  - Auto-wake-up on start

# PIC16F193X/LF193X

## Peripheral Features (Continued):

- SR Latch (555 Timer):
  - Multiple Set/Reset input options
- Integrated LCD Controller:
  - Up to 96 segments
  - Variable clock input
  - Contrast control
  - Internal voltage reference selections
- 2 Comparators:
  - Rail-to-rail inputs/outputs
  - Power mode control
  - Software enable hysteresis
- Voltage Reference module:
  - Fixed Voltage Reference (FVR) with 1.024V, 2.048V and 4.096V output levels
  - 5-bit rail-to-rail resistive DAC with positive and negative reference selection

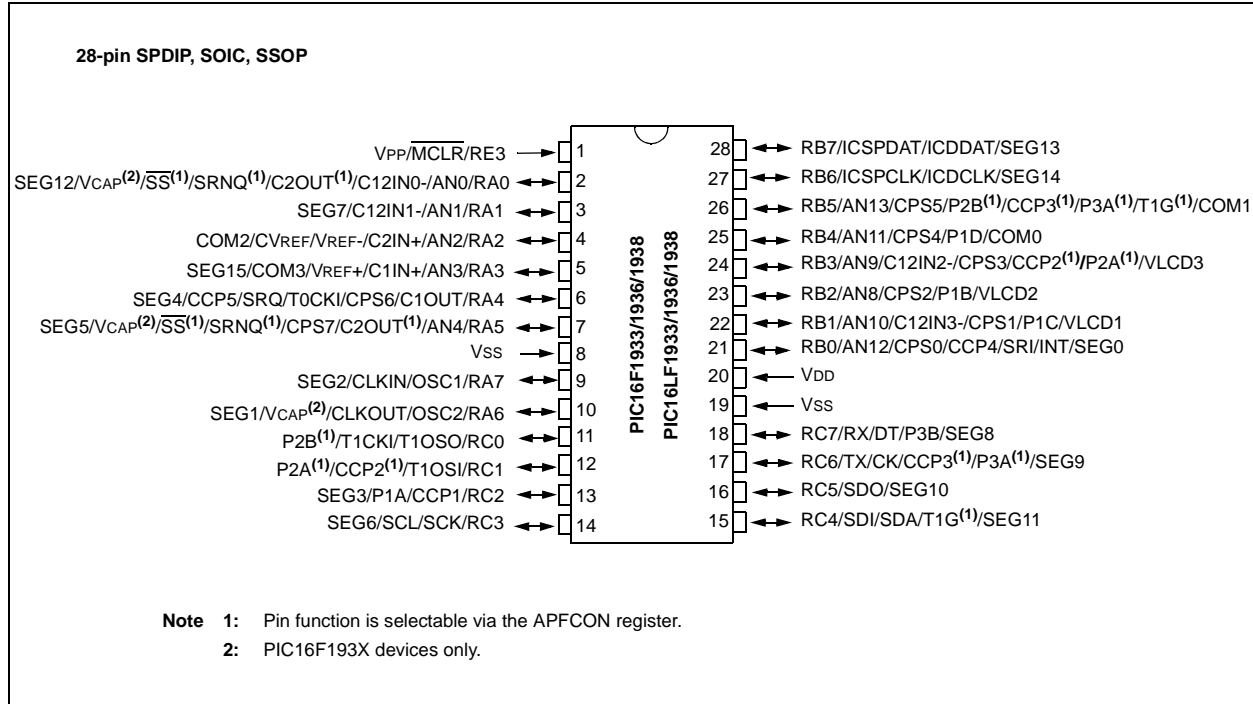
## PIC16F193X/LF193X Family Types

Device	Program Memory Flash (words)	Data EEPROM (bytes)	SRAM (bytes)	I/Os	10-bit A/D (ch)	Comparators	Timers 8/16-bit	EUSART	MI <sup>2</sup> C/SPI	ECCP Full Bridge	ECCP Half Bridge	CCP	LCD
PIC16F1933 PIC16LF1933	4096	256	256	25	11	2	4/1	Yes	Yes	1	2	2	16 <sup>(1)</sup> /4
PIC16F1934 PIC16LF1934	4096	256	256	36	14	2	4/1	Yes	Yes	2	1	2	24/4
PIC16F1936 PIC16LF1936	8192	256	512	25	11	2	4/1	Yes	Yes	1	2	2	16 <sup>(1)</sup> /4
PIC16F1937 PIC16LF1937	8192	256	512	36	14	2	4/1	Yes	Yes	2	1	2	24/4
PIC16F1938 PIC16LF1938	16384	256	1024	25	14	2	4/1	Yes	Yes	1	2	2	16 <sup>(1)</sup> /4
PIC16F1939 PIC16LF1939	16384	256	1024	36	14	2	4/1	Yes	Yes	2	1	2	24/4

**Note 1:** COM3 and SEG15 share the same physical pin on PIC16F1933/1936/1938/PIC16LF1933/1936/1938, therefore, SEG15 is not available when using 1/4 multiplex displays.

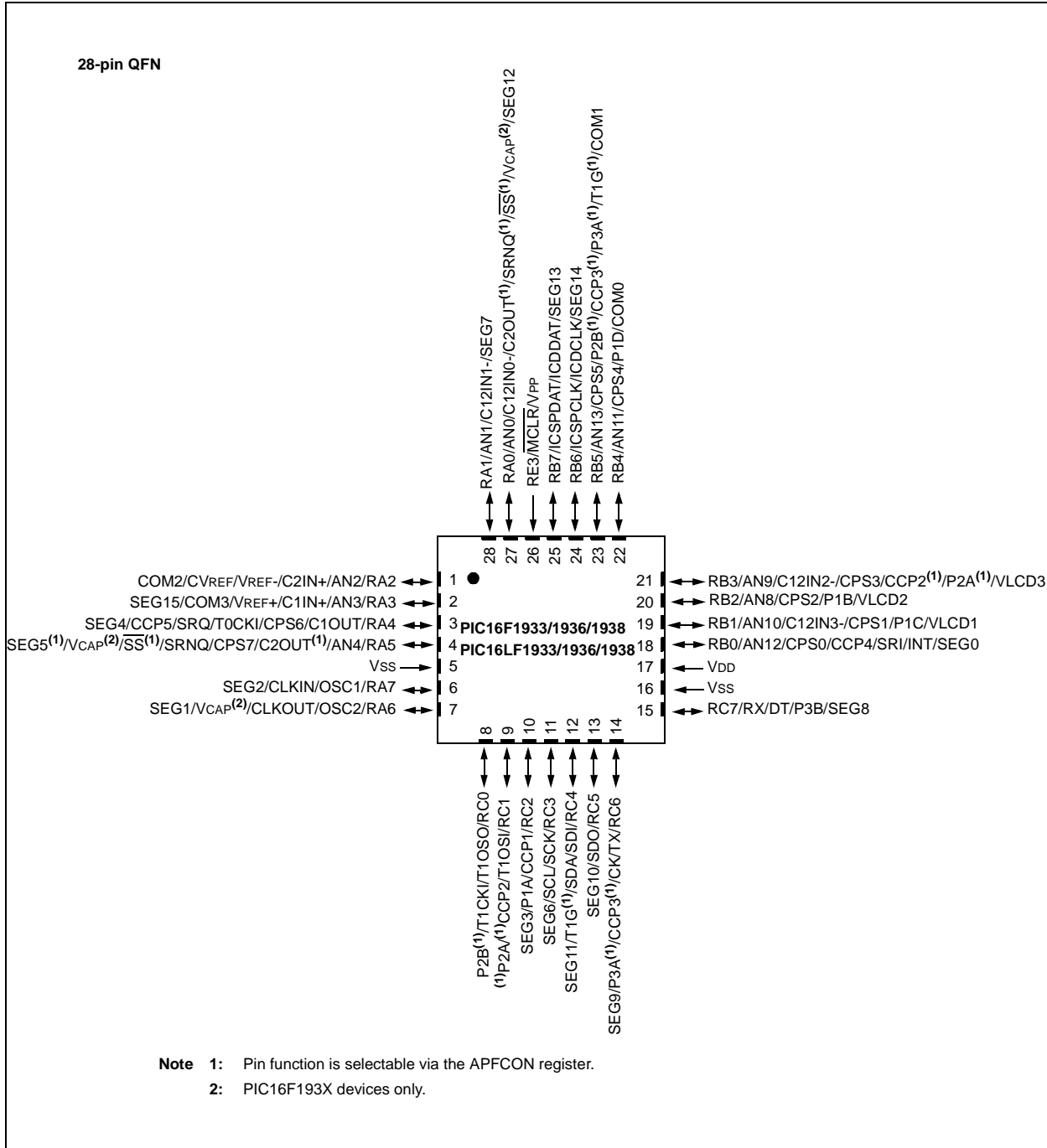
# PIC16F193X/LF193X

## Pin Diagram – 28-Pin SPDIP/SOIC/SSOP (PIC16F1933/1936/1938, PIC16LF1933/1936/1938)



# PIC16F193X/LF193X

## Pin Diagram – 28-Pin QFN (PIC16F1933/1936/1938, PIC16LF1933/1936/1938)



# PIC16F193X/LF193X

**TABLE 1: 28-PIN SUMMARY (PIC16F1933/1936/1938, PIC16LF1933/1936/1938)**

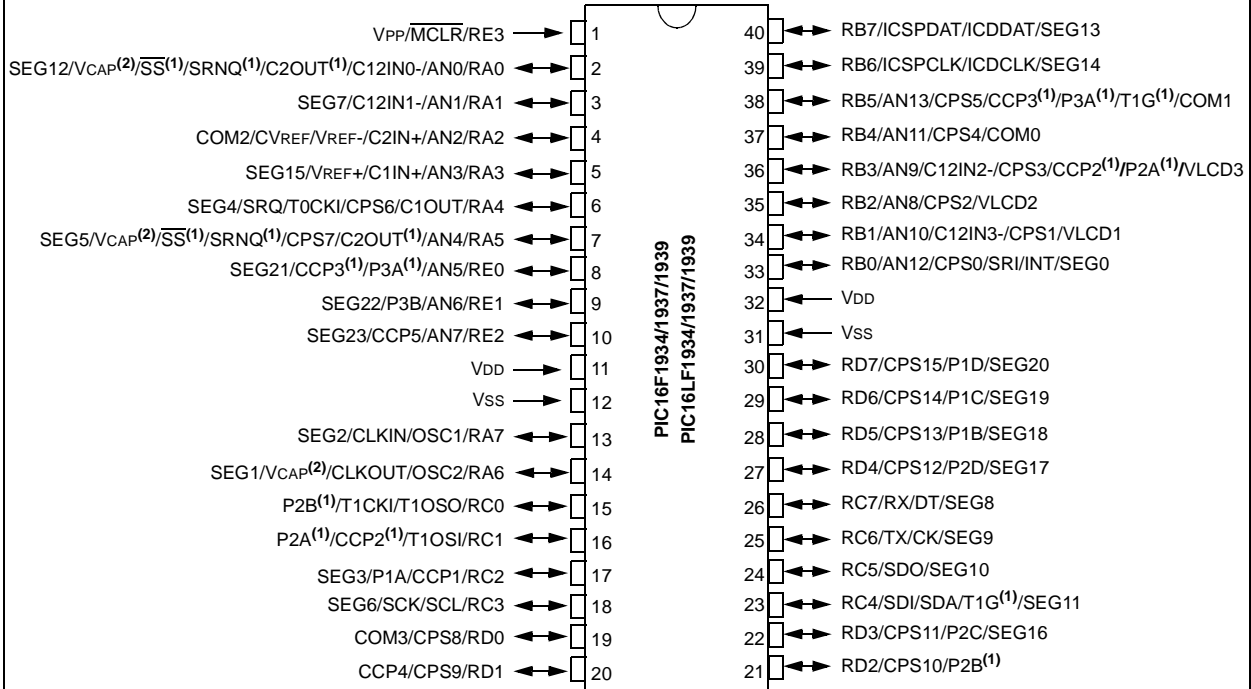
I/O	28-Pin SIP	28-Pin QFN	A/D	Cap Sense	Comparator	SR Latch	Timers	CCP	EUSART	MSSP	LCD	Interrupt	Pull-up	Basic
RA0	2	27	AN0	—	C12IN0-/C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG12	—	—	VCAP <sup>(2)</sup>
RA1	3	28	AN1	—	C12IN1-	—	—	—	—	—	SEG7	—	—	—
RA2	4	1	AN2/ VREF-	—	C2IN+/ CVREF	—	—	—	—	—	COM2	—	—	—
RA3	5	2	AN3/ VREF+	—	C1IN+	—	—	—	—	—	SEG15/ COM3	—	—	—
RA4	6	3	—	CPS6	C1OUT	SRQ	T0CKI	CCP5	—	—	SEG4	—	—	—
RA5	7	4	AN4	CPS7	C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG5	—	—	VCAP <sup>(2)</sup>
RA6	10	7	—	—	—	—	—	—	—	—	SEG1	—	—	OSC2/ CLKOUT VCAP <sup>(2)</sup>
RA7	9	6	—	—	—	—	—	—	—	—	SEG2	—	—	OSC1/ CLKIN
RB0	21	18	AN12	CPS0	—	SRI	—	CCP4	—	—	SEG0	INT/ IOC	Y	—
RB1	22	19	AN10	CPS1	C12IN3-	—	—	P1C	—	—	VLCD1	IOC	Y	—
RB2	23	20	AN8	CPS2	—	—	—	P1B	—	—	VLCD2	IOC	Y	—
RB3	24	21	AN9	CPS3	C12IN2-	—	—	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	VLCD3	IOC	Y	—
RB4	25	22	AN11	CPS4	—	—	—	P1D	—	—	COM0	IOC	Y	—
RB5	26	23	AN13	CPS5	—	—	T1G <sup>(1)</sup>	P2B <sup>(1)</sup> / CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	—	—	COM1	IOC	Y	—
RB6	27	24	—	—	—	—	—	—	—	—	SEG14	IOC	Y	ICSPCLK/ ICDCLK
RB7	28	25	—	—	—	—	—	—	—	—	SEG13	IOC	Y	ICSPDAT/ ICDDAT
RC0	11	8	—	—	—	—	T1OSO/ T1CKI	P2B <sup>(1)</sup>	—	—	—	—	—	—
RC1	12	9	—	—	—	—	T1OSI	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	—	—	—	—
RC2	13	10	—	—	—	—	—	CCP1/ P1A	—	—	SEG3	—	—	—
RC3	14	11	—	—	—	—	—	—	—	SCK/SCL	SEG6	—	—	—
RC4	15	12	—	—	—	—	T1G <sup>(1)</sup>	—	—	SDI/SDA	SEG11	—	—	—
RC5	16	13	—	—	—	—	—	—	—	SDO	SEG10	—	—	—
RC6	17	14	—	—	—	—	—	CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	TX/CK	—	SEG9	—	—	—
RC7	18	15	—	—	—	—	—	P3B	RX/DT	—	SEG8	—	—	—
RE3	1	26	—	—	—	—	—	—	—	—	—	—	Y	MCLR/VPP
VDD	20	17	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	8, 19	5, 16	—	—	—	—	—	—	—	—	—	—	—	VSS

**Note** 1: Pin functions can be moved using the APFCON register.  
 2: PIC16F193X devices only.

# PIC16F193X/LF193X

## Pin Diagram – 40-Pin PDIP (PIC16F1934/1937/1939, PIC16LF1934/1937/1939)

### 40-Pin PDIP

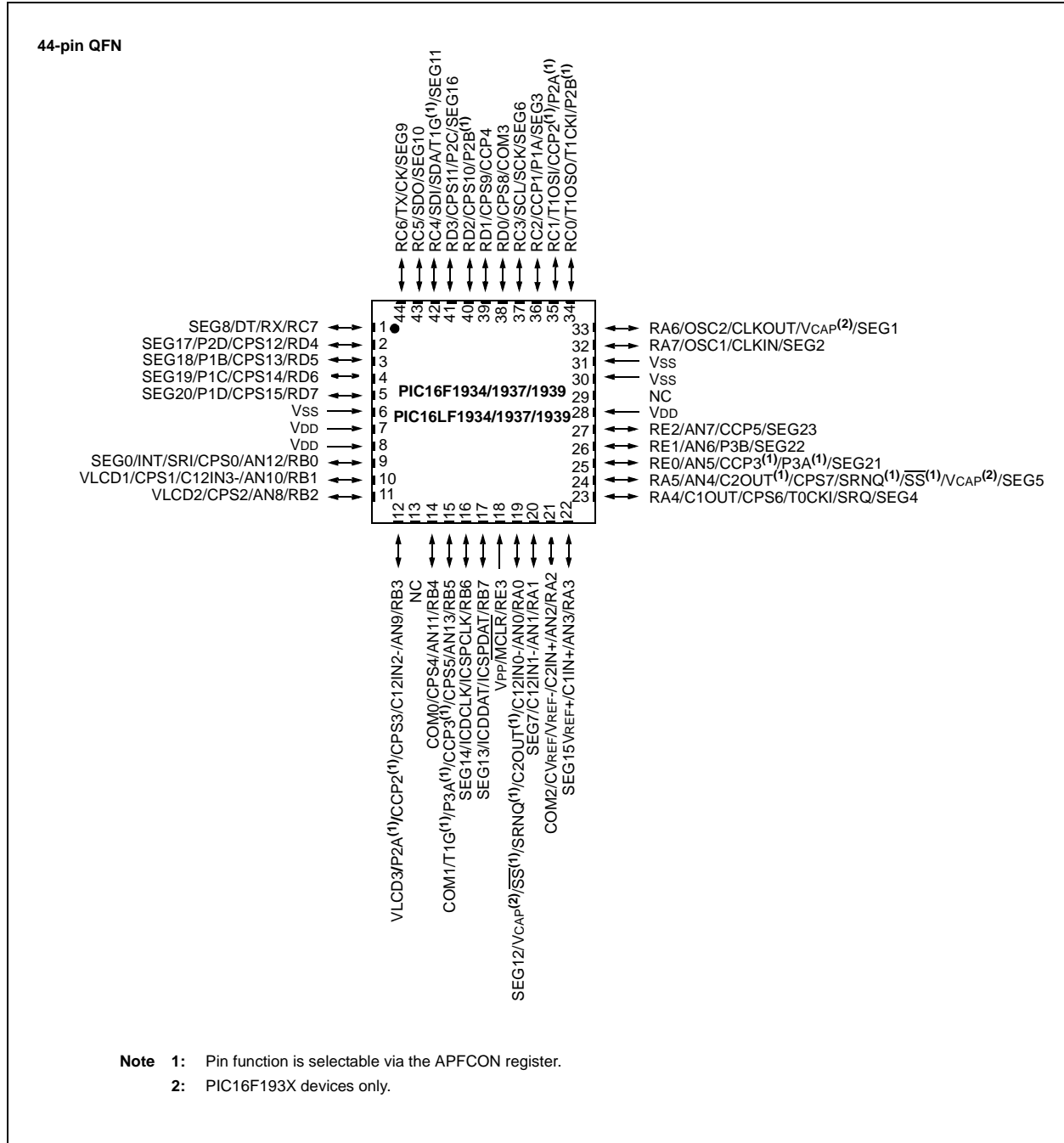


- Note** 1: Pin function is selectable via the APFCON register.  
 2: PIC16F193X devices only.



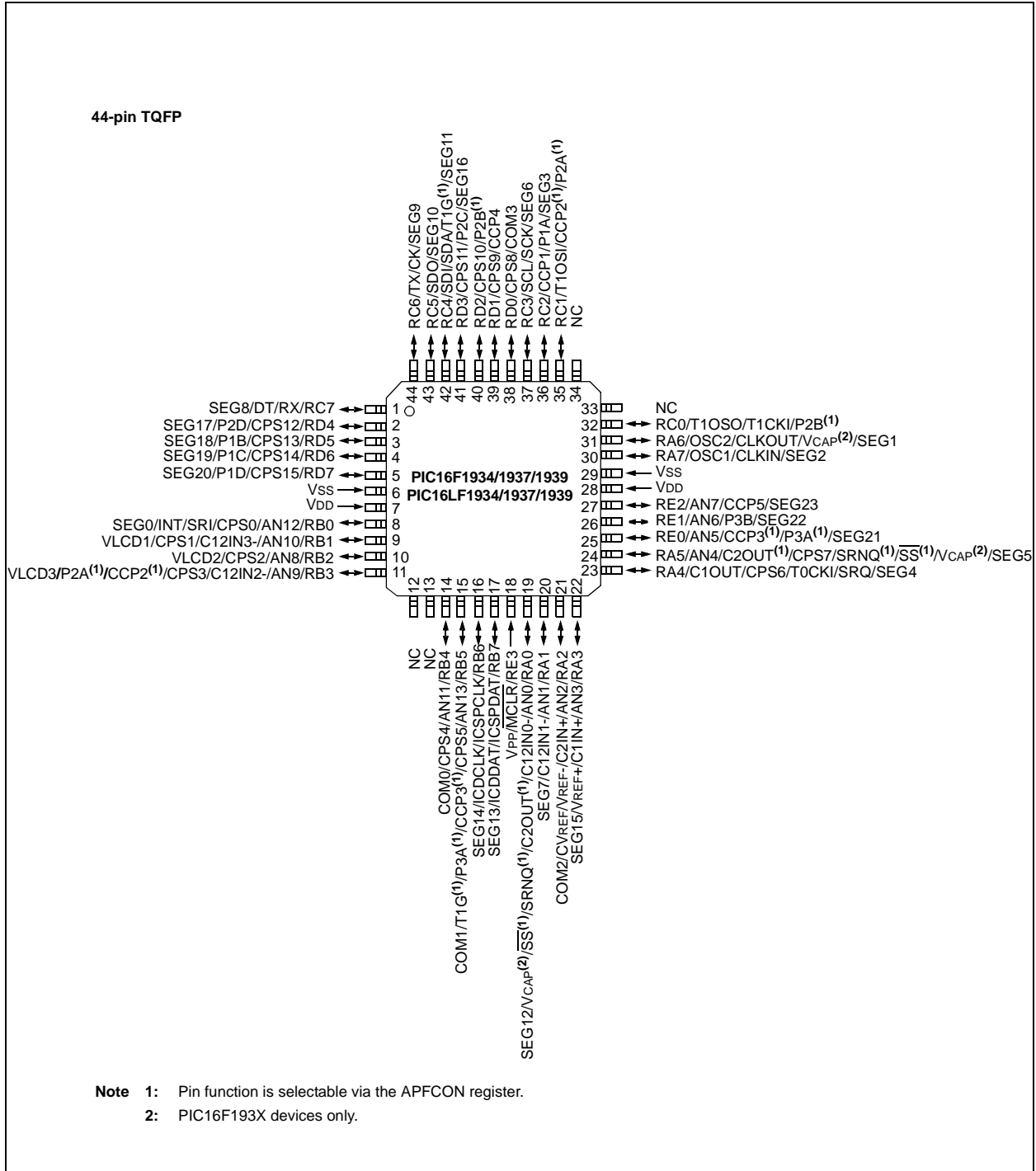
# PIC16F193X/LF193X

## Pin Diagram – 44-Pin QFN (PIC16F1934/1937/1939, PIC16LF1934/1937/1939)



# PIC16F193X/LF193X

## Pin Diagram – 44-Pin TQFP (PIC16F1934/1937/1939, PIC16LF1934/1937/1939)



# PIC16F193X/LF193X

**TABLE 2: 40/44-PIN SUMMARY(PIC16F1934/1937/1939, PIC16LF1934/1937/1939)**

I/O	40-Pin PDIP	44-Pin TQFP	44-Pin QFN	A/D	Cap Sense	Comparator	SR Latch	Timers	CCP	EUSART	MSSP	LCD	Interrupt	Pull-up	Basic
RA0	2	19	19	AN0	—	C12IN0-/C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG12	—	—	VCAP
RA1	3	20	20	AN1	—	C12IN1-	—	—	—	—	—	SEG7	—	—	—
RA2	4	21	21	AN2/VREF-	—	C2IN+/CVREF	—	—	—	—	—	COM2	—	—	—
RA3	5	22	22	AN3/VREF+	—	C1IN+	—	—	—	—	—	SEG15	—	—	—
RA4	6	23	23	—	CPS6	C1OUT	SRQ	T0CKI	—	—	—	SEG4	—	—	—
RA5	7	24	24	AN4	CPS7	C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG5	—	—	VCAP
RA6	14	31	33	—	—	—	—	—	—	—	—	SEG1	—	—	OSC2/CLKOUT VCAP
RA7	13	30	32	—	—	—	—	—	—	—	—	SEG2	—	—	OSC1/CLKIN
RB0	33	8	9	AN12	CPS0	—	SRI	—	—	—	—	SEG0	INT/IO	Y	—
RB1	34	9	10	AN10	CPS1	C12IN3-	—	—	—	—	—	VLCD1	IO	Y	—
RB2	35	10	11	AN8	CPS2	—	—	—	—	—	—	VLCD2	IO	Y	—
RB3	36	11	12	AN9	CPS3	C12IN2-	—	—	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	VLCD3	IO	Y	—
RB4	37	14	14	AN11	CPS4	—	—	—	—	—	—	COM0	IO	Y	—
RB5	38	15	15	AN13	CPS5	—	—	T1G <sup>(1)</sup>	CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	—	—	COM1	IO	Y	—
RB6	39	16	16	—	—	—	—	—	—	—	—	SEG14	IO	Y	ICSPCLK/ ICDCLK
RB7	40	17	17	—	—	—	—	—	—	—	—	SEG13	IO	Y	ICSPDAT/ ICDDAT
RC0	15	32	34	—	—	—	—	T1OSO/ T1CKI	P2B <sup>(1)</sup>	—	—	—	—	—	—
RC1	16	35	35	—	—	—	—	T1OSI	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	—	—	—	—
RC2	17	36	36	—	—	—	—	—	CCP1/ P1A	—	—	SEG3	—	—	—
RC3	18	37	37	—	—	—	—	—	—	—	SCK/SCL	SEG6	—	—	—
RC4	23	42	42	—	—	—	—	T1G <sup>(1)</sup>	—	—	SDI/SDA	SEG11	—	—	—
RC5	24	43	43	—	—	—	—	—	—	—	SDO	SEG10	—	—	—
RC6	25	44	44	—	—	—	—	—	—	TX/CK	—	SEG9	—	—	—
RC7	26	1	1	—	—	—	—	—	—	RX/DT	—	SEG8	—	—	—
RD0	19	38	38	—	CPS8	—	—	—	—	—	—	COM3	—	—	—
RD1	20	39	39	—	CPS9	—	—	—	CCP4	—	—	—	—	—	—
RD2	21	40	40	—	CPS10	—	—	—	P2B <sup>(1)</sup>	—	—	—	—	—	—
RD3	22	41	41	—	CPS11	—	—	—	P2C	—	—	SEG16	—	—	—
RD4	27	2	2	—	CPS12	—	—	—	P2D	—	—	SEG17	—	—	—
RD5	28	3	3	—	CPS13	—	—	—	P1B	—	—	SEG18	—	—	—
RD6	29	4	4	—	CPS14	—	—	—	P1C	—	—	SEG19	—	—	—
RD7	30	5	5	—	CPS15	—	—	—	P1D	—	—	SEG20	—	—	—
RE0	8	25	25	AN5	—	—	—	—	CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	—	—	SEG21	—	—	—
RE1	9	26	26	AN6	—	—	—	—	P3B	—	—	SEG22	—	—	—
RE2	10	27	27	AN7	—	—	—	—	CCP5	—	—	SEG23	—	—	—
RE3	1	18	18	—	—	—	—	—	—	—	—	—	—	Y	MCLR/VPP
VDD	11, 32	7, 28	7, 8, 28	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	12, 31	6, 29	6, 30, 31	—	—	—	—	—	—	—	—	—	—	—	VSS

**Note 1:** Pin functions can be moved using the APFCON register.

# PIC16F193X/LF193X

---

## Table of Contents

1.0	Device Overview .....	13
2.0	Memory Organization .....	19
3.0	Resets .....	55
4.0	Interrupts .....	67
5.0	Low Dropout (LDO) Voltage Regulator .....	79
6.0	I/O Ports .....	81
7.0	Interrupt-on-Change .....	101
8.0	Oscillator Module (With Fail-Safe Clock Monitor).....	105
9.0	SR Latch.....	119
10.0	Device Configuration .....	123
11.0	Analog-to-Digital Converter (ADC) Module .....	129
12.0	Comparator Module.....	141
13.0	Digital-to-Analog Converter (DAC) Module .....	149
14.0	Fixed Voltage Reference (FVR) .....	153
15.0	Timer0 Module .....	155
16.0	Timer1 Module .....	159
17.0	Timer2/4/6 Modules.....	171
18.0	Capacitive Sensing Module.....	175
19.0	Capture/Compare/PWM (ECCP1, ECCP2, ECCP3, CCP4, CCP5) Modules.....	181
20.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART).....	211
21.0	Liquid Crystal Display (LCD) Driver Module.....	239
22.0	Master Synchronous Serial Port (MSSP) Module .....	273
23.0	Data EEPROM and Flash Program Memory Control .....	321
24.0	Power-Down Mode (Sleep) .....	333
25.0	In-Circuit Serial Programming™ (ICSP™) .....	335
26.0	Instruction Set Summary .....	337
27.0	Development Support.....	351
28.0	Electrical Specifications.....	355
29.0	DC and AC Characteristics Graphs and Tables.....	389
30.0	Packaging Information.....	391
	Appendix A: Revision History.....	403
	Appendix B: Device Differences.....	403
	Index .....	405
	The Microchip Web Site.....	413
	Customer Change Notification Service .....	413
	Customer Support.....	413
	Reader Response .....	414
	Product Identification System.....	415

## 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](mailto: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](http://www.microchip.com/cn) to receive the most current information on all of our products.

# PIC16F193X/LF193X

---

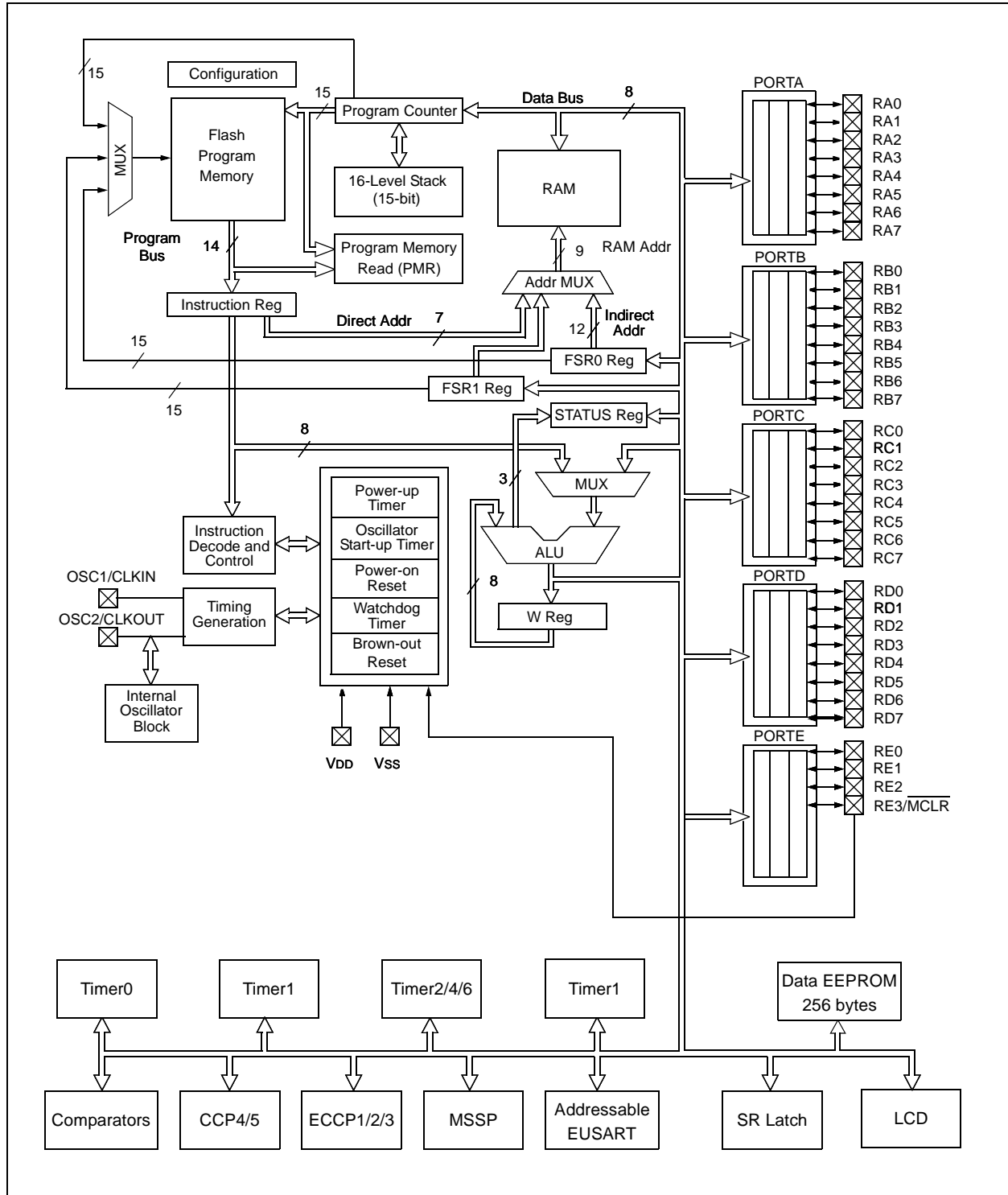
NOTES:

# PIC16F193X/LF193X

## 1.0 DEVICE OVERVIEW

The PIC16F193X/LF193X devices are described within this data sheet. They are available in 28/40/44-pin packages. Figure 1-1 shows a block diagram of the PIC16F193X/LF193X devices. Table 1-1 shows the pinout descriptions.

**FIGURE 1-1: PIC16F193X/LF193X BLOCK DIAGRAM**



# PIC16F193X/LF193X

---

## 1.1 Enhanced Mid-range CPU

PIC16F193X/LF193X devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, indirect, and relative addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 4.5 “Context Saving”**, for more information.

### 1.1.1 16-LEVEL STACK WITH OVERFLOW AND UNDERFLOW RESET

The PIC16F193X/LF193X devices have an external stack memory 15 bits wide and 16 deep. During normal operation, the stack is assumed to be 16 words deep. If enabled, a Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and cause a software Reset. See section **Section 2.4 “Stack”** for more details.

### 1.1.2 FILE SELECT REGISTERS

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one data pointer for all memory. When an FSR points to program memory, there is 1 additional instruction cycle in instructions using INDF to allow the data to be fetched. There are also new instructions to support the FSRs. See **Section 2.5 “Indirect Addressing, INDF and FSR Registers”** for more details.

### 1.1.3 INSTRUCTION SET

There are 48 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 26.0 “Instruction Set Summary”** for more details.



# PIC16F193X/LF193X

**TABLE 1-1: PIC16F193X/LF193X PINOUT DESCRIPTION**

Name	Function	Input Type	Output Type	Description
RA0/AN0/C12IN0-/C2OUT <sup>(1)</sup> /SRNQ <sup>(1)</sup> /SS <sup>(1)</sup> /VCAP <sup>(2)</sup> /SEG12	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	A/D Channel 0 input.
	C12IN0-	AN	—	Comparator C1 or C2 negative input.
	C2OUT	—	CMOS	Comparator C2 output.
	SRNQ	—	CMOS	SR Latch inverting output.
	SS	ST	—	Slave Select input.
	VCAP	Power	Power	Filter capacitor for Voltage Regulator (PIC16F193X only).
RA1/AN1/C12IN1-/SEG7	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	—	A/D Channel 1 input.
	C12IN1-	AN	—	Comparator C1 or C2 negative input.
	SEG7	—	AN	LCD Analog output.
RA2/AN2/C2IN+/VREF-/CVREF/COM2	RA2	TTL	CMOS	General purpose I/O.
	AN2	AN	—	A/D Channel 2 input.
	C2IN+	AN	—	Comparator C2 positive input.
	VREF-	AN	—	A/D Negative Voltage Reference input.
	CVREF	—	AN	Comparator Voltage Reference output.
RA3/AN3/C1IN+/VREF+/COM3 <sup>(3)</sup> /SEG15	RA3	TTL	CMOS	General purpose I/O.
	AN3	AN	—	A/D Channel 3 input.
	C1IN+	AN	—	Comparator C1 positive input.
	VREF+	AN	—	A/D Voltage Reference input.
	COM3 <sup>(3)</sup>	—	AN	LCD Analog output.
	SEG15	—	AN	LCD Analog output.
RA4/C1OUT/CPS6/T0CKI/SRQ/CCP5/SEG4	RA4	TTL	CMOS	General purpose I/O.
	C1OUT	—	CMOS	Comparator C1 output.
	CPS6	AN	—	Capacitive sensing input 6.
	T0CKI	ST	—	Timer0 clock input.
	SRQ	—	CMOS	SR Latch non-inverting output.
	CCP5	ST	CMOS	Capture/Compare/PWM5.
RA5/AN4/C2OUT <sup>(1)</sup> /CPS7/SRNQ <sup>(1)</sup> /SS <sup>(1)</sup> /VCAP <sup>(2)</sup> /SEG5	RA5	TTL	CMOS	General purpose I/O.
	AN4	AN	—	A/D Channel 4 input.
	C2OUT	—	CMOS	Comparator C2 output.
	CPS7	AN	—	Capacitive sensing input 7.
	SRNQ	—	CMOS	SR Latch inverting output.
	SS	ST	—	Slave Select input.
	VCAP	Power	Power	Filter capacitor for Voltage Regulator (PIC16F193X only).
	SEG5	—	AN	LCD Analog output.

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OD = Open Drain  
TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels    I<sup>2</sup>C™ = Schmitt Trigger input with I<sup>2</sup>C levels  
HV = High Voltage    XTAL = Crystal

- Note** 1: Pin function is selectable via the APFCON register.  
2: PIC16F193X devices only.  
3: PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices only.  
4: PORTD is available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.  
5: RE<2:0> are available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only

# PIC16F193X/LF193X

**TABLE 1-1: PIC16F193X/LF193X PINOUT DESCRIPTION (CONTINUED)**

Name	Function	Input Type	Output Type	Description
RA6/OSC2/CLKOUT/VCAP <sup>(2)</sup> /SEG1	RA6	TTL	CMOS	General purpose I/O.
	OSC2	—	XTAL	Crystal/Resonator (LP, XT, HS modes).
	CLKOUT	—	CMOS	Fosc/4 output.
	VCAP	Power	Power	Filter capacitor for Voltage Regulator (PIC16F193X only).
	SEG1	—	AN	LCD Analog output.
RA7/OSC1/CLKIN/SEG2	RA7	TTL	CMOS	General purpose I/O.
	OSC1	XTAL	—	Crystal/Resonator (LP, XT, HS modes).
	CLKIN	CMOS	—	External clock input (EC mode).
	SEG2	—	AN	LCD Analog output.
RB0/AN12/CPS0/CCP4/SRI/INT/SEG0	RB0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN12	AN	—	A/D Channel 12 input.
	CPS0	AN	—	Capacitive sensing input 0.
	CCP4	ST	CMOS	Capture/Compare/PWM4.
	SRI	—	ST	SR Latch input.
	INT	ST	—	External interrupt.
	SEG0	—	AN	LCD analog output.
RB1/AN10/C12IN3-/CPS1/P1C/VLCD1	RB1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN10	AN	—	A/D Channel 10 input.
	C12IN3-	AN	—	Comparator C1 or C2 negative input.
	CPS1	AN	—	Capacitive sensing input 1.
	P1C	—	CMOS	PWM output.
	VLCD1	AN	—	LCD analog input.
RB2/AN8/CPS2/P1B/VLCD2	RB2	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN8	AN	—	A/D Channel 8 input.
	CPS2	AN	—	Capacitive sensing input 2.
	P1B	—	CMOS	PWM output.
	VLCD2	AN	—	LCD analog input.
RB3/AN9/C12IN2-/CPS3/CCP2 <sup>(1)</sup> /P2A <sup>(1)</sup> /VLCD3	RB3	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN9	AN	—	A/D Channel 9 input.
	C12IN2-	AN	—	Comparator C1 or C2 negative input.
	CPS3	AN	—	Capacitive sensing input 3.
	CCP2	ST	CMOS	Capture/Compare/PWM2.
	P2A	—	CMOS	PWM output.
VLCD3	AN	—	LCD analog input.	

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OD = Open Drain  
TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels    I<sup>2</sup>C™ = Schmitt Trigger input with I<sup>2</sup>C levels  
HV = High Voltage    XTAL = Crystal

- Note** 1: Pin function is selectable via the APFCON register.  
2: PIC16F193X devices only.  
3: PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices only.  
4: PORTD is available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.  
5: RE<2:0> are available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only

# PIC16F193X/LF193X

**TABLE 1-1: PIC16F193X/LF193X PINOUT DESCRIPTION (CONTINUED)**

Name	Function	Input Type	Output Type	Description
RB4/AN11/CPS4/P1D/COM0	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN11	AN	—	A/D Channel 11 input.
	CPS4	AN	—	Capacitive sensing input 4.
	P1D	—	CMOS	PWM output.
	COM0	—	AN	LCD Analog output.
RB5/AN13/CPS5/P2B/CCP3 <sup>(1)</sup> /P3A <sup>(1)</sup> /T1G <sup>(1)</sup> /COM1	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN13	AN	—	A/D Channel 13 input.
	CPS5	AN	—	Capacitive sensing input 5.
	P2B	—	CMOS	PWM output.
	CCP3	ST	CMOS	Capture/Compare/PWM3.
	P3A	—	CMOS	PWM output.
	T1G	ST	—	Timer1 Gate input.
RB6/ICSPCLK/ICDCLK/SEG14	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPCLK	ST	—	Serial Programming Clock.
	ICDCLK	ST	—	In-Circuit Debug Clock.
	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	In-Circuit Data I/O.
	SEG13	—	AN	LCD Analog output.
RC0/T1OSO/T1CKI/P2B <sup>(1)</sup>	RC0	ST	CMOS	General purpose I/O.
	T1OSO	XTAL	XTAL	Timer1 oscillator connection.
	T1CKI	ST	—	Timer1 clock input.
	P2B	—	CMOS	PWM output.
RC1/T1OSI/CCP2 <sup>(1)</sup> /P2A <sup>(1)</sup>	RC1	ST	CMOS	General purpose I/O.
	T1OSI	XTAL	XTAL	Timer1 oscillator connection.
	CCP2	ST	CMOS	Capture/Compare/PWM2.
	P2A	—	CMOS	PWM output.
RC2/CCP1/P1A/SEG3	RC2	ST	CMOS	General purpose I/O.
	CCP1	ST	CMOS	Capture/Compare/PWM1.
	P1A	—	CMOS	PWM output.
	SEG3	—	AN	LCD Analog output.
RC3/SCK/SCL/SEG6	RC3	ST	CMOS	General purpose I/O.
	SCK	ST	CMOS	SPI clock.
	SCL	I <sup>2</sup> C	OD	I <sup>2</sup> C™ clock.
	SEG6	—	AN	LCD Analog output.

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OD = Open Drain  
TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels    I<sup>2</sup>C™ = Schmitt Trigger input with I<sup>2</sup>C levels  
HV = High Voltage    XTAL = Crystal

- Note 1:** Pin function is selectable via the APFCON register.  
**2:** PIC16F193X devices only.  
**3:** PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices only.  
**4:** PORTD is available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.  
**5:** RE<2:0> are available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only

# PIC16F193X/LF193X

**TABLE 1-1: PIC16F193X/LF193X PINOUT DESCRIPTION (CONTINUED)**

Name	Function	Input Type	Output Type	Description
RC4/SDI/SDA/T1G <sup>(1)</sup> /SEG11	RC4	ST	CMOS	General purpose I/O.
	SDI	ST	—	SPI data input.
	SDA	I <sup>2</sup> C	OD	I <sup>2</sup> C™ data input/output.
	T1G	ST	—	Timer1 Gate input.
	SEG11	—	AN	LCD Analog output.
RC5/SDO/SEG10	RC5	ST	CMOS	General purpose I/O.
	SDO	—	CMOS	SPI data output.
	SEG10	—	AN	LCD Analog output.
RC6/TX/CK/CCP3/P3A/SEG9	RC6	ST	CMOS	General purpose I/O.
	TX	—	CMOS	USART asynchronous transmit.
	CK	ST	CMOS	USART synchronous clock.
	CCP3	ST	CMOS	Capture/Compare/PWM3.
	P3A	—	CMOS	PWM output.
	SEG9	—	AN	LCD Analog output.
RC7/RX/DT/P3B/SEG8	RC7	ST	CMOS	General purpose I/O.
	RX	ST	—	USART asynchronous input.
	DT	ST	CMOS	USART synchronous data.
	P3B	—	CMOS	PWM output.
	SEG8	—	AN	LCD Analog output.
RD0 <sup>(4)</sup> /CPS8/COM3	RD0	ST	CMOS	General purpose I/O.
	CPS8	AN	—	Capacitive sensing input 8.
	COM3	—	AN	LCD analog output.
RD1 <sup>(4)</sup> /CPS9/CCP4	RD1	ST	CMOS	General purpose I/O.
	CPS9	AN	—	Capacitive sensing input 9.
	CCP4	ST	CMOS	Capture/Compare/PWM4.
RD2 <sup>(4)</sup> /CPS10/P2B	RD2	ST	CMOS	General purpose I/O.
	CPS10	AN	—	Capacitive sensing input 10.
	P2B	—	CMOS	PWM output.
RD3 <sup>(4)</sup> /CPS11/P2C/SEG16	RD3	ST	CMOS	General purpose I/O.
	CPS11	AN	—	Capacitive sensing input 11.
	P2C	—	CMOS	PWM output.
	SEG16	—	AN	LCD analog output.
RD4 <sup>(4)</sup> /CPS12/P2D/SEG17	RD4	ST	CMOS	General purpose I/O.
	CPS12	AN	—	Capacitive sensing input 12.
	P2D	—	CMOS	PWM output.
	SEG17	—	AN	LCD analog output.
RD5 <sup>(4)</sup> /CPS13/P1B/SEG18	RD5	ST	CMOS	General purpose I/O.
	CPS13	AN	—	Capacitive sensing input 13.
	P1D	—	CMOS	PWM output.
	SEG18	—	AN	LCD analog output.

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OD = Open Drain  
TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels    I<sup>2</sup>C™ = Schmitt Trigger input with I<sup>2</sup>C levels  
HV = High Voltage    XTAL = Crystal

- Note** 1: Pin function is selectable via the APFCON register.  
2: PIC16F193X devices only.  
3: PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices only.  
4: PORTD is available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.  
5: RE<2:0> are available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only

# PIC16F193X/LF193X

**TABLE 1-1: PIC16F193X/LF193X PINOUT DESCRIPTION (CONTINUED)**

Name	Function	Input Type	Output Type	Description
RD6 <sup>(4)</sup> /CPS14/P1C/SEG19	RD6	ST	CMOS	General purpose I/O.
	CPS14	AN	—	Capacitive sensing input 14.
	P1C	—	CMOS	PWM output.
	SEG19	—	AN	LCD analog output.
RD7 <sup>(4)</sup> /CPS15/P1D/SEG20	RD7	ST	CMOS	General purpose I/O.
	CPS15	AN	—	Capacitive sensing input 15.
	P1D	—	CMOS	PWM output.
	SEG20	—	AN	LCD analog output.
RE0 <sup>(5)</sup> /AN5/P3A <sup>(1)</sup> /CCP3 <sup>(1)</sup> /SEG21	RE0	ST	CMOS	General purpose I/O.
	AN5	AN	—	A/D Channel 5 input.
	P3A	—	CMOS	PWM output.
	CCP3	ST	CMOS	Capture/Compare/PWM3.
	SEG21	—	AN	LCD analog output.
RE1 <sup>(5)</sup> /AN6/P3B/SEG22	RE1	ST	CMOS	General purpose I/O.
	AN6	AN	—	A/D Channel 6 input.
	P3B	—	CMOS	PWM output.
	SEG22	—	AN	LCD analog output.
RE2 <sup>(5)</sup> /AN7/CCP5/SEG23	RE2	ST	CMOS	General purpose I/O.
	AN7	AN	—	A/D Channel 7 input.
	CCP5	ST	CMOS	Capture/Compare/PWM5.
	SEG23	—	AN	LCD analog output.
RE3/MCLR/VPP	RE3	TTL	—	General purpose input.
	MCLR	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OD = Open Drain  
TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels    I<sup>2</sup>C™ = Schmitt Trigger input with I<sup>2</sup>C levels  
HV = High Voltage    XTAL = Crystal

- Note** 1: Pin function is selectable via the APFCON register.  
2: PIC16F193X devices only.  
3: PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices only.  
4: PORTD is available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.  
5: RE<2:0> are available on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only

# PIC16F193X/LF193X

---

NOTES:

## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

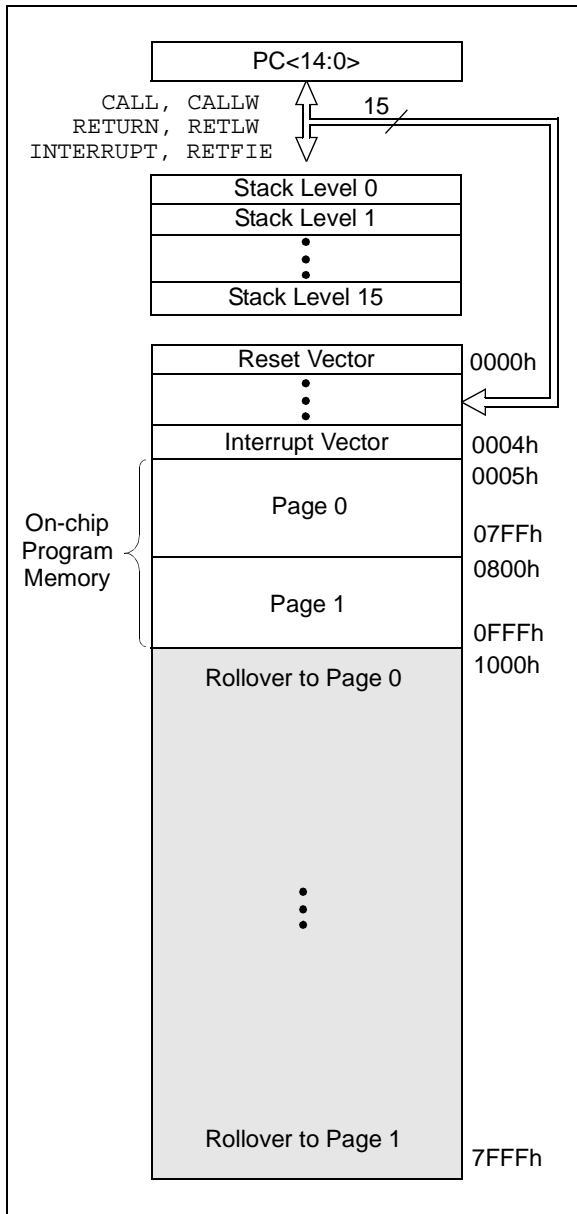
The enhanced mid-range core has a 15-bit program counter capable of addressing 32K x 14 program memory space. Table 2-1 shows the memory sizes implemented for the PIC16F193X/LF193X device family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 2-1, 2-2 and 2-3).

**TABLE 2-1: DEVICE SIZES AND ADDRESSES**

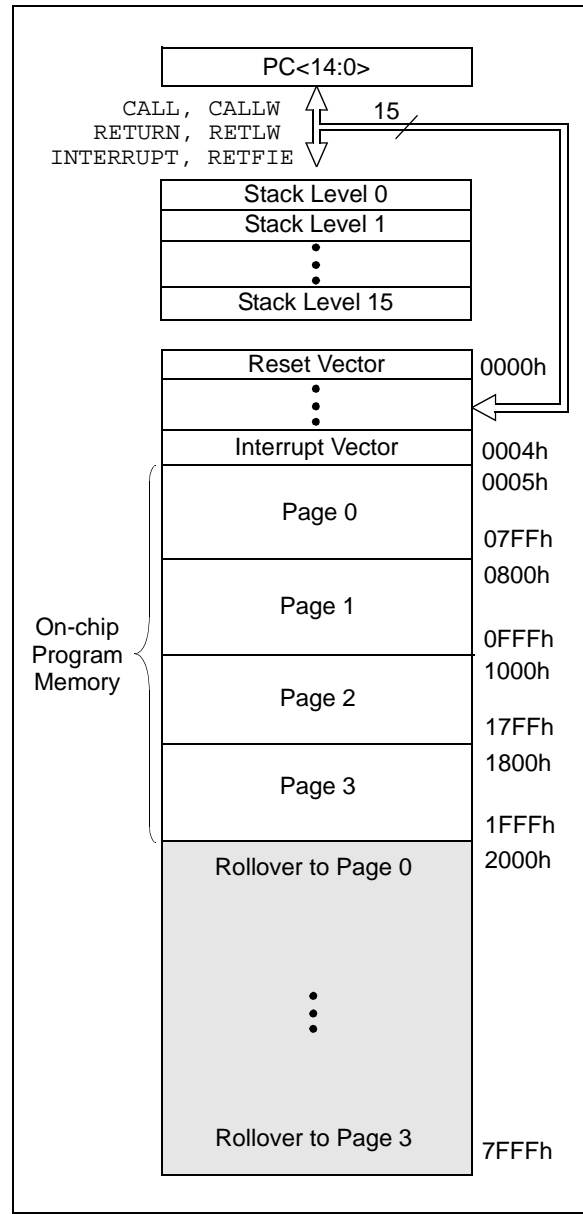
Device	Program Memory Space (Words)	Last Program Memory Address
PIC16F1933/PIC16LF1933	4,096	0FFFh
PIC16F1934/PIC16LF1934	4,096	0FFFh
PIC16F1936/PIC16LF1936	8,192	1FFFh
PIC16F1937/PIC16LF1937	8,192	1FFFh
PIC16F1938/PIC16LF1938	16,384	3FFFh
PIC16F1939/PIC16LF1939	16,384	3FFFh

# PIC16F193X/LF193X

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F1933/PIC16LF1933/PIC16F1934/PIC16LF1934**

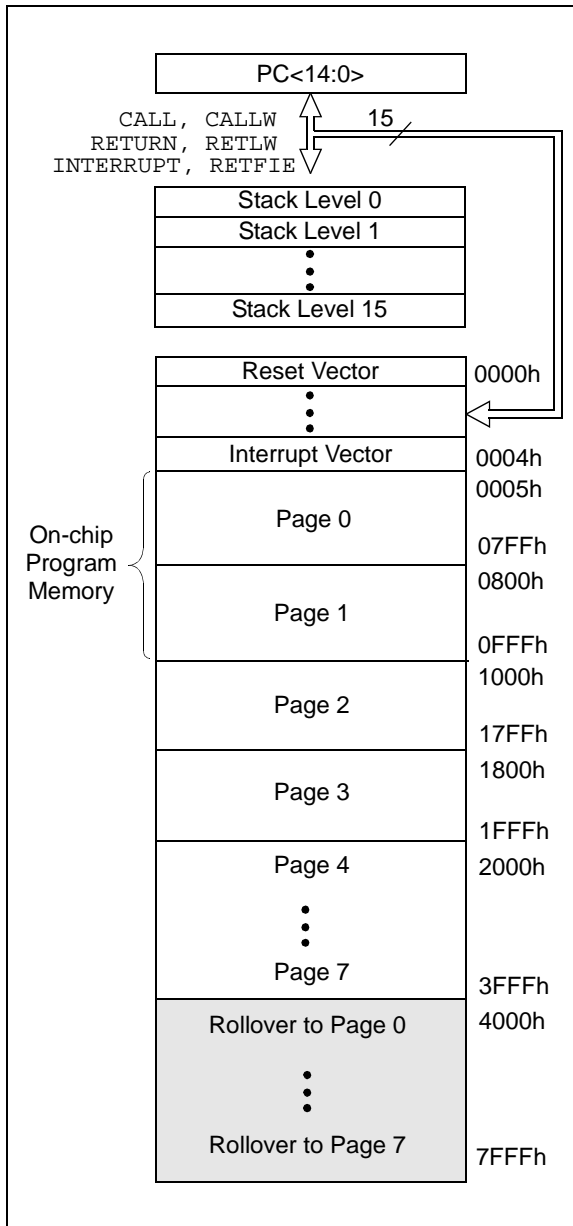


**FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F1936/PIC16LF1936/PIC16F1937/PIC16LF1937**





**FIGURE 2-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16F1938/PIC16LF1938/ PIC16F1939/PIC16LF1939**



## 2.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

### 2.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 2-1.

#### EXAMPLE 2-1: RETLW INSTRUCTION

```
constants
    brw
    retlw DATA1
    retlw DATA2
    retlw DATA3
    retlw DATA4

my_function
    ;... LOTS OF CODE...
    movlw    DATA_INDEX
    call constants
    ;... THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

### 2.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower 8 bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 2-2 demonstrates accessing the program memory via an FSR.

#### EXAMPLE 2-2: ACCESSING PROGRAM MEMORY VIA FSR

```
bsf FSR1H,7
moviw 0[INDF1]
;THE PROGRAM MEMORY IS IN W
```

# PIC16F193X/LF193X

---

## 2.2 Data Memory Organization

The data memory is partitioned in up to 32 memory banks with up to 128 bytes in a bank. Each bank consists of 12 core registers, 20 Special Function Registers (SFR), 16 common registers, and up to 80 bytes of General Purpose Registers (GPR). The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All banks contain the core SFRs and common registers. Unimplemented SFRs or GPRs will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 2.5 "Indirect Addressing, INDF and FSR Registers"** for more information.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The general purpose register file is an 8-bit RAM memory for use by your application. There are up to 80 bytes of GPR in each data memory bank.

### 2.2.2 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in the following sections. The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

TABLE 2-2: PIC16F1933/1934 MEMORY MAP, BANKS 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	—	28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	PORTD <sup>(1)</sup>	08Fh	TRISD <sup>(1)</sup>	10Fh	LATD <sup>(1)</sup>	18Fh	ANSEL <sup>(1)</sup>	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	LATE <sup>(1)</sup>	190h	ANSELE <sup>(1)</sup>	210h	WPUE	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADR <sub>L</sub>	211h	SSPBUF	291h	CCPR1L	311h	CCPR3L	391h	—
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADR <sub>H</sub>	212h	SSPADD	292h	CCPR1H	312h	CCPR3H	392h	—
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	EEDATL	213h	SSPMSK	293h	CCP1CON	313h	CCP3CON	393h	—
014h	—	094h	—	114h	CM2CON1	194h	EEDATH	214h	SSPSTAT	294h	PWM1CON	314h	PWM3CON	394h	IOCBP
015h	TMR0	095h	OPTION	115h	CMOUT	195h	EECON1	215h	SSPCON1	295h	CCP1AS	315h	CCP3AS	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSPCON2	296h	PSTR1CON	316h	PSTR3CON	396h	IOCBF
017h	TMR1H	097h	WDTCN	117h	FVRCON	197h	—	217h	SSPCON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	CCPR2L	318h	CCPR4L	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	CCPR2H	319h	CCPR4H	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TXREG	21Ah	—	29Ah	CCP2CON	31Ah	CCP4CON	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SPBRGL	21Bh	—	29Bh	PWM2CON	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	CCP2AS	31Ch	CCPR5L	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	PSTR2CON	31Dh	CCPR5H	39Dh	—
01Eh	CPSCON0	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	CCPTMRS0	31Eh	CCP5CON	39Eh	—
01Fh	CPSCON1	09Fh	—	11Fh	—	19Fh	BAUDCTR	21Fh	—	29Fh	CCPTMRS1	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	Unimplemented Read as '0'	220h	Unimplemented Read as '0'	2A0h	Unimplemented Read as '0'	320h	Unimplemented Read as '0'	3A0h	Unimplemented Read as '0'
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h		170h		1F0h		270h		2F0h		370h		3F0h	
07Fh	0FFh	17Fh	1FFh	27Fh	2FFh	37Fh	3FFh								

Legend: ■ = Unimplemented data memory locations, read as '0'.

Note 1: Not available on PIC16F1933/1936/1938/PIC16LF1933/1936/1938.

TABLE 2-3: PIC16F1933/1934 MEMORY MAP, BANKS 8-15

	BANK 8	BANK 9	BANK 10	BANK 11	BANK 12	BANK 13	BANK 14	BANK 15							
400h	INDF0	480h	INDF0	500h	INDF0	580h	INDF0	600h	INDF0	680h	INDF0	700h	INDF0	780h	INDF0
401h	INDF1	481h	INDF1	501h	INDF1	581h	INDF1	601h	INDF1	681h	INDF1	701h	INDF1	781h	INDF1
402h	PCL	482h	PCL	502h	PCL	582h	PCL	602h	PCL	682h	PCL	702h	PCL	782h	PCL
403h	STATUS	483h	STATUS	503h	STATUS	583h	STATUS	603h	STATUS	683h	STATUS	703h	STATUS	783h	STATUS
404h	FSR0L	484h	FSR0L	504h	FSR0L	584h	FSR0L	604h	FSR0L	684h	FSR0L	704h	FSR0L	784h	FSR0L
405h	FSR0H	485h	FSR0H	505h	FSR0H	585h	FSR0H	605h	FSR0H	685h	FSR0H	705h	FSR0H	785h	FSR0H
406h	FSR1L	486h	FSR1L	506h	FSR1L	586h	FSR1L	606h	FSR1L	686h	FSR1L	706h	FSR1L	786h	FSR1L
407h	FSR1H	487h	FSR1H	507h	FSR1H	587h	FSR1H	607h	FSR1H	687h	FSR1H	707h	FSR1H	787h	FSR1H
408h	BSR	488h	BSR	508h	BSR	588h	BSR	608h	BSR	688h	BSR	708h	BSR	788h	BSR
409h	WREG	489h	WREG	509h	WREG	589h	WREG	609h	WREG	689h	WREG	709h	WREG	789h	WREG
40Ah	PCLATH	48Ah	PCLATH	50Ah	PCLATH	58Ah	PCLATH	60Ah	PCLATH	68Ah	PCLATH	70Ah	PCLATH	78Ah	PCLATH
40Bh	INTCON	48Bh	INTCON	50Bh	INTCON	58Bh	INTCON	60Bh	INTCON	68Bh	INTCON	70Bh	INTCON	78Bh	INTCON
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	—	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	—	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	—	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	—	714h	—	794h	—
415h	TMR4	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—
416h	PR4	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	TMR6	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	PR6	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	T6CON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	Unimplemented Read as '0'	4A0h	Unimplemented Read as '0'	520h	Unimplemented Read as '0'	5A0h	Unimplemented Read as '0'	620h	Unimplemented Read as '0'	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
46Fh	—	4EFh	—	56Fh	—	5EFh	—	66Fh	—	6EFh	—	76Fh	—	7EFh	—
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—

Legend:  = Unimplemented data memory locations, read as '0'.

See Table 2-10 or  
Table 2-11

TABLE 2-4: PIC16F1936/1937 MEMORY MAP, BANKS 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	—	28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	PORTD <sup>(1)</sup>	08Fh	TRISD <sup>(1)</sup>	10Fh	LATD <sup>(1)</sup>	18Fh	ANSELD <sup>(1)</sup>	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	LATE <sup>(1)</sup>	190h	ANSELE <sup>(1)</sup>	210h	WPUE	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADRL	211h	SSPBUF	291h	CCPR1L	311h	CCPR3L	391h	—
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADRH	212h	SSPADD	292h	CCPR1H	312h	CCPR3H	392h	—
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	EEDATL	213h	SSPMSK	293h	CCP1CON	313h	CCP3CON	393h	—
014h	—	094h	—	114h	CM2CON1	194h	EEDATH	214h	SSPSTAT	294h	PWM1CON	314h	PWM3CON	394h	IOCBP
015h	TMR0	095h	OPTION	115h	CMOUT	195h	EECON1	215h	SSPCON1	295h	CCP1AS	315h	CCP3AS	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSPCON2	296h	PSTR1CON	316h	PSTR3CON	396h	IOCBF
017h	TMR1H	097h	WDTCN	117h	FVRCON	197h	—	217h	SSPCON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	CCPR2L	318h	CCPR4L	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	CCPR2H	319h	CCPR4H	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TXREG	21Ah	—	29Ah	CCP2CON	31Ah	CCP4CON	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SPBRGL	21Bh	—	29Bh	PWM2CON	31Bh	—	39Bh	—
01Ch	TxCON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	CCP2AS	31Ch	CCPR5L	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	PSTR2CON	31Dh	CCPR5H	39Dh	—
01Eh	CPSCON0	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	CCPTMRS0	31Eh	CCP5CON	39Eh	—
01Fh	CPSCON1	09Fh	—	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	CCPTMRS1	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 16 Bytes	3A0h	Unimplemented Read as '0'
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh	Unimplemented Read as '0'	3EFh	
070h		0F0h		170h		1F0h		270h		2F0h		370h	Accesses 70h – 7Fh	3F0h	
07Fh	0FFh	17Fh	1FFh	27Fh	2FFh	37Fh	Accesses 70h – 7Fh	3FFh	Accesses 70h – 7Fh						

Legend: ■ = Unimplemented data memory locations, read as '0'.

Note 1: Not available on PIC16F1933/1936/1938/PIC16LF1933/1936/1938.

TABLE 2-5: PIC16F1936/1937 MEMORY MAP, BANKS 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	INDF0	480h	INDF0	500h	INDF0	580h	INDF0	600h	INDF0	680h	INDF0	700h	INDF0	780h	INDF0
401h	INDF1	481h	INDF1	501h	INDF1	581h	INDF1	601h	INDF1	681h	INDF1	701h	INDF1	781h	INDF1
402h	PCL	482h	PCL	502h	PCL	582h	PCL	602h	PCL	682h	PCL	702h	PCL	782h	PCL
403h	STATUS	483h	STATUS	503h	STATUS	583h	STATUS	603h	STATUS	683h	STATUS	703h	STATUS	783h	STATUS
404h	FSR0L	484h	FSR0L	504h	FSR0L	584h	FSR0L	604h	FSR0L	684h	FSR0L	704h	FSR0L	784h	FSR0L
405h	FSR0H	485h	FSR0H	505h	FSR0H	585h	FSR0H	605h	FSR0H	685h	FSR0H	705h	FSR0H	785h	FSR0H
406h	FSR1L	486h	FSR1L	506h	FSR1L	586h	FSR1L	606h	FSR1L	686h	FSR1L	706h	FSR1L	786h	FSR1L
407h	FSR1H	487h	FSR1H	507h	FSR1H	587h	FSR1H	607h	FSR1H	687h	FSR1H	707h	FSR1H	787h	FSR1H
408h	BSR	488h	BSR	508h	BSR	588h	BSR	608h	BSR	688h	BSR	708h	BSR	788h	BSR
409h	WREG	489h	WREG	509h	WREG	589h	WREG	609h	WREG	689h	WREG	709h	WREG	789h	WREG
40Ah	PCLATH	48Ah	PCLATH	50Ah	PCLATH	58Ah	PCLATH	60Ah	PCLATH	68Ah	PCLATH	70Ah	PCLATH	78Ah	PCLATH
40Bh	INTCON	48Bh	INTCON	50Bh	INTCON	58Bh	INTCON	60Bh	INTCON	68Bh	INTCON	70Bh	INTCON	78Bh	INTCON
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	—	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	—	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	—	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	—	714h	—	794h	—
415h	TMR4	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—
416h	PR4	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	TMR6	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	PR6	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	T6CON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	Unimplemented Read as '0'	4A0h	Unimplemented Read as '0'	520h	Unimplemented Read as '0'	5A0h	Unimplemented Read as '0'	620h	Unimplemented Read as '0'	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
46Fh	—	4EFh	—	56Fh	—	5EFh	—	66Fh	—	6EFh	—	76Fh	—	7EFh	—
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—

See Table 2-10 or  
Table 2-11Legend:  = Unimplemented data memory locations, read as '0'.

TABLE 2-6: PIC16F1938/1939 MEMORY MAP, BANKS 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	—	28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	PORTD <sup>(1)</sup>	08Fh	TRISD <sup>(1)</sup>	10Fh	LATD <sup>(1)</sup>	18Fh	ANSELD <sup>(1)</sup>	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	LATE <sup>(1)</sup>	190h	ANSELE <sup>(1)</sup>	210h	WPUE	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADRL	211h	SSPBUF	291h	CCPR1L	311h	CCPR3L	391h	—
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADRH	212h	SSPADD	292h	CCPR1H	312h	CCPR3H	392h	—
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	EEDATL	213h	SSPMSK	293h	CCP1CON	313h	CCP3CON	393h	—
014h	—	094h	—	114h	CM2CON1	194h	EEDATH	214h	SSPSTAT	294h	PWM1CON	314h	PWM3CON	394h	IOCBP
015h	TMR0	095h	OPTION	115h	CMOUT	195h	EECON1	215h	SSPCON1	295h	CCP1AS	315h	CCP3AS	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSPCON2	296h	PSTR1CON	316h	PSTR3CON	396h	IOCBF
017h	TMR1H	097h	WDTCN	117h	FVRCON	197h	—	217h	SSPCON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	CCPR2L	318h	CCPR4L	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	CCPR4H	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	CCP4CON	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SPBRGL1	21Bh	—	29Bh	PWM2CON	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH1	21Ch	—	29Ch	CCP2AS	31Ch	CCPR5L	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA1	21Dh	—	29Dh	PSTR2CON	31Dh	CCPR5H	39Dh	—
01Eh	CPSCON0	09Eh	ADCON1	11Eh	—	19Eh	TXSTA1	21Eh	—	29Eh	CCPTMRS0	31Eh	CCP5CON	39Eh	—
01Fh	CPSCON1	09Fh	—	11Fh	—	19Fh	BAUDCTL1	21Fh	—	29Fh	CCPTMRS1	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 80 Bytes	3A0h	General Purpose Register 80 Bytes
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h		170h		1F0h		270h		2F0h		370h		3F0h	
07Fh		0FFh	Accesses 70h – 7Fh	17Fh	Accesses 70h – 7Fh	1FFh	Accesses 70h – 7Fh	27Fh	Accesses 70h – 7Fh	2FFh	Accesses 70h – 7Fh	37Fh	Accesses 70h – 7Fh	3FFh	Accesses 70h – 7Fh

Legend: ■ = Unimplemented data memory locations, read as '0'.

Note 1: Not available on PIC16F1933/1936/1938/PIC16LF1933/1936/1938.

**TABLE 2-7: PIC16F1938/1939 MEMORY MAP, BANKS 8-15**

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	INDF0	480h	INDF0	500h	INDF0	580h	INDF0	600h	INDF0	680h	INDF0	700h	INDF0	780h	INDF0
401h	INDF1	481h	INDF1	501h	INDF1	581h	INDF1	601h	INDF1	681h	INDF1	701h	INDF1	781h	INDF1
402h	PCL	482h	PCL	502h	PCL	582h	PCL	602h	PCL	682h	PCL	702h	PCL	782h	PCL
403h	STATUS	483h	STATUS	503h	STATUS	583h	STATUS	603h	STATUS	683h	STATUS	703h	STATUS	783h	STATUS
404h	FSR0L	484h	FSR0L	504h	FSR0L	584h	FSR0L	604h	FSR0L	684h	FSR0L	704h	FSR0L	784h	FSR0L
405h	FSR0H	485h	FSR0H	505h	FSR0H	585h	FSR0H	605h	FSR0H	685h	FSR0H	705h	FSR0H	785h	FSR0H
406h	FSR1L	486h	FSR1L	506h	FSR1L	586h	FSR1L	606h	FSR1L	686h	FSR1L	706h	FSR1L	786h	FSR1L
407h	FSR1H	487h	FSR1H	507h	FSR1H	587h	FSR1H	607h	FSR1H	687h	FSR1H	707h	FSR1H	787h	FSR1H
408h	BSR	488h	BSR	508h	BSR	588h	BSR	608h	BSR	688h	BSR	708h	BSR	788h	BSR
409h	WREG	489h	WREG	509h	WREG	589h	WREG	609h	WREG	689h	WREG	709h	WREG	789h	WREG
40Ah	PCLATH	48Ah	PCLATH	50Ah	PCLATH	58Ah	PCLATH	60Ah	PCLATH	68Ah	PCLATH	70Ah	PCLATH	78Ah	PCLATH
40Bh	INTCON	48Bh	INTCON	50Bh	INTCON	58Bh	INTCON	60Bh	INTCON	68Bh	INTCON	70Bh	INTCON	78Bh	INTCON
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	—	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	—	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	—	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	—	714h	—	794h	—
415h	TMR4	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—
416h	PR4	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	TMR6	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	PR6	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	T6CON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	General Purpose Register 80 Bytes	4A0h	General Purpose Register 80 Bytes	520h	General Purpose Register 80 Bytes	5A0h	General Purpose Register 80 Bytes	620h	General Purpose Register 48 Bytes	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
												Unimplemented Read as '0'			
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh		7EFh	
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh	

**Legend:** ■ = Unimplemented data memory locations, read as '0'.

See Table 2-10 or Table 2-11



TABLE 2-8: PIC16F193X/LF193X MEMORY MAP, BANKS 16-23

BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	INDF0	880h	INDF0	900h	INDF0	980h	INDF0	A00h	INDF0	A80h	INDF0	B00h	INDF0	B80h	INDF0
801h	INDF1	881h	INDF1	901h	INDF1	981h	INDF1	A01h	INDF1	A81h	INDF1	B01h	INDF1	B81h	INDF1
802h	PCL	882h	PCL	902h	PCL	982h	PCL	A02h	PCL	A82h	PCL	B02h	PCL	B82h	PCL
803h	STATUS	883h	STATUS	903h	STATUS	983h	STATUS	A03h	STATUS	A83h	STATUS	B03h	STATUS	B83h	STATUS
804h	FSR0L	884h	FSR0L	904h	FSR0L	984h	FSR0L	A04h	FSR0L	A84h	FSR0L	B04h	FSR0L	B84h	FSR0L
805h	FSR0H	885h	FSR0H	905h	FSR0H	985h	FSR0H	A05h	FSR0H	A85h	FSR0H	B05h	FSR0H	B85h	FSR0H
806h	FSR1L	886h	FSR1L	906h	FSR1L	986h	FSR1L	A06h	FSR1L	A86h	FSR1L	B06h	FSR1L	B86h	FSR1L
807h	FSR1H	887h	FSR1H	907h	FSR1H	987h	FSR1H	A07h	FSR1H	A87h	FSR1H	B07h	FSR1H	B87h	FSR1H
808h	BSR	888h	BSR	908h	BSR	988h	BSR	A08h	BSR	A88h	BSR	B08h	BSR	B88h	BSR
809h	WREG	889h	WREG	909h	WREG	989h	WREG	A09h	WREG	A89h	WREG	B09h	WREG	B89h	WREG
80Ah	PCLATH	88Ah	PCLATH	90Ah	PCLATH	98Ah	PCLATH	A0Ah	PCLATH	A8Ah	PCLATH	B0Ah	PCLATH	B8Ah	PCLATH
80Bh	INTCON	88Bh	INTCON	90Bh	INTCON	98Bh	INTCON	A0Bh	INTCON	A8Bh	INTCON	B0Bh	INTCON	B8Bh	INTCON
80Ch	—	88Ch	—	90Ch	—	98Ch	—	A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
80Dh	—	88Dh	—	90Dh	—	98Dh	—	A0Dh	—	A8Dh	—	B0Dh	—	B8Dh	—
80Eh	—	88Eh	—	90Eh	—	98Eh	—	A0Eh	—	A8Eh	—	B0Eh	—	B8Eh	—
80Fh	—	88Fh	—	90Fh	—	98Fh	—	A0Fh	—	A8Fh	—	B0Fh	—	B8Fh	—
810h	—	890h	—	910h	—	990h	—	A10h	—	A90h	—	B10h	—	B90h	—
811h	—	891h	—	911h	—	991h	—	A11h	—	A91h	—	B11h	—	B91h	—
812h	—	892h	—	912h	—	992h	—	A12h	—	A92h	—	B12h	—	B92h	—
813h	—	893h	—	913h	—	993h	—	A13h	—	A93h	—	B13h	—	B93h	—
814h	—	894h	—	914h	—	994h	—	A14h	—	A94h	—	B14h	—	B94h	—
815h	—	895h	—	915h	—	995h	—	A15h	—	A95h	—	B15h	—	B95h	—
816h	—	896h	—	916h	—	996h	—	A16h	—	A96h	—	B16h	—	B96h	—
817h	—	897h	—	917h	—	997h	—	A17h	—	A97h	—	B17h	—	B97h	—
818h	—	898h	—	918h	—	998h	—	A18h	—	A98h	—	B18h	—	B98h	—
819h	—	899h	—	919h	—	999h	—	A19h	—	A99h	—	B19h	—	B99h	—
81Ah	—	89Ah	—	91Ah	—	99Ah	—	A1Ah	—	A9Ah	—	B1Ah	—	B9Ah	—
81Bh	—	89Bh	—	91Bh	—	99Bh	—	A1Bh	—	A9Bh	—	B1Bh	—	B9Bh	—
81Ch	—	89Ch	—	91Ch	—	99Ch	—	A1Ch	—	A9Ch	—	B1Ch	—	B9Ch	—
81Dh	—	89Dh	—	91Dh	—	99Dh	—	A1Dh	—	A9Dh	—	B1Dh	—	B9Dh	—
81Eh	—	89Eh	—	91Eh	—	99Eh	—	A1Eh	—	A9Eh	—	B1Eh	—	B9Eh	—
81Fh	—	89Fh	—	91Fh	—	99Fh	—	A1Fh	—	A9Fh	—	B1Fh	—	B9Fh	—
820h	Unimplemented Read as '0'	8A0h	Unimplemented Read as '0'	920h	Unimplemented Read as '0'	9A0h	Unimplemented Read as '0'	A20h	Unimplemented Read as '0'	AA0h	Unimplemented Read as '0'	B20h	Unimplemented Read as '0'	BA0h	Unimplemented Read as '0'
86Fh	—	8EFh	—	96Fh	—	9EFh	—	A6Fh	—	AEFh	—	B6Fh	—	BEFh	—
870h	Accesses 70h – 7Fh	8F0h	Accesses 70h – 7Fh	970h	Accesses 70h – 7Fh	9F0h	Accesses 70h – 7Fh	A70h	Accesses 70h – 7Fh	AF0h	Accesses 70h – 7Fh	B70h	Accesses 70h – 7Fh	BF0h	Accesses 70h – 7Fh
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFFh	—	B7Fh	—	BFFh	—

Legend:  = Unimplemented data memory locations, read as '0'.

**TABLE 2-9: PIC16F193X/LF193X MEMORY MAP, BANKS 24-31**

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	INDF0	C80h	INDF0	D00h	INDF0	D80h	INDF0	E00h	INDF0	E80h	INDF0	F00h	INDF0	F80h	INDF0
C01h	INDF1	C81h	INDF1	D01h	INDF1	D81h	INDF1	E01h	INDF1	E81h	INDF1	F01h	INDF1	F81h	INDF1
C02h	PCL	C82h	PCL	D02h	PCL	D82h	PCL	E02h	PCL	E82h	PCL	F02h	PCL	F82h	PCL
C03h	STATUS	C83h	STATUS	D03h	STATUS	D83h	STATUS	E03h	STATUS	E83h	STATUS	F03h	STATUS	F83h	STATUS
C04h	FSR0L	C84h	FSR0L	D04h	FSR0L	D84h	FSR0L	E04h	FSR0L	E84h	FSR0L	F04h	FSR0L	F84h	FSR0L
C05h	FSR0H	C85h	FSR0H	D05h	FSR0H	D85h	FSR0H	E05h	FSR0H	E85h	FSR0H	F05h	FSR0H	F85h	FSR0H
C06h	FSR1L	C86h	FSR1L	D06h	FSR1L	D86h	FSR1L	E06h	FSR1L	E86h	FSR1L	F06h	FSR1L	F86h	FSR1L
C07h	FSR1H	C87h	FSR1H	D07h	FSR1H	D87h	FSR1H	E07h	FSR1H	E87h	FSR1H	F07h	FSR1H	F87h	FSR1H
C08h	BSR	C88h	BSR	D08h	BSR	D88h	BSR	E08h	BSR	E88h	BSR	F08h	BSR	F88h	BSR
C09h	WREG	C89h	WREG	D09h	WREG	D89h	WREG	E09h	WREG	E89h	WREG	F09h	WREG	F89h	WREG
C0Ah	PCLATH	C8Ah	PCLATH	D0Ah	PCLATH	D8Ah	PCLATH	E0Ah	PCLATH	E8Ah	PCLATH	F0Ah	PCLATH	F8Ah	PCLATH
C0Bh	INTCON	C8Bh	INTCON	D0Bh	INTCON	D8Bh	INTCON	E0Bh	INTCON	E8Bh	INTCON	F0Bh	INTCON	F8Bh	INTCON
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	See Table 2-12
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—	F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh	—	E0Eh	—	E8Eh	—	F0Eh	—	F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh	—	E0Fh	—	E8Fh	—	F0Fh	—	F8Fh	
C10h	—	C90h	—	D10h	—	D90h	—	E10h	—	E90h	—	F10h	—	F90h	
C11h	—	C91h	—	D11h	—	D91h	—	E11h	—	E91h	—	F11h	—	F91h	
C12h	—	C92h	—	D12h	—	D92h	—	E12h	—	E92h	—	F12h	—	F92h	
C13h	—	C93h	—	D13h	—	D93h	—	E13h	—	E93h	—	F13h	—	F93h	
C14h	—	C94h	—	D14h	—	D94h	—	E14h	—	E94h	—	F14h	—	F94h	
C15h	—	C95h	—	D15h	—	D95h	—	E15h	—	E95h	—	F15h	—	F95h	
C16h	—	C96h	—	D16h	—	D96h	—	E16h	—	E96h	—	F16h	—	F96h	
C17h	—	C97h	—	D17h	—	D97h	—	E17h	—	E97h	—	F17h	—	F97h	
C18h	—	C98h	—	D18h	—	D98h	—	E18h	—	E98h	—	F18h	—	F98h	
C19h	—	C99h	—	D19h	—	D99h	—	E19h	—	E99h	—	F19h	—	F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah	—	E1Ah	—	E9Ah	—	F1Ah	—	F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh	—	E1Bh	—	E9Bh	—	F1Bh	—	F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch	—	E1Ch	—	E9Ch	—	F1Ch	—	F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh	—	E1Dh	—	E9Dh	—	F1Dh	—	F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh	—	E1Eh	—	E9Eh	—	F1Eh	—	F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh	—	E1Fh	—	E9Fh	—	F1Fh	—	F9Fh	
C20h	Unimplemented Read as '0'	CA0h	Unimplemented Read as '0'	D20h	Unimplemented Read as '0'	DA0h	Unimplemented Read as '0'	E20h	Unimplemented Read as '0'	EA0h	Unimplemented Read as '0'	F20h	Unimplemented Read as '0'	FA0h	
C6Fh	—	CEFh	—	D6Fh	—	DEFh	—	E6Fh	—	EEFh	—	F6Fh	—	FEFh	
C70h	Accesses 70h – 7Fh	CF0h	Accesses 70h – 7Fh	D70h	Accesses 70h – 7Fh	DF0h	Accesses 70h – 7Fh	E70h	Accesses 70h – 7Fh	EF0h	Accesses 70h – 7Fh	F70h	Accesses 70h – 7Fh	FF0h	Accesses 70h – 7Fh
CFFh	—	CFFh	—	D7Fh	—	DFFh	—	E7Fh	—	EFFh	—	F7Fh	—	FFFh	

**Legend:** ■ = Unimplemented data memory locations, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-10: PIC16F1933/1936/1938  
MEMORY MAP, BANK 15**

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	—
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	—
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	—
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	—
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	—
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h	Unimplemented Read as '0'
7EFh	

Legend:  = Unimplemented data memory locations, read as '0'.

**TABLE 2-11: PIC16F1934/1937/1939  
MEMORY MAP, BANK 15**

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	LCDSE2
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	LCDDATA2
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	LCDDATA5
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	LCDDATA8
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	LCDDATA11
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h	Unimplemented Read as '0'
7EFh	

Legend:  = Unimplemented data memory locations, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-12: PIC16F193X/LF193X MEMORY MAP, BANK 31**

Bank 31	
F8Ch	Unimplemented Read as '0'
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

Legend:  = Unimplemented data memory locations, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY**

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	
<b>Bank 0</b>												
000h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
001h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
002h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
003h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
004h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
005h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
006h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
007h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
008h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
009h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
00Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
00Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCF	0000 000x	0000 000u	
00Ch	PORTA	PORTA Data Latch when written: PORTA pins when read								xxxx xxxx	uuuu uuuu	
00Dh	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu	
00Eh	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu	
00Fh <sup>(3)</sup>	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu	
010h	PORTE	—	—	—	—	RE3	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>	---- xxxx	---- uuuu	
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
012h	PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	0000 00-0	0000 00-0	
013h	PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	-000 0-0-	-000 0-0-	
014h	PIR4	Unimplemented								—	—	
015h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu	
016h	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu	
017h	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu	
018h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	0000 00-0	uuuu uu-u	
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	0000 0x00	uuuu uxuu	
01Ah	TMR2	Timer 2 Module Register								0000 0000	0000 0000	
01Bh	PR2	Timer 2 Period Register								1111 1111	1111 1111	
01Ch	T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000	
01Dh	—	Unimplemented								—	—	
01Eh	CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	0--- 0000	0--- 0000	
01Fh	CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	---- 0000	---- 0000	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**Note 2:** These registers can be addressed from any bank.  
**Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 1</b>												
080h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
081h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
082h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
083h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
084h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
085h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
086h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
087h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
088h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
089h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
08Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
08Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
08Ch	TRISA	PORTA Data Direction Register								1111 1111	1111 1111	
08Dh	TRISB	PORTB Data Direction Register								1111 1111	1111 1111	
08Eh	TRISC	PORTC Data Direction Register								1111 1111	1111 1111	
08Fh <sup>(3)</sup>	TRISD	PORTD Data Direction Register								1111 1111	1111 1111	
090h	TRISE	—	—	—	—	TRISE3	TRISE2 <sup>(3)</sup>	TRISE1 <sup>(3)</sup>	TRISE0 <sup>(3)</sup>	---- 1111	---- 1111	
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
092h	PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	0000 00-0	0000 00-0	
093h	PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	-000 0-0-	-000 0-0-	
094h	—	Unimplemented								—	—	
095h	OPTION_REG	$\overline{WPUEN}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
096h	PCON	STKOVF	STKUNF	—	—	$\overline{RMCLR}$	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	00-- 11qq	qq-- qquu	
097h	WDTCON	—	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	--01 0110	--01 0110	
098h	OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	--00 0000	
099h	OSCCON	SPLLEN	IRCF3	IRCF2	IRCF1	IRCF0	—	SCS1	SCS0	0011 1-00	0011 1-00	
09Ah	OSCSTAT	T1OSCR	PLL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFR	00q0 0q0-	qqqq qq0-	
09Bh	ADRESL	A/D Result Register Low								xxxx xxxx	uuuu uuuu	
09Ch	ADRESH	A/D Result Register High								xxxx xxxx	uuuu uuuu	
09Dh	ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	ADON	-000 0000	-000 0000	
09Eh	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	—	ADNREF	ADPREF1	ADPREF0	0000 -000	0000 -000	
09Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 2</b>												
100h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
101h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
102h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
103h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
104h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
105h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
106h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
107h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
108h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
109h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
10Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
10Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
10Ch	LATA	PORTA Data Latch								xxxx xxxx	uuuu uuuu	
10Dh	LATB	PORTB Data Latch								xxxx xxxx	uuuu uuuu	
10Eh	LATC	PORTC Data Latch								xxxx xxxx	uuuu uuuu	
10Fh <sup>(3)</sup>	LATD	PORTD Data Latch								xxxx xxxx	uuuu uuuu	
110h	LATE	—	—	—	—	LATE3	LATE2 <sup>(3)</sup>	LATE1 <sup>(3)</sup>	LATE0 <sup>(3)</sup>	---- -xxx	---- -uuu	
111h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100	0000 -100	
112h	CM1CON1	C1INTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH1	C1NCH0	0000 --00	0000 --00	
113h	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	0000 -100	0000 -100	
114h	CM2CON1	C2INTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH1	C2NCH0	0000 --00	0000 --00	
115h	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	---- -000	---- -000	
116h	BORCON	SBOREN	—	—	—	—	—	—	BORRDY	1---- -q	u---- -u	
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	0q00 0000	0q00 0000	
118h	DACCON0	DACEN	DACLPS	DACOE	---	DACPSS1	DACPSS0	---	DACNSS	000- 00-0	000- 00-0	
119h	DACCON1	---	---	---	DACR4	DACR3	DACR2	DACR1	DACR0	---0 0000	---0 0000	
11Ah	SRCON0	SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR	0000 0000	0000 0000	
11Bh	SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E	0000 0000	0000 0000	
11Ch	—	Unimplemented								—	—	
11Dh	APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	-000 0000	-000 0000	
11Eh	—	Unimplemented								—	—	
11Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 3</b>												
180h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
181h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
182h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
183h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
184h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
185h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
186h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
187h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
188h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
189h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
18Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
18Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	0000 000x	0000 000u	
18Ch	ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	--11 1111	--11 1111	
18Dh	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	--11 1111	
18Eh	—	Unimplemented								—	—	
18Fh <sup>(3)</sup>	ANSELD	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111	
190h <sup>(3)</sup>	ANSELE	—	—	—	—	—	ANSE2	ANSE1	ANSE0	---- -111	---- -111	
191h	EEADRL	EEPROM / Program Memory Address Register Low Byte								0000 0000	0000 0000	
192h	EEADRH	—	EEPROM / Program Memory Address Register High Byte								-000 0000	-000 0000
193h	EEDATL	EEPROM / Program Memory Read Data Register Low Byte								xxxx xxxx	uuuu uuuu	
194h	EEDATH	—	—	EEPROM / Program Memory Read Data Register High Byte						--xx xxxx	--uu uuuu	
195h	EECON1	EEPGD	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	0000 x000	0000 q000	
196h	EECON2	EEPROM control register 2								0000 0000	0000 0000	
197h	—	Unimplemented								—	—	
198h	—	Unimplemented								—	—	
199h	RCREG	USART Receive Data Register								0000 0000	0000 0000	
19Ah	TXREG	USART Transmit Data Register								0000 0000	0000 0000	
19Bh	SPBRGL	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000	
19Ch	SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000	
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x	
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010	
19Fh	BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.



# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 4</b>												
200h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
201h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
202h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
203h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
204h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
205h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
206h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
207h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
208h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
209h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
20Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
20Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
20Ch	—	Unimplemented								—	—	
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111	
20Eh	—	Unimplemented								—	—	
20Fh	—	Unimplemented								—	—	
210h	WPUE	—	—	—	—	WPUE3	—	—	—	---- 1---	---- 1---	
211h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu	
212h	SSPADD	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0	0000 0000	0000 0000	
213h	SSPMSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111	
214h	SSPSTAT	SMP	CKE	$D/\overline{A}$	P	S	$R/\overline{W}$	UA	BF	0000 0000	0000 0000	
215h	SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000	
216h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000	
217h	SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000	
218h	—	Unimplemented								—	—	
219h	—	Unimplemented								—	—	
21Ah	—	Unimplemented								—	—	
21Bh	—	Unimplemented								—	—	
21Ch	—	Unimplemented								—	—	
21Dh	—	Unimplemented								—	—	
21Eh	—	Unimplemented								—	—	
21Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**Note 2:** These registers can be addressed from any bank.  
**Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 5</b>												
280h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
281h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
282h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
283h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
284h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
285h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
286h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
287h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
288h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
289h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
28Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
28Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	0000 000x	0000 000u	
28Ch	—	Unimplemented								—	—	
28Dh	—	Unimplemented								—	—	
28Eh	—	Unimplemented								—	—	
28Fh	—	Unimplemented								—	—	
290h	—	Unimplemented								—	—	
291h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu	
292h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu	
293h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000	
294h	PWM1CON	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	0000 0000	
295h	CCP1AS	CCP1ASE	CCP1AS2	CCP1AS1	CCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	0000 0000	
296h	PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	---0 0001	---0 0001	
297h	—	Unimplemented								—	—	
298h	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu	
299h	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu	
29Ah	CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	0000 0000	
29Bh	PWM2CON	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	0000 0000	
29Ch	CCP2AS	CCP2ASE	CCP2AS2	CCP2AS1	CCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	0000 0000	
29Dh	PSTR2CON	—	—	—	STR2SYNC	STR2D	STR2C	STR2B	STR2A	---0 0001	---0 0001	
29Eh	CCPTMRS0	C4TSEL1	C4TSEL0	C3TSEL1	C3TSEL0	C2TSEL1	C2TSEL0	C1TSEL1	C1TSEL0	0000 0000	0000 0000	
29Fh	CCPTMRS1	—	—	—	—	—	—	C5TSEL1	C5TSEL0	---- --00	---- --00	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 6</b>												
300h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
301h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
302h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
303h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
304h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
305h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
306h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
307h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
308h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
309h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
30Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
30Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
30Ch	—	Unimplemented								—	—	
30Dh	—	Unimplemented								—	—	
30Eh	—	Unimplemented								—	—	
30Fh	—	Unimplemented								—	—	
310h	—	Unimplemented								—	—	
311h	CCPR3L	Capture/Compare/PWM Register 3 (LSB)								xxxx xxxx	uuuu uuuu	
312h	CCPR3H	Capture/Compare/PWM Register 3 (MSB)								xxxx xxxx	uuuu uuuu	
313h	CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	0000 0000	
314h	PWM3CON	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000	0000 0000	
315h	CCP3AS	CCP3ASE	CCP3AS2	CCP3AS1	CCP3AS0	PSS3AC1	PSS3AC0	PSS3BD1	PSS3BD0	0000 0000	0000 0000	
316h	PSTR3CON	—	—	—	STR3SYNC	STR3D	STR3C	STR3B	STR3A	---0 0001	---0 0001	
317h	—	Unimplemented								—	—	
318h	CCPR4L	Capture/Compare/PWM Register 4 (LSB)								xxxx xxxx	uuuu uuuu	
319h	CCPR4H	Capture/Compare/PWM Register 4 (MSB)								xxxx xxxx	uuuu uuuu	
31Ah	CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	--00 0000	--00 0000	
31Bh	—	Unimplemented								—	—	
31Ch	CCPR5L	Capture/Compare/PWM Register 5 (LSB)								xxxx xxxx	uuuu uuuu	
31Dh	CCPR5H	Capture/Compare/PWM Register 5 (MSB)								xxxx xxxx	uuuu uuuu	
31Eh	CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	--00 0000	--00 0000	
31Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**Note 2:** These registers can be addressed from any bank.  
**Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 7</b>												
380h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
381h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
382h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
383h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
384h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
385h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
386h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
387h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
388h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
389h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
38Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
38Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
38Ch	—	Unimplemented								—	—	
38Dh	—	Unimplemented								—	—	
38Eh	—	Unimplemented								—	—	
38Fh	—	Unimplemented								—	—	
390h	—	Unimplemented								—	—	
391h	—	Unimplemented								—	—	
392h	—	Unimplemented								—	—	
393h	—	Unimplemented								—	—	
394h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	0000 0000	0000 0000	
395h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	0000 0000	0000 0000	
396h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	0000 0000	0000 0000	
397h	—	Unimplemented								—	—	
398h	—	Unimplemented								—	—	
399h	—	Unimplemented								—	—	
39Ah	—	Unimplemented								—	—	
39Bh	—	Unimplemented								—	—	
39Ch	—	Unimplemented								—	—	
39Dh	—	Unimplemented								—	—	
39Eh	—	Unimplemented								—	—	
39Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 8</b>												
400h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
401h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
402h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
403h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
404h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
405h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
406h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
407h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
408h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
409h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
40Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
40Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 000x	0000 000u	
40Ch	—	Unimplemented								—	—	
40Dh	—	Unimplemented								—	—	
40Eh	—	Unimplemented								—	—	
40Fh	—	Unimplemented								—	—	
410h	—	Unimplemented								—	—	
411h	—	Unimplemented								—	—	
412h	—	Unimplemented								—	—	
413h	—	Unimplemented								—	—	
414h	—	Unimplemented								—	—	
415h	TMR4	Timer 4 Module Register								0000 0000	0000 0000	
416h	PR4	Timer 4 Period Register								1111 1111	1111 1111	
417h	T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	-000 0000	
418h	—	Unimplemented								—	—	
419h	—	Unimplemented								—	—	
41Ah	—	Unimplemented								—	—	
41Bh	—	Unimplemented								—	—	
41Ch	TMR6	Timer 6 Module Register								0000 0000	0000 0000	
41Dh	PR6	Timer 6 Period Register								1111 1111	1111 1111	
41Eh	T6CON	—	T6OUTPS3	T6OUTPS2	T6OUTPS1	T6OUTPS0	TMR6ON	T6CKPS1	T6CKPS0	-000 0000	-000 0000	
41Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**Note 2:** These registers can be addressed from any bank.  
**Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Banks 9-14</b>												
x00h/ x80h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x00h/ x81h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x02h/ x82h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
x03h/ x83h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---b qnuu	
x04h/ x84h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
x05h/ x85h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
x06h/ x86h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
x07h/ x87h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
x08h/ x88h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
x09h/ x89h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
x0Ah/ x8Ah <sup>(1),(2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
x0Bh/ x8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 000x	0000 000u	
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.
- 2:** These registers can be addressed from any bank.
- 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 15</b>												
780h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
781h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
782h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
783h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
784h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
785h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
786h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
787h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
788h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
789h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
78Ah <sup>(1, 2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
78Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
78Ch	—	Unimplemented								—	—	
78Dh	—	Unimplemented								—	—	
78Eh	—	Unimplemented								—	—	
78Fh	—	Unimplemented								—	—	
790h	—	Unimplemented								—	—	
791h	LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	000- 0011	000- 0011	
792h	LCDPS	WFT	BIASMD	LCD A	WA	LP3	LP2	LP1	LP0	0000 0000	0000 0000	
793h	LCDREF	LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	000- 000-	000- 000-	
794h	LCDCST	—	—	—	—	—	LCDCST2	LCDCST1	LCDCST0	---- -000	---- -000	
795h	LCDRL	LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0	0000 -000	0000 -000	
796h	—	Unimplemented								—	—	
797h	—	Unimplemented								—	—	
798h	LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu	
799h	LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu	
79Ah	LCDSE2 <sup>(3)</sup>	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	0000 0000	uuuu uuuu	
79Bh	—	Unimplemented								—	—	
79Ch	—	Unimplemented								—	—	
79Dh	—	Unimplemented								—	—	
79Eh	—	Unimplemented								—	—	
79Fh	—	Unimplemented								—	—	
7A0h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	xxxx xxxx	uuuu uuuu	
7A1h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	xxxx xxxx	uuuu uuuu	
7A2h	LCDDATA2 <sup>(3)</sup>	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	xxxx xxxx	uuuu uuuu	
7A3h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	xxxx xxxx	uuuu uuuu	
7A4h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	xxxx xxxx	uuuu uuuu	
7A5h	LCDDATA5 <sup>(3)</sup>	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	xxxx xxxx	uuuu uuuu	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**2:** These registers can be addressed from any bank.  
**3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
<b>Bank 15 (Continued)</b>											
7A6h	LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	xxxx xxxx	uuuu uuuu
7A7h	LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	xxxx xxxx	uuuu uuuu
7A8h	LCDDATA8 <sup>(3)</sup>	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	xxxx xxxx	uuuu uuuu
7A9h	LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	xxxx xxxx	uuuu uuuu
7AAh	LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	xxxx xxxx	uuuu uuuu
7ABh	LCDDATA11 <sup>(3)</sup>	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	xxxx xxxx	uuuu uuuu
7ACh — 7EFh	—	Unimplemented								—	—

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.
- 2:** These registers can be addressed from any bank.
- 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.



# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Banks 16-30</b>												
x00h/ x80h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x00h/ x81h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x02h/ x82h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
x03h/ x83h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---b rrrr	
x04h/ x84h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
x05h/ x85h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
x06h/ x86h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
x07h/ x87h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
x08h/ x88h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
x09h/ x89h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
x0Ah/ x8Ah <sup>(1),(2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
x0Bh/ x8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 000x	0000 000u	
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.
- Note 2:** These registers can be addressed from any bank.
- Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

# PIC16F193X/LF193X

**TABLE 2-13: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets	
<b>Bank 31</b>												
F80h <sup>(2)</sup>	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
F81h <sup>(2)</sup>	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
F82h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
F83h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
F84h <sup>(2)</sup>	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
F85h <sup>(2)</sup>	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
F86h <sup>(2)</sup>	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
F87h <sup>(2)</sup>	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
F88h <sup>(2)</sup>	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
F89h <sup>(2)</sup>	WREG	Working Register								0000 0000	uuuu uuuu	
F8Ah <sup>(1),(2)</sup>	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
F8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 000x	0000 000u	
F8Ch — FE3h	—	Unimplemented								—	—	
FE4h	STATUS_SHAD						Z	DC	C	---- -xxx	---- -uuu	
FE5h	WREG_SHAD	Working Register Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu	
FE6h	BSR_SHAD									---x xxxx	---u uuuu	
FE7h	PCLATH_SHAD		Program Counter Latch High Register Normal (Non-ICD) Shadow								-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	Indirect Data Memory Address 0 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu	
FE9h	FSR0H_SHAD	Indirect Data Memory Address 0 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu	
FEAh	FSR1L_SHAD	Indirect Data Memory Address 1 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu	
FEBh	FSR1H_SHAD	Indirect Data Memory Address 1 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu	
FECh	—	Unimplemented								—	—	
FEDh	STKPTR	—	—	—	Current Stack pointer					---1 1111	---1 1111	
FEEh	TOSL	Top of Stack Low byte								xxxx xxxx	uuuu uuuu	
FEFh	TOSH	—	Top of Stack High byte								-xxx xxxx	-uuu uuuu

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.  
**Note 2:** These registers can be addressed from any bank.  
**Note 3:** These registers/bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.

## 2.2.3 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation of the PIC16F193X/LF193X.

These registers are listed below:

- INDF0
- INDF1
- PCL
- STATUS
- FSR0 Low
- FSR0 High
- FSR1 Low
- FSR1 High
- BSR
- WREG
- PCLATH
- INTCON

<p><b>Note:</b> The core registers are the first 12 addresses of every data memory bank.</p>
--

# PIC16F193X/LF193X

## 2.2.3.1 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 TO and 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 (Refer to Section 26.0 "Instruction Set Summary").

**Note 1:** The C and DC bits operate as Borrow and Digit Borrow out bits, respectively, in subtraction.

**REGISTER 2-1: STATUS: STATUS REGISTER**

U-0	U-0	U-0	R-1/q	R-1/q	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	TO	PD	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
bit 7							bit 0

**Legend:**

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

- bit 7-5      **Unimplemented:** Read as '0'
- bit 4      **TO:** Time-out bit  
             1 = After power-up, CLRWD<sub>T</sub> instruction or SLEEP instruction  
             0 = A WDT time-out occurred
- bit 3      **PD:** Power-down bit  
             1 = After power-up or by the CLRWD<sub>T</sub> 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/Digit Borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)<sup>(1)</sup>  
             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/Borrow bit<sup>(1)</sup> (ADDWF, ADDLW, SUBLW, SUBWF instructions)<sup>(1)</sup>  
             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 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-order or low-order bit of the source register.

## 2.2.3.2 OPTION register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- External INT interrupt
- Timer0
- Weak pull-ups

### REGISTER 2-2: OPTION\_REG: OPTION REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUEN	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

#### Legend:

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

- bit 7      **WPUEN:** Weak Pull-up Enable bit  
 1 = All weak pull-ups are disabled (except MCLR, if it is enabled)  
 0 = Weak pull-ups are enabled by individual WPUx latch values
- bit 6      **INTEDG:** Interrupt Edge Select bit  
 1 = Interrupt on rising edge of RB0/INT pin  
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5      **T0CS:** Timer0 Clock Source Select bit  
 1 = Transition on RA4/T0CKI pin  
 0 = Internal instruction cycle clock (Fosc/4)
- bit 4      **T0SE:** Timer0 Source Edge Select bit  
 1 = Increment on high-to-low transition on RA4/T0CKI pin  
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3      **PSA:** Prescaler Assignment bit  
 1 = Prescaler is inactive and has no effect on the Timer0 interrupt rate  
 0 = Prescaler is active and affects the Timer0 interrupt rate
- bit 2-0    **PS<2:0>:** Prescaler Rate Select bits

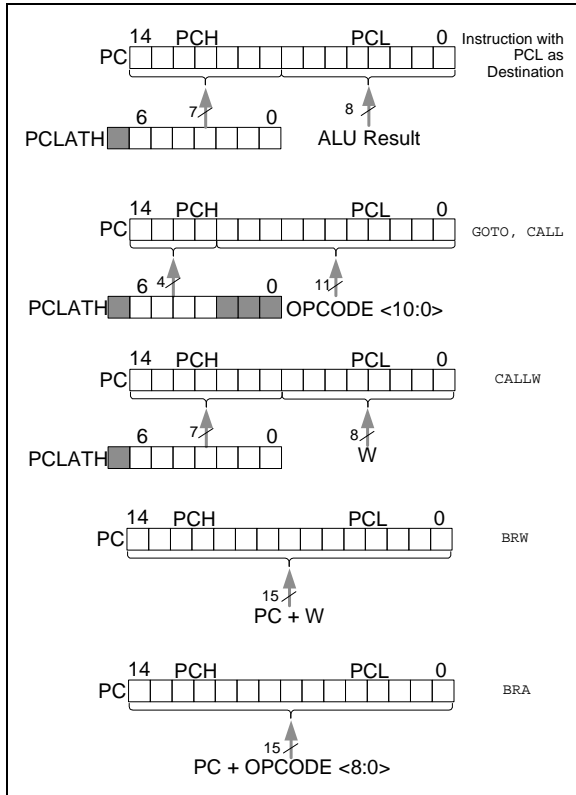
Bit Value	Timer0 Rate
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

# PIC16F193X/LF193X

## 2.3 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-4 shows the five situations for the loading of the PC.

**FIGURE 2-4: LOADING OF PC IN DIFFERENT SITUATIONS**



### 2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 7 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

### 2.3.2 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).

### 2.3.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

### 2.3.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.

## 2.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figures 2-1 and 2-3). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are 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, if the STVREN bit = 0 (Configuration Word 2). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on).

**Note 1:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

## 2.4.1 ACCESSING THE STACK

The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is 5 bits to allow detection of overflow and underflow.

During normal program operation, `CALL`, `CALLW` and Interrupts will increment STKPTR while `RETURN` and `RETFIE` will decrement STKPTR. At any time STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a `CALL` or `CALLW` will write the PC and then increment the STKPTR, and a return will decrement the PC and then unload the PC.

## 2.4.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Word 2 is programmed, the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

## 2.5 Indirect Addressing, INDF and FSR Registers

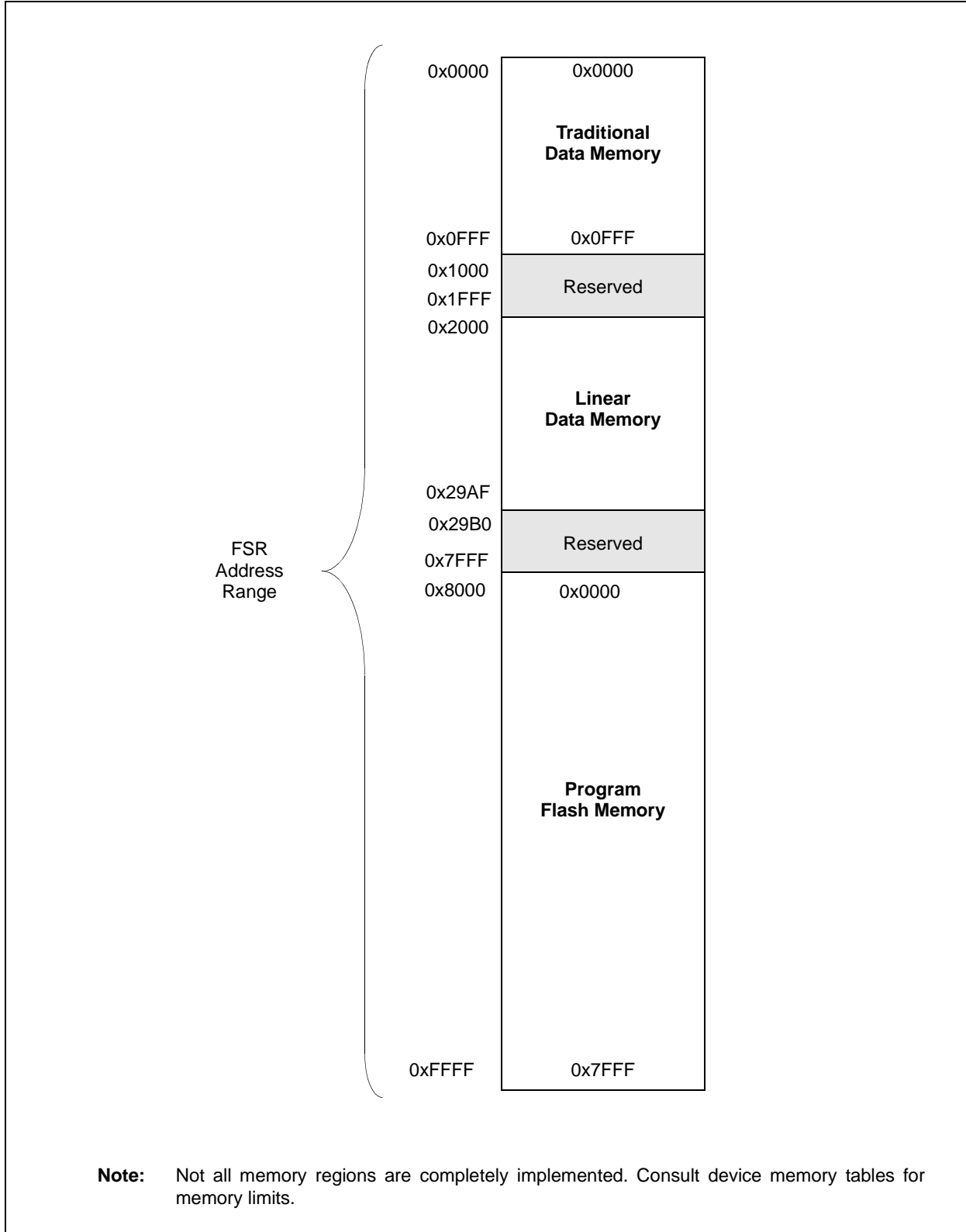
The INDF<sub>n</sub> registers are not physical registers. Any instruction that accesses an INDF<sub>n</sub> register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSR<sub>n</sub> address specifies one of the two INDF<sub>n</sub> registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSR<sub>n</sub> register value is created by the pair FSR<sub>nH</sub> and FSR<sub>nL</sub>.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

# PIC16F193X/LF193X

FIGURE 2-5: INDIRECT ADDRESSING

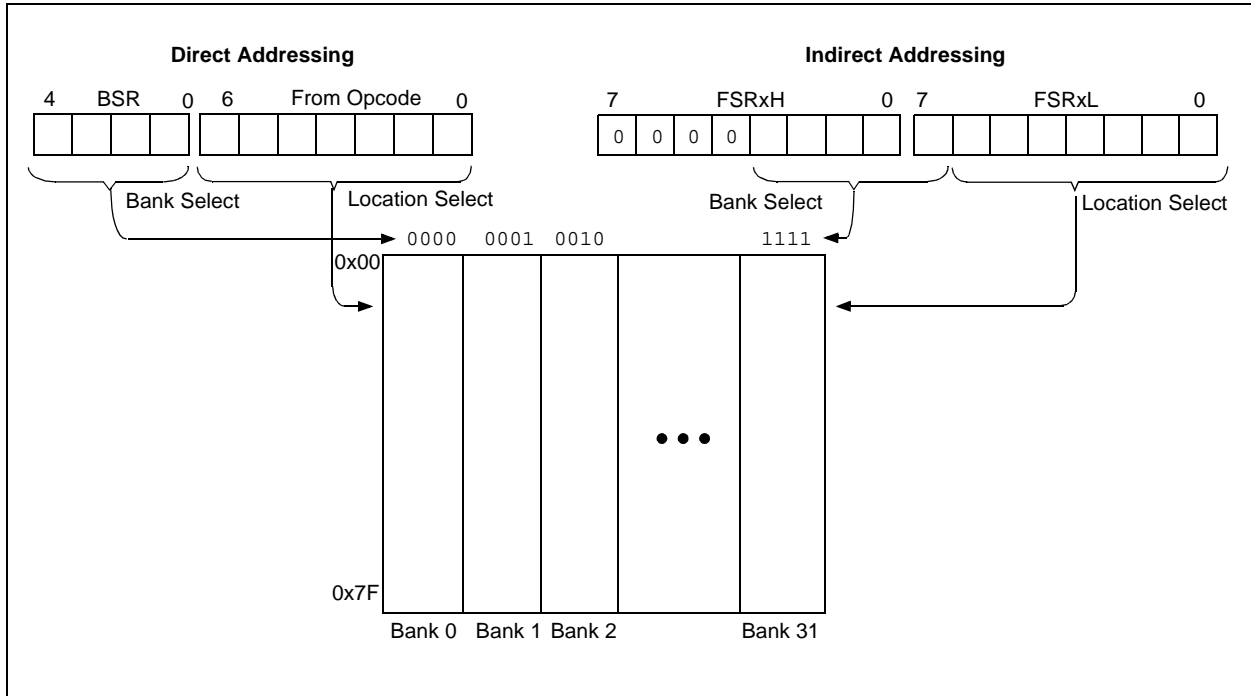




## 2.5.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address 0x000 to FSR address 0xFFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

**FIGURE 2-6: TRADITIONAL DATA MEMORY MAP**



# PIC16F193X/LF193X

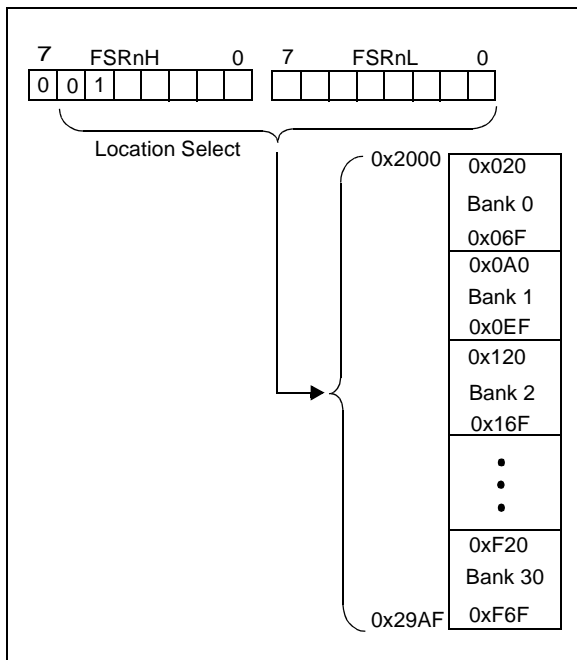
## 2.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

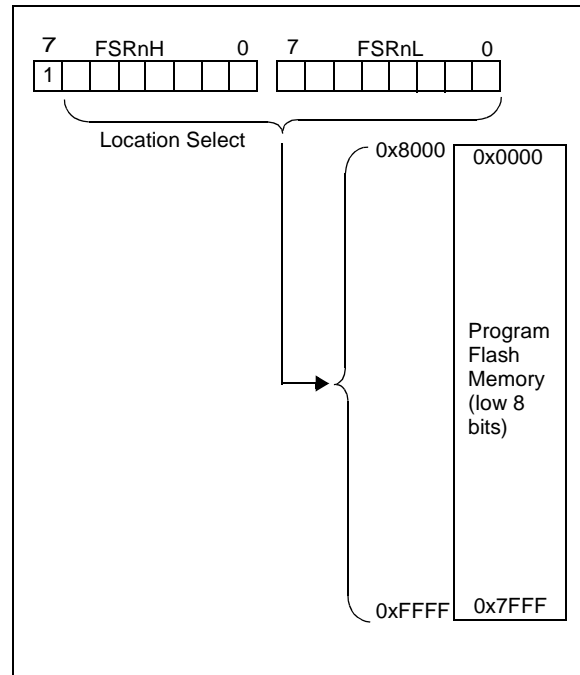
**FIGURE 2-7: LINEAR DATA MEMORY MAP**



## 2.5.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower 8 bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

**FIGURE 2-8: PROGRAM FLASH MEMORY MAP**



## 3.0 RESETS

The PIC16F193X/LF193X differentiates between various kinds of Reset:

- Power-on Reset (POR)
- WDT Reset during normal operation
- MCLR Reset
- Brown-out Reset (BOR)
- RESET instruction
- Stack Overflow
- Stack Underflow

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:

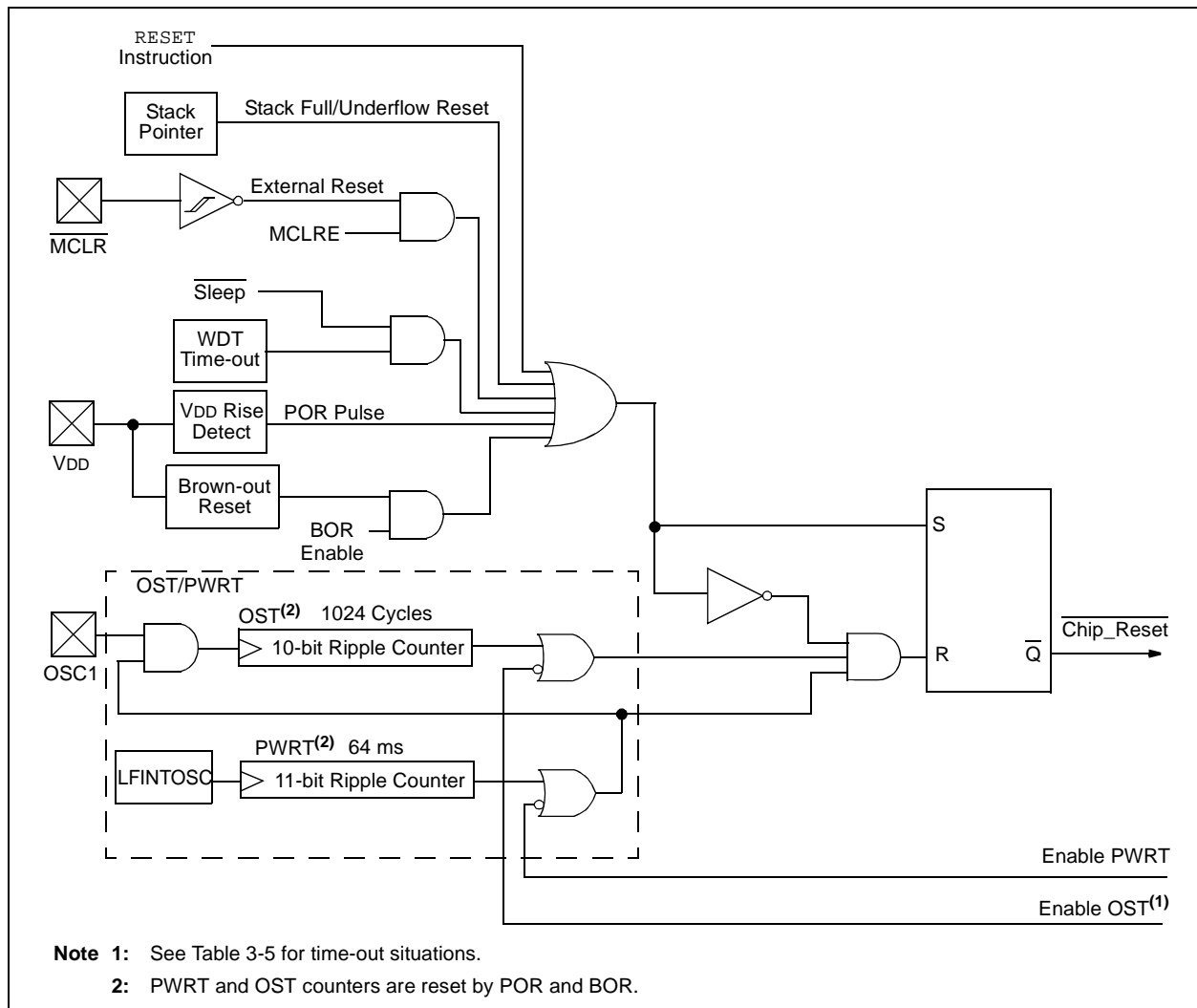
- Power-on Reset (POR)
- MCLR Reset
- WDT Reset
- Brown-out Reset (BOR)

Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation.  $\overline{TO}$  and  $\overline{PD}$  bits are set or cleared differently in different Reset situations, as indicated in Table 3-6. These bits are used in software to determine the nature of the Reset.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The MCLR Reset path has a noise filter to detect and ignore small pulses. See **Section 28.0 "Electrical Specifications"** for pulse width specifications.

**FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC16F193X/LF193X

**TABLE 3-1: STATUS BITS AND THEIR SIGNIFICANCE**

STKOVF	STKUNF	RMCLR	RI	POR	BOR	TO	PD	Condition
0	0	1	1	0	x	1	1	Power-on Reset or LDO Reset
0	0	1	1	0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	0	1	1	0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
0	0	1	1	u	0	1	1	Brown-out Reset
u	u	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	0	u	u	u	u	u	$\overline{MCLR}$ Reset during normal operation
u	u	0	u	u	u	1	0	$\overline{MCLR}$ Reset during Sleep
u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

**TABLE 3-2: RESET CONDITION FOR SPECIAL REGISTERS<sup>(2)</sup>**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
$\overline{MCLR}$ Reset during normal operation	0000h	---u uuuu	uu-- 0uuu
$\overline{MCLR}$ Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1 <sup>(1)</sup>	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

**2:** If a Status bit is not implemented, that bit will be read as '0'.

## 3.1 MCLR

The PIC16F193X/LF193X has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

It should be noted that a Reset does not drive the MCLR pin low.

Voltages applied to the pin that exceed its specification can result in both MCLR Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 3-2, is suggested.

An internal MCLR option is enabled by clearing the MCLRE bit in the Configuration Word register. When MCLRE = 0, the Reset signal to the chip is generated internally. When the MCLRE = 1, the RE3/MCLR pin becomes an external Reset input. In this mode, the RE3/MCLR pin has a weak pull-up to VDD. In-Circuit Serial Programming is not affected by selecting the internal MCLR option.

Low-voltage programming (LVP) mode will override MCLRE.

For additional information, refer to Application Note AN607, "Power-up Trouble Shooting" (DS00607).

## 3.3 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 WDT oscillator. For more information, see **Section 8.5 "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, 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.

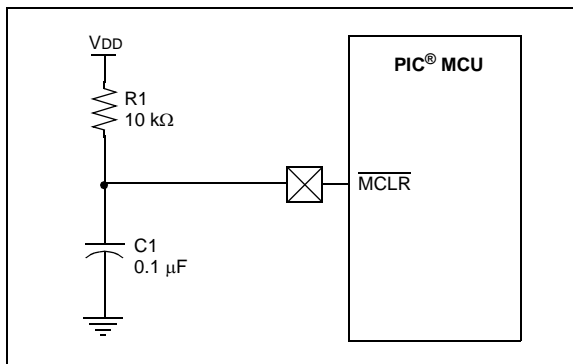
The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 28.0 "Electrical Specifications"**).

**Note:** The Power-up Timer is enabled by the PWRTE bit in the Configuration Word.

**FIGURE 3-2: RECOMMENDED MCLR CIRCUIT**



## 3.2 Power-on Reset (POR)

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. A maximum rise time for VDD is required. See **Section 28.0 "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 3.5 "Brown-Out Reset (BOR)"**).

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.

## 3.4 Watchdog Timer (WDT)

The WDT has the following features:

- Independent prescaler from Timer0
- Time-out period is from 1.024 ms to 268 seconds, typical
- Enabled by Configuration bits WDTE<1:0>
- Can be disabled during Sleep
- Controlled by WDTCN register

WDT is cleared under certain conditions described in Table 3-3.

### 3.4.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz internal oscillator.

**Note:** When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset. When the OST count has expired, the WDT will begin counting (if enabled).

# PIC16F193X/LF193X

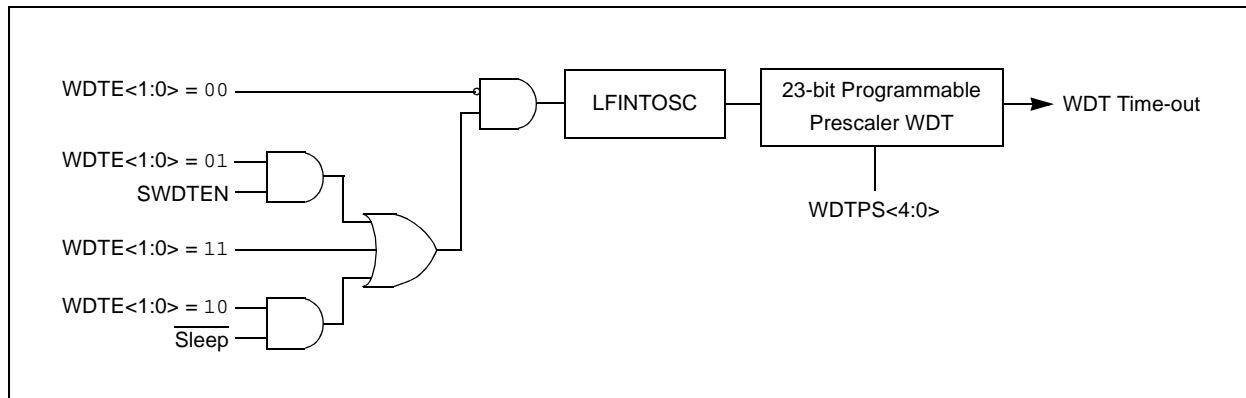
## 3.4.2 WDT CONTROL

The WDTE<1:0> bits are located in the Configuration Word Register 1. When set to '11', the WDT runs continuously. When entering Sleep the WDT is always cleared. When set to '10', the WDT is enabled while running, and disabled during Sleep. When '01' the WDT is under control of the SWDTEN bit, and when '00' the WDT is always disabled.

The WDTCON register contains the SWDTEN bit and WDTPS<4:0> bits. When the WDTE<1:0> bits in the Configuration Word 1 register are anything but '01', the SWDTEN bit has no effect. When WDTE = 01, the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable the WDT and clearing the bit will disable the WDT.

The WDTPS<4:0> bits control the prescaler. See Register 3-1. The Reset value of WDTCON gives a nominal WDT interval of ~2s. Upon Reset, the SWDTEN value will leave the WDT disabled if WDTE<1:0> is '01' in the Configuration Word. The prescaler will always be cleared on a Reset.

**FIGURE 3-3: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 3-3: WDT STATUS**

Conditions	WDT
WDTE<1:0> = 00	Cleared
WDTE<1:0> = 01 and SWDTEN = 0	
WDTE<1:0> = 10 and enter Sleep	
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
Change INTOSC divider (IRCF bits)	Unaffected

# PIC16F193X/LF193X

## REGISTER 3-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
bit 7							bit 0

### Legend:

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

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

bit 5-1      **WDTPS<4:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

- 00000 = 1:32 (Interval 1 ms typ)
- 00001 = 1:64 (Interval 2 ms typ)
- 00010 = 1:128 (Interval 4 ms typ)
- 00011 = 1:256 (Interval 8 ms typ)
- 00100 = 1:512 (Interval 16 ms typ)
- 00101 = 1:1024 (Interval 32 ms typ)
- 00110 = 1:2048 (Interval 64 ms typ)
- 00111 = 1:4096 (Interval 128 ms typ)
- 01000 = 1:8192 (Interval 256 ms typ)
- 01001 = 1:16384 (Interval 512 ms typ)
- 01010 = 1:32768 (Interval 1s typ)
- 01011 = 1:65536 (Interval 2s typ) (Reset value)

- 01100 = 1:131072 ( $2^{17}$ ) (Interval 4s typ)
- 01101 = 1:262144 ( $2^{18}$ ) (Interval 8s typ)
- 01110 = 1:524288 ( $2^{19}$ ) (Interval 16s typ)
- 01111 = 1:1048576 ( $2^{20}$ ) (Interval 32s typ)
- 10000 = 1:2097152 ( $2^{21}$ ) (Interval 64s typ)
- 10001 = 1:4194304 ( $2^{22}$ ) (Interval 128s typ)
- 10010 = 1:8388608 ( $2^{23}$ ) (Interval 256s typ)

10011 = Reserved. Results in minimum interval (1:32)

•  
•  
•

11111 = Reserved. Results in minimum interval (1:32)

bit 0      **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 00:

This bit is ignored.

If WDTE<1:0> = 01:

1 = WDT is turned on

0 = WDT is turned off

If WDTE<1:0> = 1x:

This bit is ignored.

# PIC16F193X/LF193X

## 3.5 Brown-Out Reset (BOR)

Brown-out Reset is enabled by programming the BOREN<1:0> bits in the Configuration register. The brown-out trip point is selectable from two trip points via the BORV bit in the Configuration register.

Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

Two bits are used to enable the BOR. When BOREN = 11, the BOR is always enabled. When BOREN = 10, the BOR is enabled, but disabled during

Sleep. When BOREN = 01, the BOR is controlled by the SBOREN bit of the BORCON register. When BOREN = 00, the BOR is disabled.

If VDD falls below VBOR for greater than parameter (TBOR) (see **Section 28.0 “Electrical Specifications”**), the Brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not ensured to occur if VDD falls below VBOR for more than parameter (TBOR).

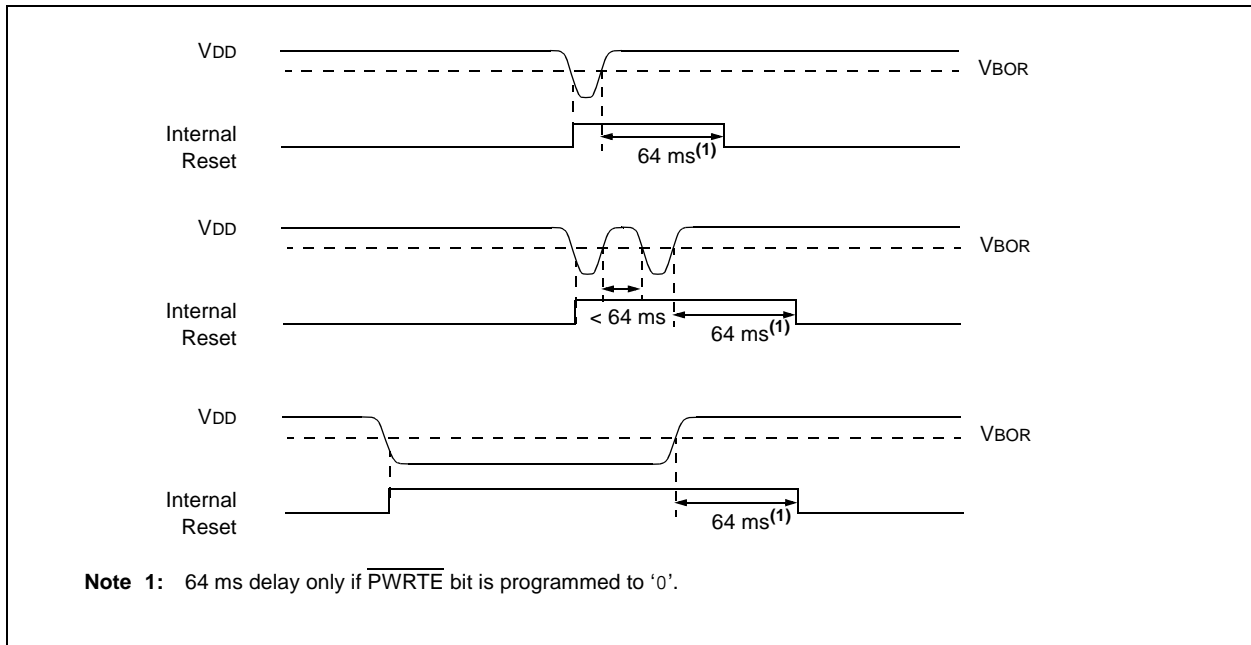
If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

**TABLE 3-4: BOR OPERATING MODES**

BOREN Config bits	SBOREN	Device Mode	BOR Mode	Device Operation upon release of POR	Device Operation upon wake-up from Sleep
BOR_ON (11)	X	X	Active	Waits for BOR ready <sup>(1)</sup>	
BOR_NSLEEP (10)	X	Awake	Active	Waits for BOR ready	
BOR_NSLEEP (10)	X	Sleep	Disabled		
BOR_SBOREN (01)	1	X	Active	Begins immediately	
BOR_SBOREN (01)	0	X	Disabled	Begins immediately	
BOR_OFF (00)	X	X	Disabled	Begins immediately	

**Note 1:** Even though this case specifically waits for the BOR, the BOR is already operating, so there is no delay in startup.

**FIGURE 3-4: BROWN-OUT SITUATIONS**





# PIC16F193X/LF193X

## REGISTER 3-2: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	U-0	U-0	U-0	U-0	U-0	U-0	R/W-q/u
SBOREN	—	—	—	—	—	—	BORRDY
bit 7							bit 0

### Legend:

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

bit 7 **SBOREN:** Software Brown-out Reset Enable bit

If BOREN ≠ 01:

SBOREN is read/write, but has no effect on the BOR.

If BOREN = 01:

1 = BOR Enabled

0 = BOR Disabled

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit

1 = The Brown-out Reset circuit is active and armed

0 = The Brown-out Reset circuit is disabled or is warming up

# PIC16F193X/LF193X

---

## 3.5.1 BOR HIBERNATE/REARM

The BOR circuit has an output that feeds into the POR circuit and rearms the POR within the operating range of the BOR. This early rearming of the POR ensures that the device will remain in Reset in the event that VDD falls below the operating range of the BOR circuitry.

## 3.6 Reset Instruction

A `RESET` instruction will cause a device Reset. The RI bit in the PCON register will be set to '0'. See Table 3-6 for default conditions after a `RESET` instruction has occurred.

## 3.7 Stack Overflow/Underflow

Device Resets on Stack Overflow and Stack Underflow conditions are enabled by setting the STVREN bit in Configuration Word 2. When STVREN is set, an overflow or underflow condition will set the appropriate STKOVF or STKUNF bit in the PCON register and then cause a device Reset. When STVREN is cleared, an overflow or underflow condition will set the appropriate STKOVF or STKUNF bit, but not cause a device Reset. The STKOVF or STKUNF bit is cleared by user software or a Power-on Reset.

## 3.8 Power-Up Time-out Sequence

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR or BOR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and `PWRTE` bit status. For example, in EC mode with `PWRTE` bit = 1 (PWRT disabled), there will be no time out at all. Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences.

Since the time outs occur from the POR pulse, if `MCLR` is kept low long enough, the time-outs will expire. Then, bringing `MCLR` high will begin execution immediately (see Figure 3-6). This is useful for testing purposes or to synchronize more than one PIC16F193X/LF193X device operating in parallel.

Table 3-7 shows the Reset conditions for some special registers.

## 3.9 Power Control (PCON) Register

The Power Control (PCON) register has six Status bits to indicate what type of Reset that last occurred.

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

The PCON register bits are shown in Register 3-3.

### 3.9.1 PCON REGISTER

The Power Control (PCON) register contains flag bits (refer to Table 3-6) to differentiate between a:

- Power-on Reset ( $\overline{\text{POR}}$ )
- Brown-out Reset ( $\overline{\text{BOR}}$ )
- Reset Instruction Reset ( $\overline{\text{RI}}$ )
- Stack Overflow Reset (STKOVF)
- Stack Underflow Reset (STKUVF)

### REGISTER 3-3: PCON: POWER CONTROL REGISTER

R/W-0/q	R/W-0/q	U-0	U-0	R/W-1/q	R/W-1/q	R/W-q/u	R/W-q/u
STKOVF	STKUNF	—	—	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

#### Legend:

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

bit 7	<b>STKOVF:</b> Stack Overflow Flag bit 1 = A Stack Overflow occurred (more CALLS than fit on the stack) 0 = A Stack Overflow has not occurred or set to '0' by firmware
bit 6	<b>STKUNF:</b> Stack Underflow Flag bit 1 = A Stack Underflow occurred (more RETURNS than CALLS) 0 = A Stack Underflow has not occurred or set to '0' by firmware
bit 5-4	<b>Unimplemented:</b> Read as '0'
bit 3	<b><math>\overline{\text{RMCLR}}</math>:</b> MCLR Reset Flag bit 1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to '1' by firmware 0 = A $\overline{\text{MCLR}}$ Reset has occurred (set to '0' in hardware when a $\overline{\text{MCLR}}$ Reset occurs)
bit 2	<b><math>\overline{\text{RI}}</math>:</b> RESET Instruction Flag bit 1 = A RESET instruction has not been executed or set to '1' by firmware 0 = A RESET instruction has been executed (set to '0' in hardware upon executing a RESET instruction)
bit 1	<b><math>\overline{\text{POR}}</math>:</b> 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	<b><math>\overline{\text{BOR}}</math>:</b> Brown-out Reset Status bit 1 = No Brown-out Reset occurred 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

# PIC16F193X/LF193X

**TABLE 3-5: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up and Brown-out Reset		Wake-up from Sleep or Oscillator Switch
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	
XT, HS, LP	$64 \text{ ms} + 1024 \cdot \text{Tosc}$	$1024 \cdot \text{Tosc}$	$1024 \cdot \text{Tosc}$
External RC	64 ms	—	—
EC	64 ms	—	—
INTOSC	64 ms	1 $\mu\text{s}$	1 $\mu\text{s}$

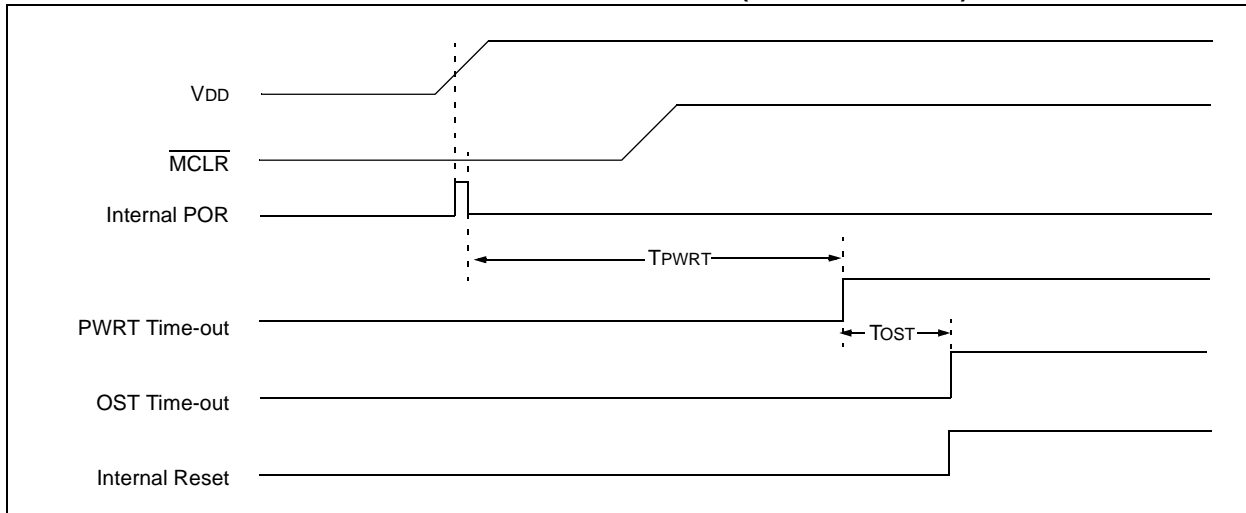
**Note 1:** LP mode with T1OSC disabled.

**TABLE 3-6: RESET BITS AND THEIR SIGNIFICANCE**

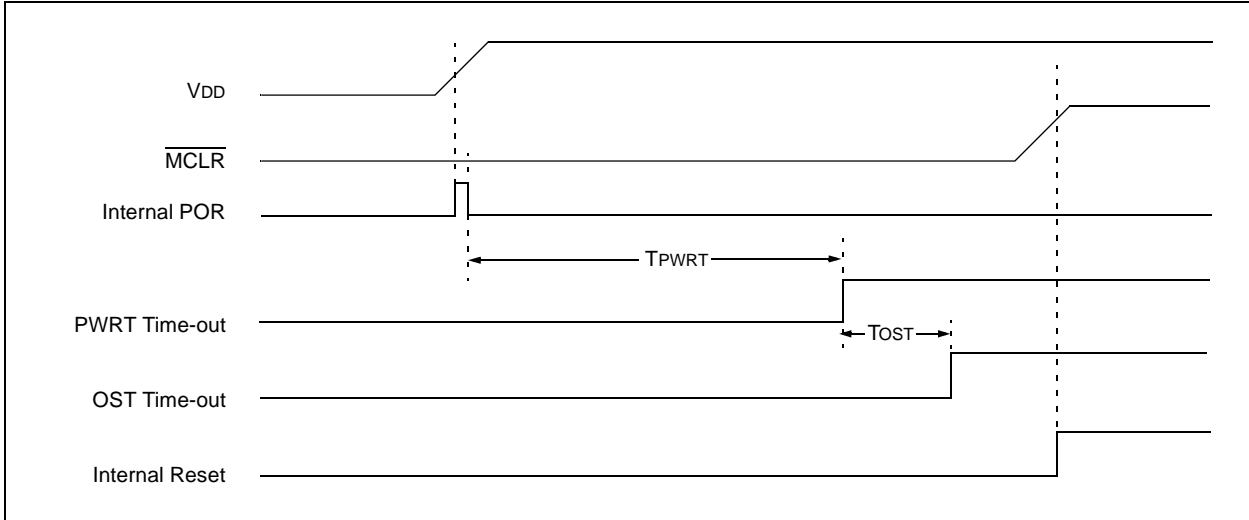
STKOVF	STKUNF	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	0	1	1	0	x	1	1	Power-on Reset
0	0	1	1	0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
0	0	1	u	u	0	1	1	Brown-out Reset
u	u	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	0	u	u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	0	u	u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep
u	u	u	0	u	u	u	u	RESET instruction executed
1	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

**Legend:** u = unchanged, x = unknown

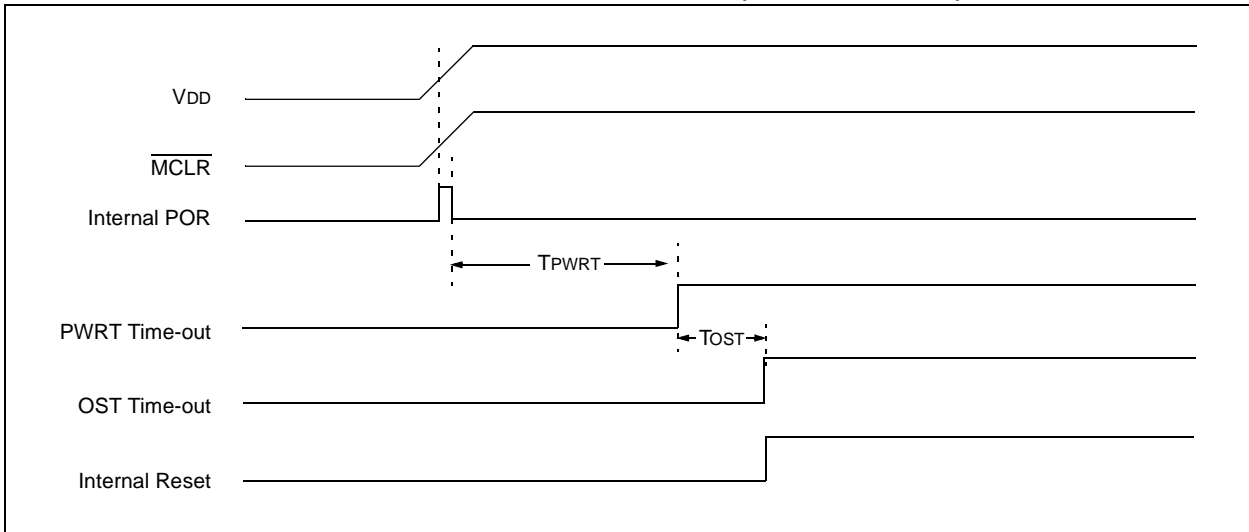
**FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ ): CASE 1**



**FIGURE 3-6: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ ): CASE 2**



**FIGURE 3-7: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  WITH VDD): CASE 3**



# PIC16F193X/LF193X

**TABLE 3-7: RESET CONDITION FOR SPECIAL REGISTERS<sup>(2)</sup>**

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
MCLR Reset during normal operation	0000h	---u uuuu	uu-- 0uuu
MCLR Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1 <sup>(1)</sup>	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

**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.

**2:** If a Status bit is not implemented, that bit will be read as '0'.

**TABLE 3-8: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	63
PCON	STKOVF	STKUNF	—	—	RMCLR	RI	POR	BOR	65
STATUS	—	—	—	TO	PD	Z	DC	C	50
WDTCON	—	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	61

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition. Shaded cells are not used by Resets.

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

## 4.0 INTERRUPTS

The PIC16F193X/LF193X device family features an interruptible core, allowing certain events to preempt normal program flow. An Interrupt Service Routine (ISR) is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

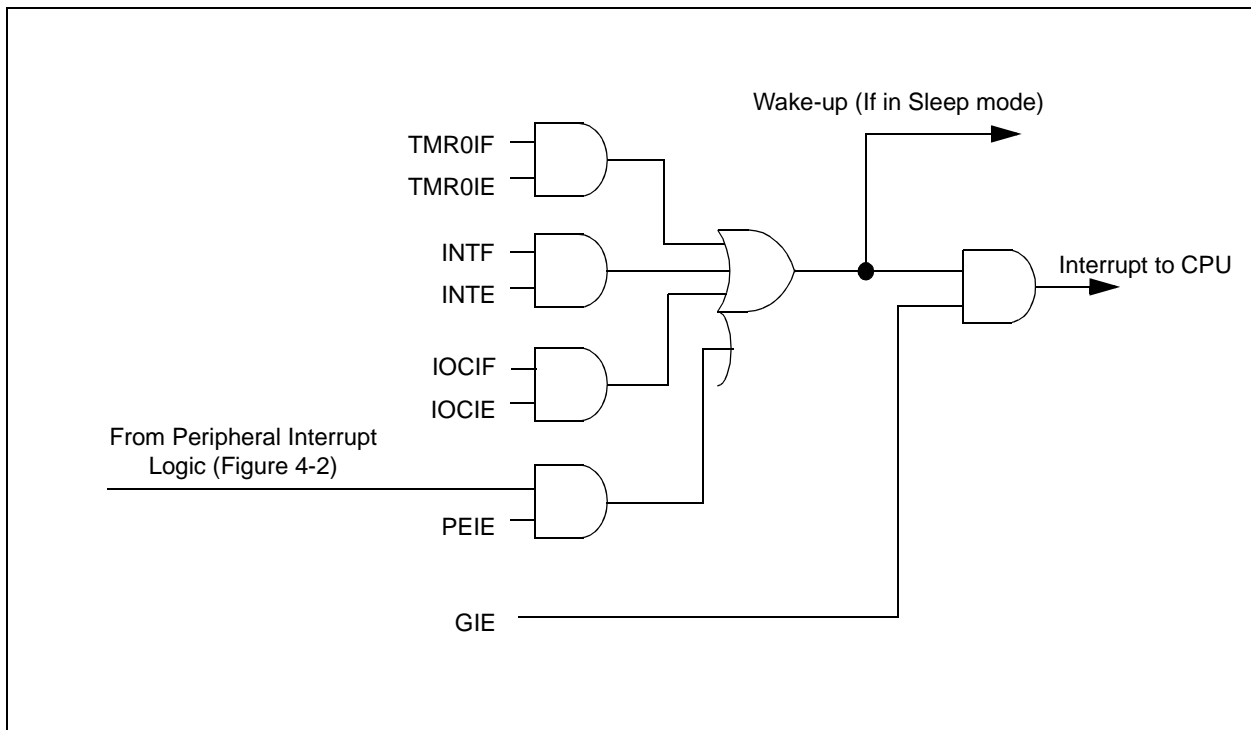
The PIC16F193X/LF193X device family has 23 interrupt sources, differentiated by corresponding interrupt enable and flag bits:

- External Edge Detect on INT Pin Interrupt
- Interrupt-on-Change Interrupt
- A/D Conversion Complete Interrupt
- EEPROM Write Complete Interrupt
- EUSART Receive Interrupt
- EUSART Transmit Interrupt
- LCD Module Interrupt
- Oscillator Fail Interrupt

- Timer0 Overflow Interrupt
- Timer1 Gate Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match with PR2 Interrupt
- Timer4 Match with PR4 Interrupt
- Timer6 Match with PR6 Interrupt
- Comparator C1 Interrupt
- Comparator C2 Interrupt
- CCP1 Event Interrupt
- CCP2 Event Interrupt
- CCP3 Event Interrupt
- CCP4 Event Interrupt
- CCP5 Event Interrupt
- MSSP Event Interrupt
- MSSP Bus Collision Interrupt

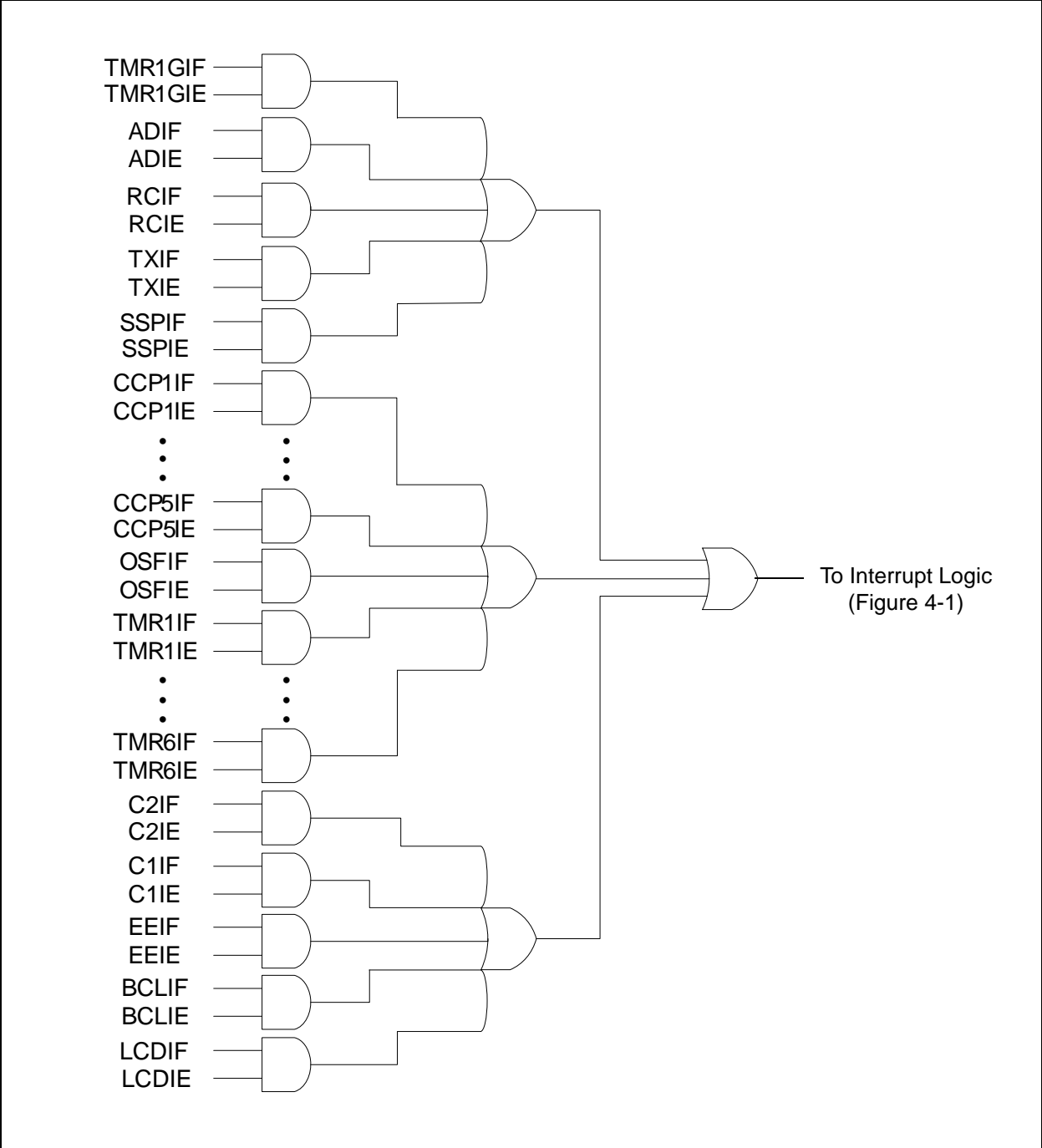
A block diagram of the interrupt logic is shown in Figure 4-1.

**FIGURE 4-1: INTERRUPT LOGIC**



# PIC16F193X/LF193X

FIGURE 4-2: PERIPHERAL INTERRUPT LOGIC





## 4.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1, PIE2 and PIE3 registers)

The INTCON, PIR1, PIR2 and PIR3 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- PC is loaded with the interrupt vector 0004h

The ISR determines the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt

that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The RETFIE instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

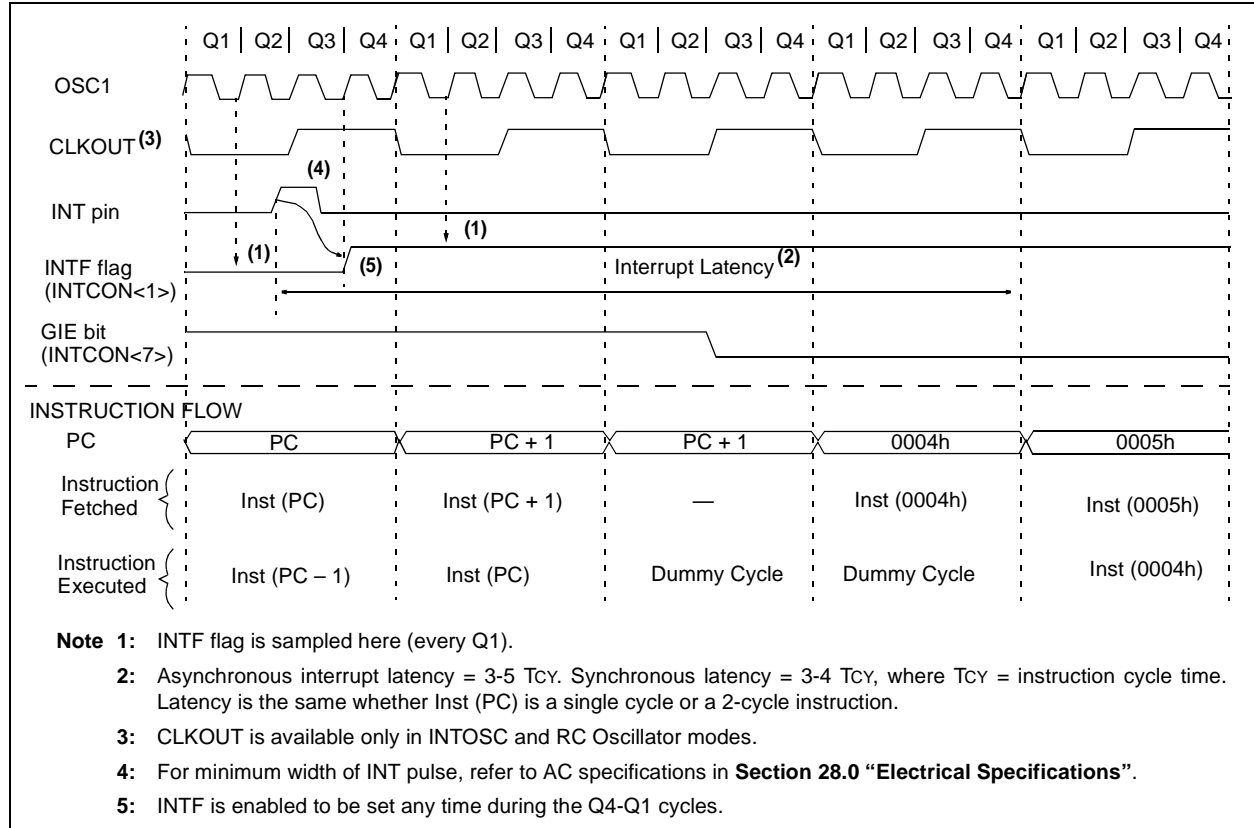
**Note 1:** Individual interrupt flag bits are set, regardless of the state of any other enable bits.

**2:** All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

## 4.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is 3 or 4 instruction cycles. For asynchronous interrupts, the latency is 3 to 5 instruction cycles, depending on when the interrupt occurs. See Figure 4-3 for timing details.

**FIGURE 4-3: INT PIN INTERRUPT TIMING**



# PIC16F193X/LF193X

---

## 4.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to the **Section 24.0 “Power-Down Mode (Sleep)”** for more details.

## 4.4 INT Pin

The external interrupt, INT pin, causes an asynchronous, edge-triggered interrupt. The INTEDG bit of the OPTION register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector. This interrupt is disabled by clearing the INTE bit of the INTCON register.

## 4.5 Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for  $\overline{TO}$  and  $\overline{PD}$ )
- BSR register
- FSR registers
- PCLATH register

Upon exit from the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. Depending on the user's application, other registers may also need to be saved.

## 4.5.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, interrupt-on-change and external INT 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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 4-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF <sup>(1)</sup>	INTF	IOCF
bit 7							bit 0

**Legend:**

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

- bit 7      **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      **TMR0IE:** Timer0 Overflow Interrupt Enable bit  
1 = Enables the Timer0 interrupt  
0 = Disables the Timer0 interrupt
- bit 4      **INTE:** RB0/INT External Interrupt Enable bit  
1 = Enables the RB0/INT external interrupt  
0 = Disables the RB0/INT external interrupt
- bit 3      **IOCF:** Interrupt-on-Change Enable bit<sup>(1)</sup>  
1 = Enables the interrupt-on-change  
0 = Disables the interrupt-on-change
- bit 2      **TMR0IF:** Timer0 Overflow Interrupt Flag bit<sup>(2)</sup>  
1 = TMR0 register has overflowed (must be cleared in software)  
0 = TMR0 register did not overflow
- bit 1      **INTF:** INT External Interrupt Flag bit  
1 = The INT external interrupt occurred (must be cleared in software)  
0 = The INT external interrupt did not occur
- bit 0      **IOCF:** Interrupt-on-Change Interrupt Flag bit  
1 = When at least one of the interrupt-on-change pins changed state (must be cleared in software)  
0 = None of the interrupt-on-change pins have changed state

**Note 1:** TMR0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing TMR0IF bit.

# PIC16F193X/LF193X

## 4.5.2 PIE1 REGISTER

The PIE1 register contains the interrupt enable bits, as shown in Register 4-2.

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

### REGISTER 4-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

- bit 7      **TMR1GIE:** Timer1 Gate Interrupt Enable bit  
1 = Enable the Timer1 Gate Acquisition complete interrupt  
0 = Disable the Timer1 Gate Acquisition complete interrupt
- bit 6      **ADIE:** A/D Converter (ADC) Interrupt Enable bit  
1 = Enables the ADC interrupt  
0 = Disables the ADC interrupt
- bit 5      **RCIE:** USART Receive Interrupt Enable bit  
1 = Enables the USART receive interrupt  
0 = Disables the USART receive interrupt
- bit 4      **TXIE:** USART Transmit Interrupt Enable bit  
1 = Enables the USART transmit interrupt  
0 = Disables the USART transmit interrupt
- bit 3      **SSPIE:** Synchronous Serial Port (SSP) Interrupt Enable bit  
1 = Enables the SSP interrupt  
0 = Disables the SSP interrupt
- bit 2      **CCP1IE:** CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt
- bit 1      **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit  
1 = Enables the Timer2 to PR2 match interrupt  
0 = Disables the Timer2 to PR2 match interrupt
- bit 0      **TMR1IE:** Timer1 Overflow Interrupt Enable bit  
1 = Enables the Timer1 overflow interrupt  
0 = Disables the Timer1 overflow interrupt

## 4.5.3 PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 4-3.

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

**REGISTER 4-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2**

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

**Legend:**

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

- bit 7      **OSFIE:** Oscillator Fail Interrupt Enable bit  
           1 = Enables the Oscillator Fail interrupt  
           0 = Disables the Oscillator Fail interrupt
  
- bit 6      **C2IE:** Comparator C2 Interrupt Enable bit  
           1 = Enables the Comparator C2 interrupt  
           0 = Disables the Comparator C2 interrupt
  
- bit 5      **C1IE:** Comparator C1 Interrupt Enable bit  
           1 = Enables the Comparator C1 interrupt  
           0 = Disables the Comparator C1 interrupt
  
- bit 4      **EEIE:** EEPROM Write Completion Interrupt Enable bit  
           1 = Enables the EEPROM Write Completion interrupt  
           0 = Disables the EEPROM Write Completion interrupt
  
- bit 3      **BCLIE:** MSSP Bus Collision Interrupt Enable bit  
           1 = Enables the MSSP Bus Collision Interrupt  
           0 = Disables the MSSP Bus Collision Interrupt
  
- bit 2      **LCDIE:** LCD Module Interrupt Enable bit  
           1 = Enables the LCD module interrupt  
           0 = Disables the LCD module interrupt
  
- bit 1      **Unimplemented:** Read as '0'
  
- bit 0      **CCP2IE:** CCP2 Interrupt Enable bit  
           1 = Enables the CCP2 interrupt  
           0 = Disables the CCP2 interrupt

# PIC16F193X/LF193X

## 4.5.4 PIE3 REGISTER

The PIE3 register contains the interrupt enable bits, as shown in Register 4-4.

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

### REGISTER 4-4: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

- bit 7      **Unimplemented:** Read as '0'
- bit 6      **CCP5IE:** CCP5 Interrupt Enable bit  
1 = Enables the CCP5 interrupt  
0 = Disables the CCP5 interrupt
- bit 5      **CCP4IE:** CCP4 Interrupt Enable bit  
1 = Enables the CCP4 interrupt  
0 = Disables the CCP4 interrupt
- bit 4      **CCP3IE:** CCP3 Interrupt Enable bit  
1 = Enables the CCP3 interrupt  
0 = Disables the CCP3 interrupt
- bit 3      **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit  
1 = Enables the TMR6 to PR6 Match interrupt  
0 = Disables the TMR6 to PR6 Match interrupt
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit  
1 = Enables the TMR4 to PR4 Match interrupt  
0 = Disables the TMR4 to PR4 Match interrupt
- bit 0      **Unimplemented:** Read as '0'

## 4.5.5 PIR1 REGISTER

The PIR1 register contains the interrupt flag bits, as shown in Register 4-5.

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

### REGISTER 4-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

**Legend:**

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

- bit 7      **TMR1GIF:** Timer1 Gate Interrupt Flag bit  
             1 = Timer1 Gate is inactive  
             0 = Timer1 Gate is active
  
- bit 6      **ADIF:** A/D Converter Interrupt Flag bit  
             1 = A/D conversion complete (must be cleared in software)  
             0 = A/D conversion has not completed or has not been started
  
- 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:** Timer2 to PR2 Interrupt Flag bit  
             1 = A Timer2 to PR2 match occurred (must be cleared in software)  
             0 = No Timer2 to PR2 match occurred
  
- bit 0      **TMR1IF:** Timer1 Overflow Interrupt Flag bit  
             1 = The TMR1 register overflowed (must be cleared in software)  
             0 = The TMR1 register did not overflow

# PIC16F193X/LF193X

## 4.5.6 PIR2 REGISTER

The PIR2 register contains the interrupt flag bits, as shown in Register 4-6.

**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 of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 4-6: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

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

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

- bit 7      **OSFIF:** Oscillator Fail Interrupt Flag  
1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)  
0 = No oscillator failure has been detected
- bit 6      **C2IF:** Comparator C2 Interrupt Flag  
1 = An enabled edge was detected on Comparator C2 (must be cleared in software)  
0 = No enabled edge was detected on Comparator C2
- bit 5      **C1IF:** Comparator C1 Interrupt Flag  
1 = An enabled edge was detected on Comparator C1 (must be cleared in software)  
0 = No enabled edge was detected on Comparator C1
- bit 4      **EEIF:** EEPROM Write Completion Interrupt Flag bit  
1 = The EEPROM Write operation has completed (must be cleared in software)  
0 = The EEPROM Write operation has not completed or has not been started
- bit 3      **BCLIF:** MSSP Bus Collision Interrupt Flag bit  
1 = A Bus Collision was detected (must be cleared in software)  
0 = No Bus collision was detected
- bit 2      **LCDIF:** LCD Module Interrupt Flag bit  
1 = The LCD module has completed displaying a frame (must be cleared in software).  
0 = The LCD module has not completed displaying a frame
- bit 1      **Unimplemented:** Read as '0'
- bit 0      **CCP2IF:** CCP2 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



## 4.5.7 PIR3 REGISTER

The PIR3 register contains the interrupt enable bits, as shown in Register 4-7.

**REGISTER 4-7: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—
bit 7							bit 0

**Legend:**

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

bit 7	<b>Unimplemented:</b> Read as '0'
bit 6	<p><b>CCP5IF:</b> CCP5 Interrupt Flag bit</p> <p><u>Capture Mode</u></p> <p>1 = A TMR1 register capture occurred (must be cleared in software)</p> <p>0 = No TMR1 register capture occurred</p> <p><u>Compare Mode</u></p> <p>1 = A TMR1 register compare match occurred (must be cleared in software)</p> <p>0 = No TMR1 register compare match occurred</p> <p><u>PWM Mode</u></p> <p>Unused in this mode</p>
bit 5	<p><b>CCP4IF:</b> CCP4 Interrupt Flag bit</p> <p><u>Capture Mode</u></p> <p>1 = A TMR1 register capture occurred (must be cleared in software)</p> <p>0 = No TMR1 register capture occurred</p> <p><u>Compare Mode</u></p> <p>1 = A TMR1 register compare match occurred (must be cleared in software)</p> <p>0 = No TMR1 register compare match occurred</p> <p><u>PWM Mode</u></p> <p>Unused in this mode</p>
bit 4	<p><b>CCP3IF:</b> CCP3 Interrupt Flag bit</p> <p><u>Capture Mode</u></p> <p>1 = A TMR1 register capture occurred (must be cleared in software)</p> <p>0 = No TMR1 register capture occurred</p> <p><u>Compare Mode</u></p> <p>1 = A TMR1 register compare match occurred (must be cleared in software)</p> <p>0 = No TMR1 register compare match occurred</p> <p><u>PWM Mode</u></p> <p>Unused in this mode</p>
bit 3	<p><b>TMR6IF:</b> TMR6 to PR6 Match Interrupt Flag bit</p> <p>1 = TMR6 to PR6 post-scaled match occurred (must be cleared in software)</p> <p>0 = No TMR6 to PR6 match occurred</p>
bit 2	<b>Unimplemented:</b> Read as '0'
bit 1	<p><b>TMR4IF:</b> TMR4 to PR4 Match Interrupt Flag bit</p> <p>1 = TMR4 to PR4 post-scaled match occurred (must be cleared in software)</p> <p>0 = No TMR4 to PR4 match occurred</p>
bit 0	<b>Unimplemented:</b> Read as '0'

# PIC16F193X/LF193X

**TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	51
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	79

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

## 5.0 LOW DROPOUT (LDO) VOLTAGE REGULATOR

The PIC16F193X devices differ from the PIC16LF193X devices due to an internal Low Dropout (LDO) voltage regulator. The PIC16F193X contain an internal LDO, while the PIC16LF193X do not.

The lithography of the die allows a maximum operating voltage of 3.6V on the internal digital logic. In order to continue to support 5.0V designs, a LDO voltage regulator is integrated on the die. The LDO voltage regulator allows for the internal digital logic to operate at 3.2V, while I/O's operate at 5.0V ( $V_{DD}$ ).

The LDO voltage regulator requires an external bypass capacitor for stability. One of three pins, denoted as VCAP, can be configured for the external bypass capacitor. It is recommended that the capacitor be a ceramic cap between 0.1 to 1.0  $\mu\text{F}$ .

On power-up, the external capacitor will look like a large load on the LDO voltage regulator. To prevent erroneous operation, the device is held in Reset while a constant current source charges the external capacitor. After the cap is fully charged, the device is released from Reset. For more information, refer to **Section 28.0 "Electrical Specifications"**.

See Configuration Word 2 register (Register 10-2) for VCAP enable bits.

# PIC16F193X/LF193X

---

NOTES:

## 6.0 I/O PORTS

Depending on the device selected and peripherals enabled, there are up to five ports available. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

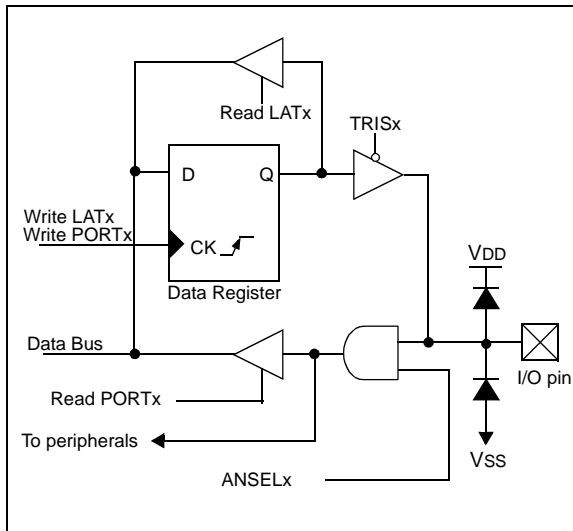
Each port has three registers for its operation. These registers are:

- TRISx registers (data direction register)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

Ports with analog functions also have an ANSELx register which can disable the digital input and save power. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 6-1.

**FIGURE 6-1: GENERIC I/O PORT OPERATION**



# PIC16F193X/LF193X

## 6.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 6-1. For this device family, the following functions can be moved between different pins.

- $\overline{SS}$  (Slave Select)
- CCP2
- CCP3
- Timer1 Gate
- SR Latch SRnQ output
- Comparator C2 output

These bits have no effect on the values of any TRIS register. PORT and TRIS overrides will be routed to the correct pin. The unselected pin will be unaffected.

**REGISTER 6-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER**

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CCP3SEL	T1GSEL	P2BSEL	SRnQSEL	C2OUTSEL	SSSEL	CCP2SEL
bit 7							bit 0

**Legend:**

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

- bit 7 **Unimplemented:** Read as '0'.
- bit 6 **CCP3SEL:** CCP3 Input/Output Pin Selection bit  
For 28-Pin Devices (PIC16F1933/1936/1938):  
 0 = CCP3/P3A function is on RC6/TX/CK/CCP3/P3A/SEG9  
 1 = CCP3/P3A function is on RB5/AN13/CPS5/CCP3/P3A/T1G/COM1  
For 40-Pin Devices (PIC16F1934/1937/1939):  
 0 = CCP3/P3A function is on RE0/AN5/CCP3/P3A/SEG21  
 1 = CCP3/P3A function is on RB5/AN13/CPS5/CCP3/P3A/T1G/COM1
- bit 5 **T1GSEL:** Timer1 Gate Input Pin Selection bit  
 0 = T1G function is on RB5/AN13/CPS5/CCP3/P3A/T1G/COM1  
 1 = T1G function is on RC4/SDI/SDA/T1G/SEG11
- bit 4 **P2BSEL:** CCP2 PWM B Output Pin Selection bit  
For 28-Pin Devices (PIC16F1933/1936/1938):  
 0 = P2B function is on RC0/T1OSO/T1CKI/P2B  
 1 = P2B function is on RB5/AN13/P2B/CPS5/T1G/COM1  
For 40-Pin Devices (PIC16F1934/1937/1939):  
 0 = P2B function is on RC0/T1OSO/T1CKI/P2B  
 1 = P2B function is on RD2/CPS10/P2B
- bit 3 **SRnQSEL:** SR Latch nQ Output Pin Selection bit  
 0 = SRnQ function is on RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub>  
 1 = SRnQ function is on RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub>
- bit 2 **C2OUTSEL:** Comparator C2 Output Pin Selection bit  
 0 = C2OUT function is on RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub>  
 1 = C2OUT function is on RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub>
- bit 1 **SSSEL:**  $\overline{SS}$  Input Pin Selection bit  
 0 =  $\overline{SS}$  function is on RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub>  
 1 =  $\overline{SS}$  function is on RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub>
- bit 0 **CCP2SEL:** CCP2 Input/Output Pin Selection bit  
 0 = CCP2/P2A function is on RC1/T1OSI/CCP2/P2A  
 1 = CCP2/P2A function is on RB3/AN9/C12IN2-/CPS3/CCP2/P2A/VLCD3

## 6.2 PORTA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 6-4). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 6-1 shows how to initialize PORTA.

Reading the PORTA register (Register 6-2) 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 (LATA).

The TRISA register (Register 6-4) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should 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:** The ANSELA register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

### EXAMPLE 6-1: INITIALIZING PORTA

```
BANKSEL PORTA      ;
CLRF  PORTA        ;Init PORTA
BANKSEL LATA       ;Data Latch
CLRF  LATA         ;
BANKSEL ANSELA     ;
CLRF  ANSELA       ;digital I/O
BANKSEL TRISA      ;
MOVLW 0Ch          ;Set RA<3:2> as inputs
MOVWF TRISA        ;and set RA<7:4,1:0>
                  ;as outputs
```

### REGISTER 6-2: PORTA: PORTA REGISTER

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

**Legend:**

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

bit 7-0      **RA<7:0>**: PORTA I/O Value bits<sup>(1)</sup>

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

### REGISTER 6-3: LATA: PORTA DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
bit 7							bit 0

**Legend:**

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

bit 7-0      **LATA<7:0>**: PORTA Output Latch Value bits<sup>(1)</sup>

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

# PIC16F193X/LF193X

## 6.2.1 ANSELA REGISTER

The ANSELA register (Register 6-5) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

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

### REGISTER 6-4: TRISA: PORTA TRI-STATE REGISTER

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

#### Legend:

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

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

### REGISTER 6-5: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

#### Legend:

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

bit 7-6                      **Unimplemented**: Read as '0'  
bit 5-0                      **ANSA<5:0>**: Analog Select between Analog or Digital Function on pins RA<5:0>, respectively  
0 = Digital I/O. Pin is assigned to port or digital special function.  
1 = Analog input. Pin is assigned as analog input<sup>(1)</sup>. Digital input buffer disabled.

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



## 6.2.2 PORTA FUNCTIONS AND OUTPUT PRIORITIES

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

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

### RA0

1. VCAP (enabled by Configuration Word)
2. SEG12 (LCD)
3. SRNQ (SR Latch)
4. C2OUT (Comparator)
5. RA0

### RA1

1. SEG7 (LCD)
2. RA1

### RA2

1. COM2 (LCD)
2. DACOUT (DAC)
3. RA2

### RA3

1. COM3 (LCD), 28-pin only
2. SEG15 (LCD)
3. RA3

### RA4

1. SEG4 (LCD)
2. SRQ (SR Latch)
3. C1OUT (Comparator)
4. CCP5 (CCP), 28-pin only
5. RA4

### RA5

1. VCAP (enabled by Configuration Word)
2. SEG5 (LCD)
3. SRNQ (SR Latch)
4. C2OUT (Comparator)
5. RA5

### RA6

1. VCAP (enabled by Configuration Word)
2. OSC2 (enabled by Configuration Word)
3. CLKOUT (enabled by Configuration Word)
4. SEG1 (LCD)
5. RA6

### RA7

1. OSC1/CLKIN (enabled by Configuration Word)
2. SEG2 (LCD)
3. RA7

# PIC16F193X/LF193X

**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	137
ADCON1	ADFM	ADCS2	ADCS1	ADCS0	—	ADREF	ADREF1	ADREF0	138
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	86
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	84
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	148
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	148
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	180
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	181
CONFIG2 <sup>(1)</sup>	—	—	VCAPEN1	VCAPEN0	—	—	—	—	128
DACCON0	DACEN	DACLPS	DACOE	---	DACPSS1	DACPSS0	---	DACNSS	153
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	85
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	247
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	247
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	51
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	85
SRCON0	SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR	122
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	277
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86

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

**Note 1:** PIC16F193X only.

## 6.3 PORTB and TRISB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 6-9). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-2 shows how to initialize PORTB.

Reading the PORTB register (Register 6-6) 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 TRISB register (Register 6-9) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. Example 6-2 shows how to initialize PORTB.

### EXAMPLE 6-2: INITIALIZING PORTB

```
BANKSEL PORTB      ;
CLRF  PORTB        ;Init PORTB
BANKSEL ANSELB
CLRF  ANSELB       ;Make RB<7:0> digital
BANKSEL TRISB      ;
MOVLW B'11110000' ;Set RB<7:4> as inputs
                        ;and RB<3:0> as outputs
MOVWF TRISB        ;
```

**Note:** The ANSELB register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

### 6.3.1 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 (see Register 6-8). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the WPUEN bit of the OPTION register.

### 6.3.2 INTERRUPT-ON-CHANGE

All of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:0> enable or disable the interrupt function for each pin. The interrupt-on-change feature is disabled on a Power-on Reset. Reference **Section 7.0 "Interrupt-On-Change"** for more information.

# PIC16F193X/LF193X

## REGISTER 6-6: PORTB: PORTB REGISTER

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

### Legend:

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

bit 7-0                      **RB<7:0>**: PORTB I/O Pin bit  
1 = Port pin is > V<sub>IH</sub>  
0 = Port pin is < V<sub>IL</sub>

## REGISTER 6-7: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
bit 7							bit 0

### Legend:

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

bit 7-0                      **LATB<7:0>**: PORTB Output Latch Value bits<sup>(1)</sup>

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

## REGISTER 6-8: WPUB: WEAK PULL-UP PORTB REGISTER

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

### Legend:

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

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

**Note 1:** Global WPUEN bit of the OPTION register must be cleared for individual pull-ups to be enabled.  
**2:** The weak pull-up device is automatically disabled if the pin is in configured as an output.

## 6.3.3 ANSELB REGISTER

The ANSELB register (Register 6-10) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

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

### REGISTER 6-9: TRISB: PORTB TRI-STATE REGISTER

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

#### Legend:

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

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

### REGISTER 6-10: ANSELB: PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
bit 7							bit 0

#### Legend:

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

bit 7-6      **Unimplemented:** Read as '0'  
 bit 5-0      **ANSB<5:0>**: Analog Select between Analog or Digital Function on Pins RB<5:0>, respectively  
 0 = Digital I/O. Pin is assigned to port or digital special function.  
 1 = Analog input. Pin is assigned as analog input<sup>(1)</sup>. Digital input buffer disabled.

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

# PIC16F193X/LF193X

## 6.3.4 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each PORTB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

### RB0

1. SEG0 (LCD)
2. CCP4, 28-pin only
3. RB0

### RB1

1. P1C (ECCP1), 28-pin only
2. RB1

### RB2

1. P1B (ECCP1), 28-pin only
2. RB2

### RB3

1. CCP2/P2A
2. RB3

### RB4

1. COM0
2. P1D, 28-pin only
3. RB4

### RB5

1. COM1
2. P2B, 28-pin only
3. P3A
4. RB5

### RB6

1. ICSPCLK (Programming)
2. ICDCLK (enabled by Configuration Word)
3. SEG14 (LCD)
4. RB6

### RB7

1. ICSPDAT (Programming)
2. ICDDAT (enabled by Configuration Word)
3. SEG13 (LCD)
4. RB7

**TABLE 6-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	137
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	84
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	TOXCS	180
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	181
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	73
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	104
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	104
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	104
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	90
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	247
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	247
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	51
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	90
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	170
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	90

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

## 6.4 PORTC and TRISC Registers

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 6-13). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-3 shows how to initialize PORTC.

Reading the PORTC register (Register 6-11) 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 TRISC register (Register 6-13) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

### EXAMPLE 6-3: INITIALIZING PORTC

```
BANKSEL PORTC      ;
CLRF   PORTC       ;Init PORTC
BANKSEL TRISC      ;
MOVLW  B'00001100' ;Set RC<3:2> as inputs
MOVWF  TRISC       ;and set RC<7:4,1:0>
                          ;as outputs
```

The location of the CCP2 function is controlled by the CCP2SEL bit in the APFCON register (refer to Register 6-1).

### REGISTER 6-11: PORTC: PORTC REGISTER

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

#### Legend:

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

bit 7-0              **RC<7:0>**: PORTC General Purpose I/O Pin bits  
1 = Port pin is > V<sub>IH</sub>  
0 = Port pin is < V<sub>IL</sub>

### REGISTER 6-12: LATC: PORTC DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
bit 7							bit 0

#### Legend:

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

bit 7-0              **LATC<7:0>**: PORTC Output Latch Value bits<sup>(1)</sup>

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

# PIC16F193X/LF193X

---

## REGISTER 6-13: TRISC: PORTC TRI-STATE REGISTER

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

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

bit 7-0

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

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

0 = PORTC pin configured as an output



## 6.4.1 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

### RC0

1. T1OSO (Timer1 Oscillator)
2. P2B (CCP)
3. RC0

### RC1

1. T1OSI (Timer1 Oscillator)
2. P2A (CCP)
3. RC1

### RC1

1. SEG3 (LCD)
2. P1A (CCP)
3. RC2

### RC3

1. SEG6 (LCD)
2. SCL (MSSP)
3. SCK (MSSP)
4. RC3

### RC4

1. SEG11 (LCD)
2. SDA (MSSP)
3. RC4

### RC5

1. SEG10 (LCD)
2. SDL (MSSP)
3. RC5

### RC6

1. SEG9 (LCD)
2. TX (EUSART)
3. CK (EUSART)
4. P3A (CCP), 28-pin only
5. RC6

### RC7

1. SEG8 (LCD)
2. DT (EUSART)
3. P3B (CCP), 28 pin only
4. RC7

**TABLE 6-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	84
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	93
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	247
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	247
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	93
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	277
SSPSTAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	276
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	169
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	222
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94

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

# PIC16F193X/LF193X

## 6.5 PORTD and TRISD Registers

PORTD<sup>(1)</sup> is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISD (Register 6-16). Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-4 shows how to initialize PORTD.

Reading the PORTD register (Register 6-14) 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 TRISD register (Register 6-16) controls the PORTD pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISD register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

### EXAMPLE 6-4: INITIALIZING PORTD

```
BANKSEL PORTD      ;
CLRF  PORTD        ;Init PORTD
BANKSEL ANSELDD    ;
CLRF  ANSELDD      ;Make PORTD digital
BANKSEL TRISD      ;
MOVLW B'00001100' ;Set RD<3:2> as inputs
MOVWF TRISD        ;and set RD<7:4,1:0>
                   ;as outputs
```

**Note 1:** PORTD is available on PIC16F1936 and PIC16F1938 only.

### REGISTER 6-14: PORTD: PORTD REGISTER<sup>(1)</sup>

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
bit 7							bit 0

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

bit 7-0      **RD<7:0>**: PORTD General Purpose I/O Pin bits  
 1 = Port pin is > V<sub>IH</sub>  
 0 = Port pin is < V<sub>IL</sub>

**Note 1:** PORTD is not implemented on PIC16F1933/1936/1938 devices, read as '0'.

### REGISTER 6-15: LATD: PORTD DATA LATCH REGISTER

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

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

bit 7-0      **LATD<7:0>**: PORTD Output Latch Value bits<sup>(1,2)</sup>

**Note 1:** Writes to PORTD are actually written to corresponding LATD register. Reads from PORTD register is return of actual I/O pin values.  
**2:** PORTD implemented on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.

## 6.5.1 ANSELD REGISTER

The ANSELD register (Register 6-17) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELD bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELD bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode

will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

**Note:** The ANSELD register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

### REGISTER 6-16: TRISD: PORTD TRI-STATE REGISTER<sup>(1)</sup>

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
bit 7							bit 0

**Legend:**

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

bit 7-0 **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 1:** TRISD is not implemented on PIC16F1933/1936/1938 devices, read as '0'.  
**2:** PORTD implemented on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.

### REGISTER 6-17: ANSELD: PORTD ANALOG SELECT REGISTER<sup>(2)</sup>

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

**Legend:**

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

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

- Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.  
**2:** ANSELD register is not implemented on the PIC16F1933/1936/1938. Read as '0'.  
**3:** PORTD implemented on PIC16F1934/1937/1939/PIC16LF1934/1937/1939 devices only.

# PIC16F193X/LF193X

## 6.5.2 PORTD FUNCTIONS AND OUTPUT PRIORITIES

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

### RD0

1. COM3 (LCD)
2. RD0

### RD1

1. CCP4 (CCP)
2. RD1

### RD2

1. P2B (CCP)
2. RD2

### RD3

1. SEG16 (LCD)
2. P2C (CCP)
3. RD3

### RD4

1. SEG17 (LCD)
2. P2D (CCP)
3. RD4

### RD5

1. SEG18 (LCD)
2. P1B (CCP)
3. RD5

### RD6

1. SEG19 (LCD)
2. P1C (CCP)
3. RD6

### RD7

1. SEG20 (LCD)
2. P1D (CCP)
3. RD7

**TABLE 6-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD<sup>(1)</sup>**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	97
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	180
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	181
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	96
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	247
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	96
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	97

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

**Note 1:** These registers are not implemented on the PIC16F1933/1936/1938 devices, read as '0'.

## 6.6 PORTE and TRISE Registers

PORTE<sup>(1)</sup> is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). The exception is RE3, which is input only and its TRIS bit will always read as '1'. Example 6-5 shows how to initialize PORTE.

Reading the PORTE register (Register 6-18) 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. RE3 reads '0' when MCLRE = 1.

**Note 1:** RE<2:0> and TRISE<2:0> pins are available on PIC16F1936 and PIC16F1938 only.

### EXAMPLE 6-5: INITIALIZING PORTE

```
BANKSEL PORTE      ;
CLRF  PORTE        ;Init PORTE
BANKSEL ANSELE     ;
CLRF  ANSELE       ;digital I/O
BANKSEL TRISE      ;
MOVLW B'00001100' ;Set RE<3:2> as inputs
MOVWF TRISE        ;and set RE<1:0>
                   ;as outputs
```

### REGISTER 6-18: PORTE: PORTE REGISTER

U-0	U-0	U-0	U-0	R-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	RE3	RE2 <sup>(1)</sup>	RE1 <sup>(1)</sup>	RE0 <sup>(1)</sup>
bit 7							bit 0

#### Legend:

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

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

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

**Note 1:** RE<2:0> are not implemented on the PIC16F1933/1936/1938. Read as '0'.

### REGISTER 6-19: LATE: PORTE DATA LATCH REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	LATE3	LATE2	LATE1	LATE0
bit 7							bit 0

#### Legend:

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

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

bit 3-0      **LATE<3:0>:** PORTE Output Latch Value bits<sup>(1)</sup>

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

# PIC16F193X/LF193X

## REGISTER 6-20: WPUE: WEAK PULL-UP PORTE REGISTER

U-0	U-0	U-0	U-0	R/W-1/1	U-0	U-0	U-0
—	—	—	—	WPUE3	—	—	—
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

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

bit 3      **WPUE:** Weak Pull-up Register bit

1 = Pull-up enabled

0 = Pull-up disabled

bit 2-0      **Unimplemented:** Read as '0'

**Note 1:** Global WPUEN bit of the OPTION register must be cleared for individual pull-ups to be enabled.

**2:** The weak pull-up device is automatically disabled if the pin is in configured as an output.

## 6.6.1 ANSELE REGISTER

The ANSELE register (Register 6-22) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELE bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

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

The TRISE register (Register 6-21) controls the PORTE pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISE register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**Note:** The ANSELE register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

### REGISTER 6-21: TRISE: PORTE TRI-STATE REGISTER

U-0	U-0	U-0	U-0	R-1	R/W-1	R/W-1	R/W-1
—	—	—	—	TRISE3	TRISE2 <sup>(1)</sup>	TRISE1 <sup>(1)</sup>	TRISE0 <sup>(1)</sup>
bit 7							bit 0

**Legend:**

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

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

bit 3      **TRISE3:** RE3 Port Tri-state Control bit  
This bit is always '1' as RE3 is an input only

bit 2-0      **TRISE<2:0>:** RE<2:0> Tri-State Control bits<sup>(1)</sup>  
1 = PORTE pin configured as an input (tri-stated)  
0 = PORTE pin configured as an output

**Note 1:** TRISE<2:0> are not implemented on the PIC16F1933/1936/1938/PIC16LF1933/1936/1938. Read as '0'.

### REGISTER 6-22: ANSELE: PORTE ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1
—	—	—	—	—	ANSE2 <sup>(2)</sup>	ANSE1 <sup>(2)</sup>	ANSE0 <sup>(2)</sup>
bit 7							bit 0

**Legend:**

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

bit 7-3      **Unimplemented:** Read as '0'

bit 2-0      **ANSE<2:0>:** Analog Select between Analog or Digital Function on Pins RE<2:0>, respectively  
0 = Digital I/O. Pin is assigned to port or digital special function.  
1 = Analog input. Pin is assigned as analog input<sup>(1)</sup>. Digital input buffer disabled.

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

**Note 2:** ANSELE register is not implemented on the PIC16F1933/1936/1938/PIC16LF1933/1936/1938. Read as '0'

# PIC16F193X/LF193X

## 6.6.2 PORTE FUNCTIONS AND OUTPUT PRIORITIES

Each PORTE pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

### RE0

1. SEG21 (LCD)
2. CCP3/P3A (CCP)
3. RE0

### RE1

1. SEG22 (LCD)
2. P3B (CCP)
3. RE1

### RE2

1. SEG23 (LCD)
2. CCP5 (CCP)
3. RE2

**TABLE 6-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE<sup>(1)</sup>**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	137
ANSELE	—	—	—	—	—	ANSE2	ANSE1	ANSE0	101
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
LATE	—	—	—	—	LATE3	LATE2	LATE1	LATE0	99
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	247
PORTE	—	—	—	—	RE3	RE2	RE1	RE0	99
TRISE	—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0	101
WPUE	—	—	—	—	WPUE3	—	—	—	100

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

**Note 1:** These registers are not implemented on the PIC16F1933/1936/1938 devices, read as '0'.



## 7.0 INTERRUPT-ON-CHANGE

The PORTB pins can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual PORTB pin, or combination of PORTB pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 7-1 is a block diagram of the IOC module.

### 7.1 Enabling the Module

To allow individual PORTB pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

### 7.2 Individual Pin Configuration

For each PORTB pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated IOCBP<sub>x</sub> bit of the IOCBP register is set. To enable a pin to detect a falling edge, the associated IOCBN<sub>x</sub> bit of the IOCBN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCBP<sub>x</sub> bit and the IOCBN<sub>x</sub> bit of the IOCBP and IOCBN registers, respectively.

### 7.3 Interrupt Flags

The IOCBF<sub>x</sub> bits located in the IOCBF register are status flags that correspond to the Interrupt-on-change pins of PORTB. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCBF<sub>x</sub> bits.

### 7.4 Clearing Interrupt Flags

The individual status flags, (IOCBF<sub>x</sub> bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

#### EXAMPLE 7-1:

```
MOVLW  0xff
XORWF  IOCBF, W
ANDWF  IOCBF, F
```

### 7.5 Operation in Sleep

The Interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCBF register will be updated prior to the first instruction executed out of Sleep.

# PIC16F193X/LF193X

## REGISTER 7-1: IOCBP: INTERRUPT-ON-CHANGE POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0
bit 7							bit 0

### Legend:

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

bit 7-0                      **IOCBP<7:0>**: Interrupt-on-Change Positive Edge Enable bits  
1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.  
0 = Interrupt-on-Change disabled for the associated pin.

## REGISTER 7-2: IOCBN: INTERRUPT-ON-CHANGE NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
bit 7							bit 0

### Legend:

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

bit 7-0                      **IOCBN<7:0>**: Interrupt-on-Change Negative Edge Enable bits  
1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.  
0 = Interrupt-on-Change disabled for the associated pin.

## REGISTER 7-3: IOCBF: INTERRUPT-ON-CHANGE FLAG REGISTER

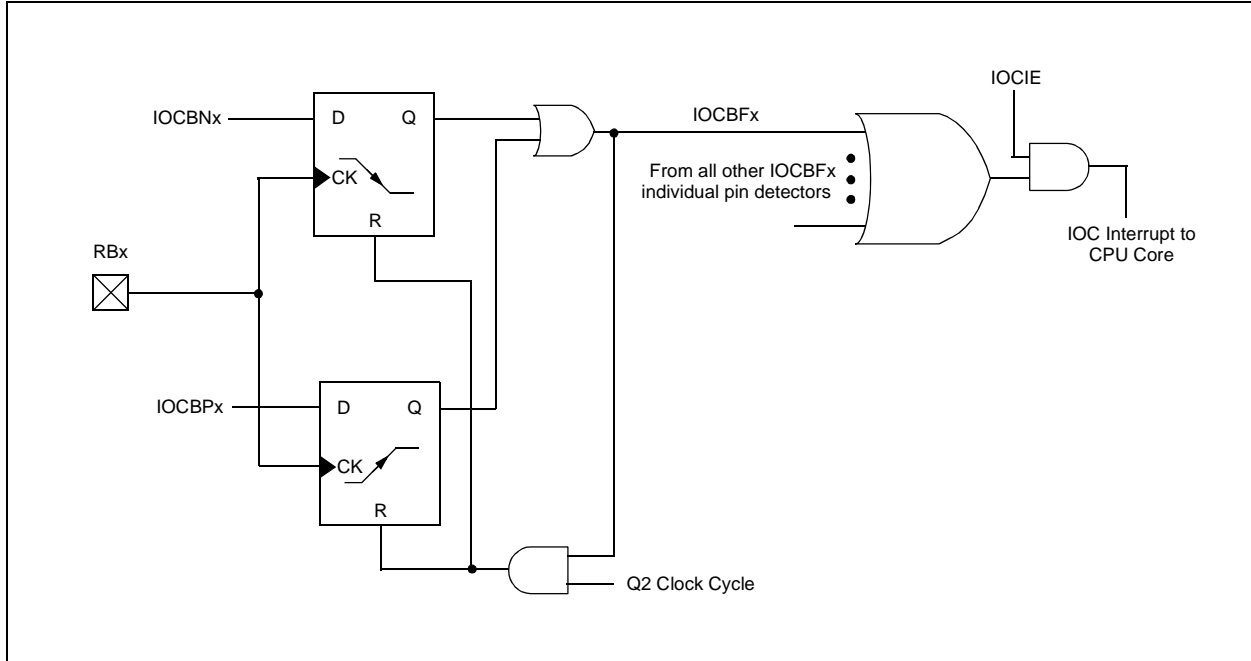
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0
bit 7							bit 0

### Legend:

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

bit 7-0                      **IOCBF<7:0>**: Interrupt-on-Change Flag bits  
1 = An enabled change was detected on the associated pin.  
Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.  
0 = No change was detected, or the user cleared the detected change.

**FIGURE 7-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM**



**TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91
INTCON	GIE	PEIE	TMR0IE	INTE	IOCBIE	TMR0IF	INTF	IOCIF	73
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	104
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	104
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	104
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91

**Legend:** x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by Interrupt-on-Change.

# PIC16F193X/LF193X

---

NOTES:

## 8.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

### 8.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 8-1 illustrates a block diagram of the oscillator module.

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 three internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

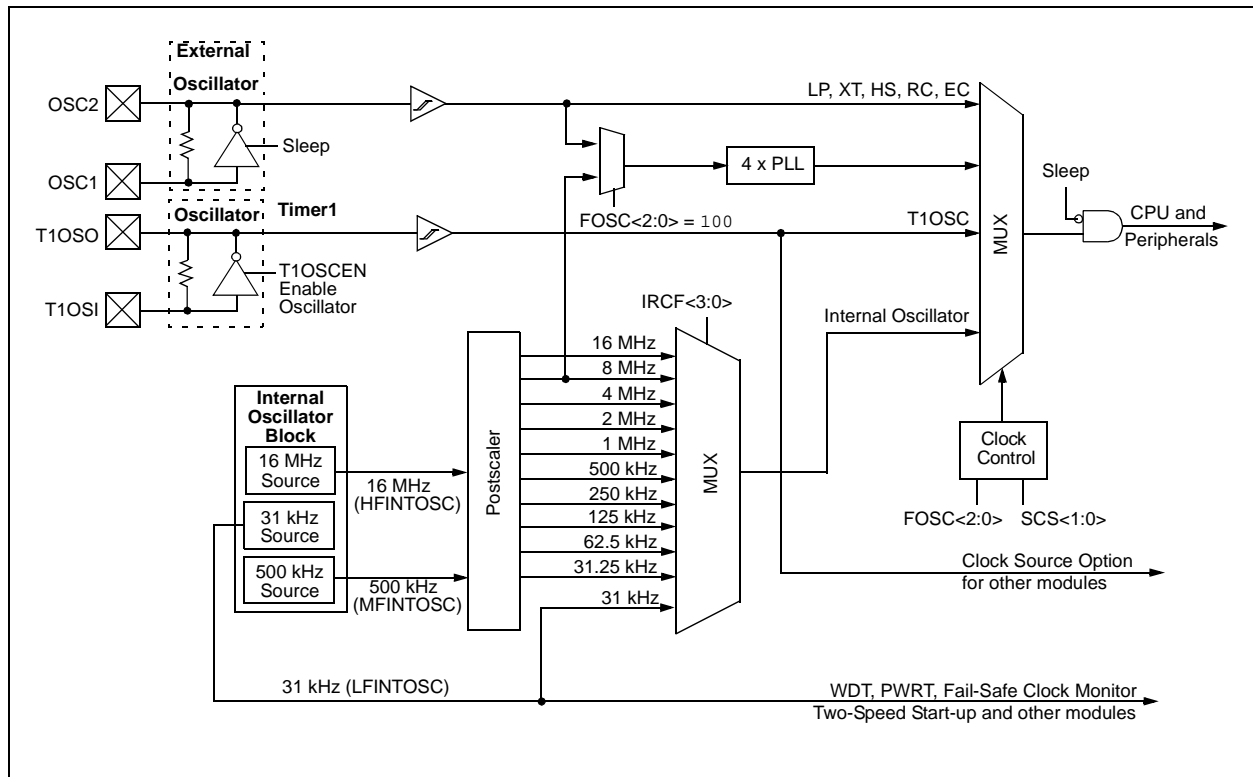
- Selectable system clock source between external or internal via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes)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

The oscillator module can be configured in one of six clock modes.

1. EC – External clock.
2. LP – 32 kHz Low-Power Crystal 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).
6. INTOSC – Internal oscillator.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1). The internal clock can be generated from two internal oscillators. The HFINTOSC is a calibrated high-frequency oscillator. The MFINTOSC is a calibrated medium-frequency oscillator. The LFINTOSC is an uncalibrated low-frequency oscillator.

**FIGURE 8-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM**



# PIC16F193X/LF193X

## 8.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 8-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- System clock select bits (SCS)
- Software PLL enable bit (SPLLEN)

**REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER**

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF3	IRCF2	IRCF1	IRCF0	—	SCS1	SCS0
bit 7							bit 0

**Legend:**

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

bit 7 **SPLLEN:** Software PLL Enable bit  
If PLEN = 1:  
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)

If PLEN = 0:  
1 = 4x PLL is enabled  
0 = 4x PLL is disabled

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits

000x = 31 kHz LF  
0010 = 31.25 kHz MF  
0011 = 31.25 kHz HF<sup>(2)</sup>  
0100 = 62.5 kHz MF  
0101 = 125 kHz MF  
0110 = 250 kHz MF  
0111 = 500 kHz MF (default upon Reset)  
1000 = 125 kHz HF<sup>(2)</sup>  
1001 = 250 kHz HF<sup>(2)</sup>  
1010 = 500 kHz HF<sup>(2)</sup>  
1011 = 1 MHz HF  
1100 = 2 MHz HF  
1101 = 4 MHz HF  
1110 = 8 MHz HF  
1111 = 16 MHz HF

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **SCS<1:0>:** System Clock Select bits

1x = Internal oscillator block  
01 = Timer1 oscillator  
00 = Clock determined by CONFIG1[FOSC<2:0>].

- Note 1:** Reset state depends on state of the IESO Configuration bit.  
**Note 2:** Duplicate frequency derived from HFINTOSC.

## 8.3 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 oscillator module. The oscillator module has two internal oscillators: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC), 500 kHz (MFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bit of the OSCCON register. See **Section 8.6 “Clock Switching”** for additional information. When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 8-1.

## 8.4 External Clock Modes

### 8.4.1 OSCILLATOR START-UP TIMER (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when 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 oscillator module.

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 8.6.3 “Timer1 Oscillator Ready (T1OSCR) Bit”**).

**TABLE 8-1: OSCILLATOR SWITCHING DELAYS**

Switch From	Switch To	Frequency	Oscillator Delay
Sleep/POR	LFINTOSC <sup>(1)</sup> MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	Oscillator Warm-up Delay (TWARM)
Sleep/POR	EC, RC <sup>(1)</sup>	DC – 32 MHz	2 cycles
LFINTOSC	EC, RC <sup>(1)</sup>	DC – 32 MHz	1 cycle of each
Sleep/POR	Timer1 Oscillator LP, XT, HS <sup>(1)</sup>	32 kHz-20 MHz	1024 Clock Cycles (OST)
Any clock source	MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs (approx.)
Any clock source	LFINTOSC <sup>(1)</sup>	31 kHz	1 cycle of each
Any clock source	Timer1 Oscillator	32 kHz	1024 Clock Cycles (OST)
PLL inactive	PLL active	16-32 MHz	2 ms (approx.)

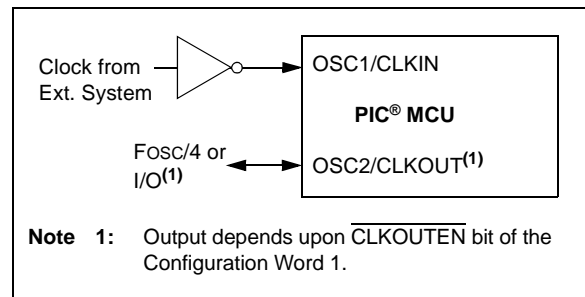
**Note 1:** PLL inactive.

### 8.4.2 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 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 8-2 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 PIC® MCU 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 8-2: EXTERNAL CLOCK (EC) MODE OPERATION**



# PIC16F193X/LF193X

## 8.4.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 8-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

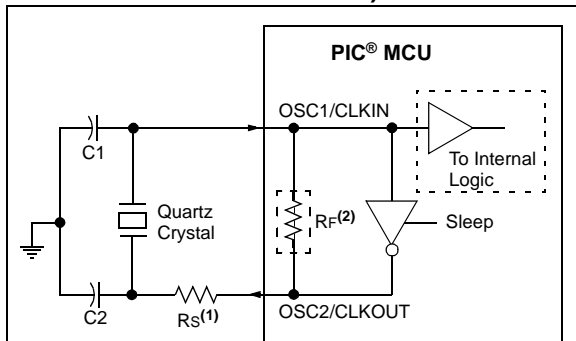
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

**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.

**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.

Figure 8-3 and Figure 8-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

**FIGURE 8-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**



- Note 1:** A series resistor ( $R_S$ ) may be required for quartz crystals with low drive level.
- 2:** The value of  $R_F$  varies with the Oscillator mode selected (typically between 2 M $\Omega$  to 10 M $\Omega$ ).

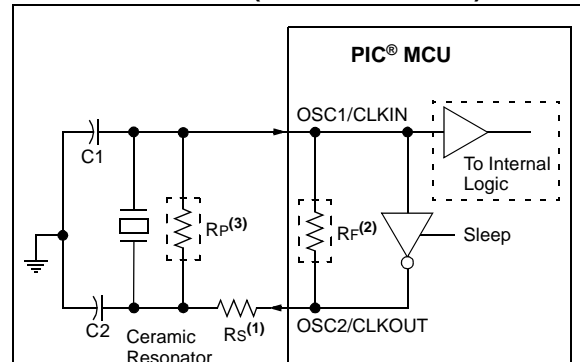
**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**2:** Always verify oscillator performance over the  $V_{DD}$  and temperature range that is expected for the application.

**3:** For oscillator design assistance, reference the following Microchip Applications Notes:

- AN826, "Crystal Oscillator Basics and Crystal Selection for rPIC<sup>®</sup> and PIC<sup>®</sup> Devices" (DS00826)
- AN849, "Basic PIC<sup>®</sup> Oscillator Design" (DS00849)
- AN943, "Practical PIC<sup>®</sup> Oscillator Analysis and Design" (DS00943)
- AN949, "Making Your Oscillator Work" (DS00949)

**FIGURE 8-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)**



- Note 1:** A series resistor ( $R_S$ ) may be required for ceramic resonators with low drive level.
- 2:** The value of  $R_F$  varies with the Oscillator mode selected (typically between 2 M $\Omega$  to 10 M $\Omega$ ).
- 3:** An additional parallel feedback resistor ( $R_P$ ) may be required for proper ceramic resonator operation.

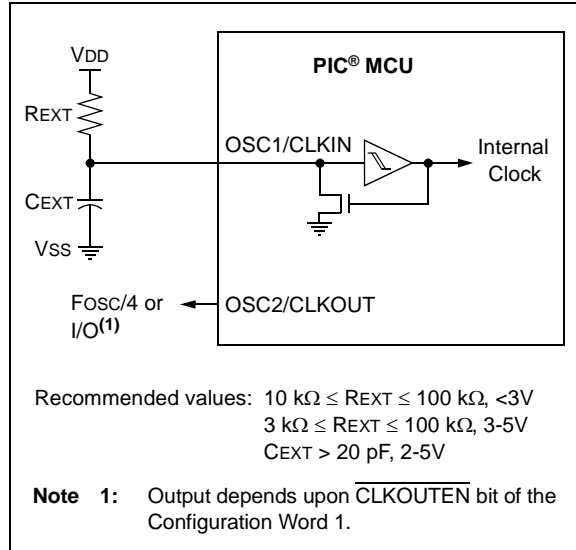


## 8.4.4 EXTERNAL RC MODE

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.

The RC circuit connects to OSC1. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 8-5 shows the external RC mode connections.

**FIGURE 8-5: EXTERNAL RC MODES**



The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{EXT}$ ) and capacitor ( $C_{EXT}$ ) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

## 8.5 Internal Clock Modes

The oscillator module has three 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 16 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 8-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 8-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits  $\text{IRCF}<2:0>$  of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bits of the OSCCON register. See **Section 8.6 "Clock Switching"** for more information.

### 8.5.1 INTOSC MODE

The INTOSC mode configures the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the  $\text{FOSC}<2:0>$  bits in the Configuration Word Register 1 (CONFIG1).

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

# PIC16F193X/LF193X

---

## 8.5.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 8-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 8-1). One of nine frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 8.5.7 “Frequency Select Bits (IRCF)”** for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits for the desired HF frequency (see Register 8-1), and
- FOSC<2:0> = 100, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

The High Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running and can be utilized.

The High Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

The High Frequency Internal Oscillator Status Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

## 8.5.3 MFINTOSC

The Medium-Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 8-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 8-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 8.5.7 “Frequency Select Bits (IRCF)”** for more information.

The MFINTOSC is enabled by:

- Configure the IRCF<3:0> bits for the desired HF frequency (see Register 8-1), and
- FOSC<2:0> = 100, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

The Medium Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running and can be utilized.

## 8.5.4 LFINTOSC

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

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 8-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See **Section 8.5.7 “Frequency Select Bits (IRCF)”** for more information. 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<3:0> bits of the OSCCON register = 000) as the system clock source (SCS bits of the OSCCON register = 1x), or when any of the following are enabled:

- Configure the IRCF<3:0> bits for the desired LF frequency (see Register 8-1), and
- FOSC<2:0> = 100, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

Peripherals that use the LFINTOSC are:

- LCD
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The Low Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running and can be utilized.

## 8.5.5 OSCSTAT REGISTER

The OSCSTAT register contains flags that represent the current status of the oscillators module.

### REGISTER 8-2: OSCSTAT: OSCILLATOR STATUS REGISTER

R-0/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
T1OSCR	PLLr	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

#### Legend:

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

- bit 7      **T1OSCR:** Timer1 Oscillator Ready bit  
 If the Timer1 oscillator is enabled (T1OSCEN = 1)  
     1 = Timer1 oscillator is ready and can be switched to  
     0 = Timer1 oscillator is not ready  
 else Timer1 oscillator is disabled - clock source is T1CKI  
     1 = Timer1 oscillator is always ready
- bit 6      **PLLr** 4x PLL Ready bit  
     1 = 4x PLL is ready and can be switched to  
     0 = 4x PLL oscillator is not ready
- bit 5      **OSTS:** Oscillator Start-up Time-out Status bit  
     1 = Device running from the clock defined by FOSC<3:0> of the CONFIG1 register  
     0 = Device running from the internal oscillator (HFINTOSC, MFINTOSC or LFINTOSC)
- bit 4      **HFIOFR:** High Frequency Internal Oscillator Ready bit  
     1 = 16 MHz Internal Oscillator (HFINTOSC) is ready and can be switched to  
     0 = 16 MHz Internal Oscillator (HFINTOSC) is not ready
- bit 3      **HFIOFL:** High Frequency Internal Oscillator Status Locked bit (2% Stable)  
     1 = 16 MHz Internal Oscillator (HFINTOSC) is in lock  
     0 = 16 MHz Internal Oscillator (HFINTOSC) has not yet locked
- bit 2      **MFIOFR:** Medium Frequency Internal Oscillator (500 kHz HFINTOSC Output) Ready bit  
     1 = 500 kHz Internal Oscillator (MFINTOSC) is ready and can be switched to  
     0 = 500 kHz Internal Oscillator (MFINTOSC) is not ready
- bit 1      **LFIOFR:** Low Frequency Internal Oscillator Ready bit  
     1 = 31 kHz Internal Oscillator (LFINTOSC) is ready and can be switched to  
     0 = 31 kHz Internal Oscillator (LFINTOSC) is not ready
- bit 0      **HFIOFS:** High Frequency Internal Oscillator Stable bit (0.5% Stable)  
     1 = 16 MHz Internal Oscillator (HFINTOSC) is in communications stable  
     0 = 16 MHz Internal Oscillator (HFINTOSC) is not yet communications stable

# PIC16F193X/LF193X

## 8.5.6 OSCTUNE REGISTER

The HFINTOSC and MFINTOSC are factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 8-3).

The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

The OSCTUNE register applies the same adjustment to both the HFINTOSC and the MFINTOSC simultaneously.

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 8-3: OSCTUNE: OSCILLATOR TUNING REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	
—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	
bit 7								bit 0

#### Legend:

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

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

bit 5-0 **TUN<4:0>:** Frequency Tuning bits

011111 = Maximum frequency

011110 =

•

•

•

000001 =

000000 = Oscillator module is running at the factory-calibrated frequency.

111111 =

•

•

•

100000 = Minimum frequency

## 8.5.7 FREQUENCY SELECT BITS (IRCF)

The output of the 16 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 8-1). The Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

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

**Note:** Following any Reset, the IRCF<2:0> bits of the OSCCON register 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.

## 8.5.8 INTERNAL OSCILLATOR CLOCK SWITCH TIMING

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 8-6). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

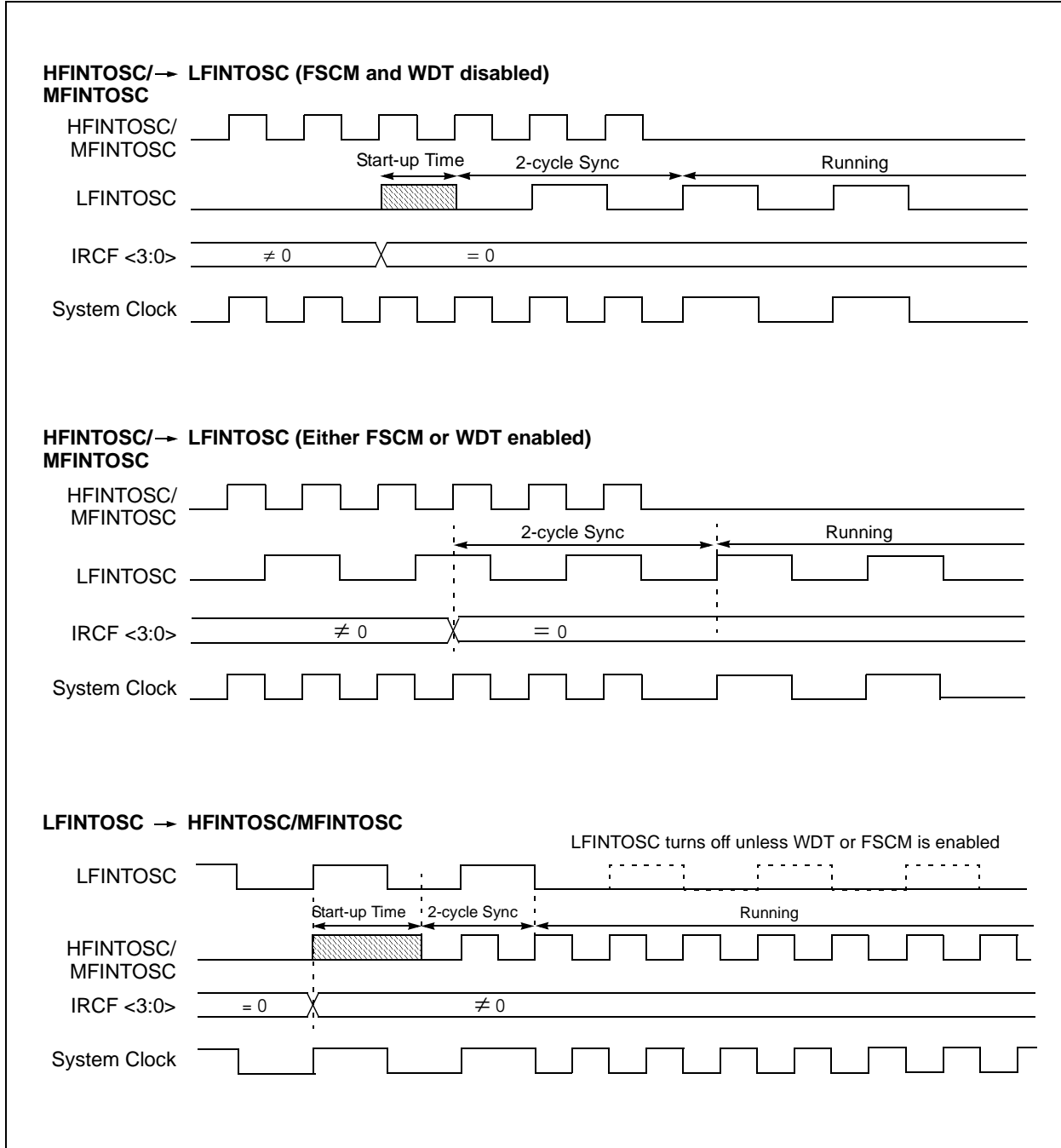
See Figure 8-6 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 8-1.

Start-up delay specifications are located in the oscillator tables of **Section 28.0 “Electrical Specifications”**.

# PIC16F193X/LF193X

**FIGURE 8-6: INTERNAL OSCILLATOR SWITCH TIMING**



## 8.6 Clock Switching

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

### 8.6.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1).
- When the SCS bits of the OSCCON register = 01, the system clock source is the Timer1 oscillator.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bit of the OSCCON register 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 bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source.

### 8.6.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1), 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.

### 8.6.3 TIMER1 OSCILLATOR READY (T1OSCR) BIT

The Timer1 Oscillator Ready (T1OSCR) bit of the OSCSTAT register indicates whether the Timer1 oscillator is ready to be used. After the T1OSCR bit is set, the SCS bits can be configured to select the Timer1 oscillator.

## 8.7 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 external oscillator to become stable.

**Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCSTAT register to remain clear.

When the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 8.4.1 “Oscillator Start-up Timer (OST)”**). The OST 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 of the OSCSTAT register is set, program execution switches to the external oscillator.

### 8.7.1 TWO-SPEED START-UP MODE CONFIGURATION

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

- IESO (of the Configuration Word Register 1) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 00.
- FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1) configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after Power-up Timer (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.

# PIC16F193X/LF193X

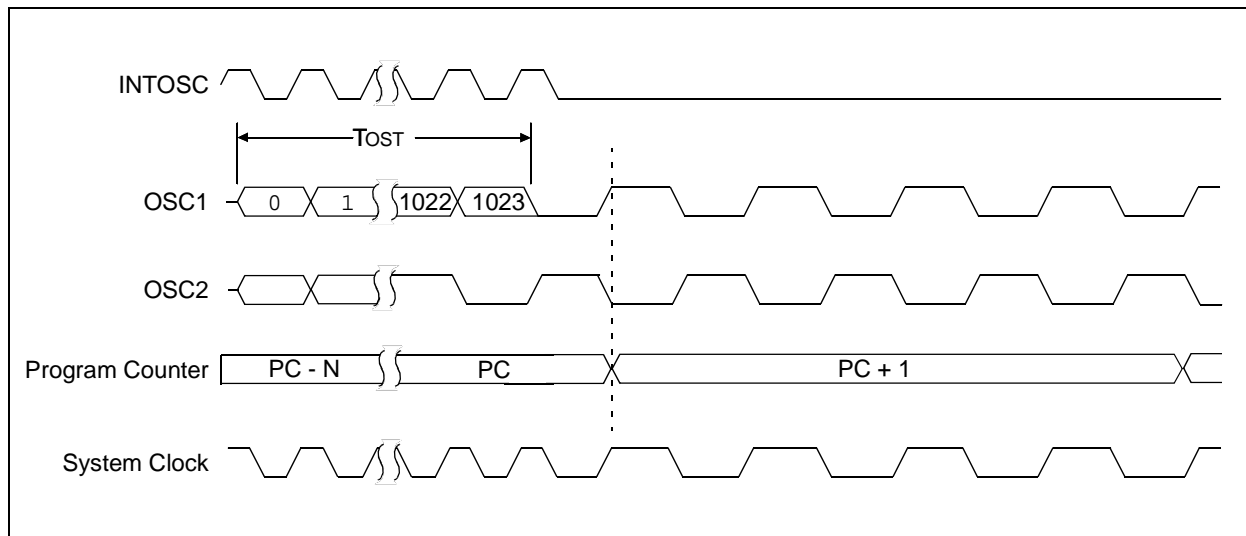
## 8.7.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<3:0> bits of the OSCCON register.
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.

## 8.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCSTAT register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1), or the internal oscillator.

**FIGURE 8-7: TWO-SPEED START-UP**

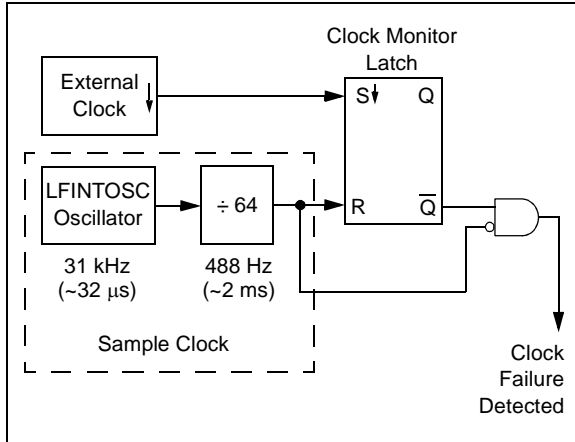




## 8.8 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word Register 1 (CONFIG1). The FSCM is applicable to all external Oscillator modes (LP, XT, HS, EC, Timer1 Oscillator and RC).

**FIGURE 8-8: FSCM BLOCK DIAGRAM**



### 8.8.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 8-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

### 8.8.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR2 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<3:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

### 8.8.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or changing the SCS bits of the OSCCON register. When the SCS bits are changed, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

### 8.8.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

**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 Status bits in the OSCSTAT register to verify the oscillator start-up and that the system clock switchover has successfully completed.

# PIC16F193X/LF193X

**FIGURE 8-9: FSCM TIMING DIAGRAM**



**TABLE 8-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG1 <sup>(2)</sup>	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	126
OSCCON	SPLLEN	IRCF3	IRCF2	IRCF1	IRCF0	—	SCS1	SCS0	108
OSCSTAT	T1OSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	113
OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	114
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	169

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

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

**2:** See Configuration Word Register 1 (Register 10-1) for operation of all register bits.

## 9.0 SR LATCH

The module consists of a single SR Latch with multiple Set and Reset inputs as well as separate latch outputs. The SR Latch module includes the following features:

- Programmable input selection
- SR Latch output is available internally/externally
- Separate Q and  $\bar{Q}$  outputs
- Firmware Set and Reset

### 9.1 Latch Operation

The latch is a Set-Reset latch that does not depend on a clock source. Each of the Set and Reset inputs are active-high. The latch can be Set or Reset by CxOUT, SRI pin, or variable clock. Additionally the SRPS and the SRPR bits of the SRCON0 register may be used to Set or Reset the SR Latch, respectively. The latch is Reset-dominant, therefore, if both Set and Reset inputs are high the latch will go to the Reset state. Both the SRPS and SRPR bits are self resetting which means that a single write to either of the bits is all that is necessary to complete a latch Set or Reset operation.

### 9.2 Latch Output

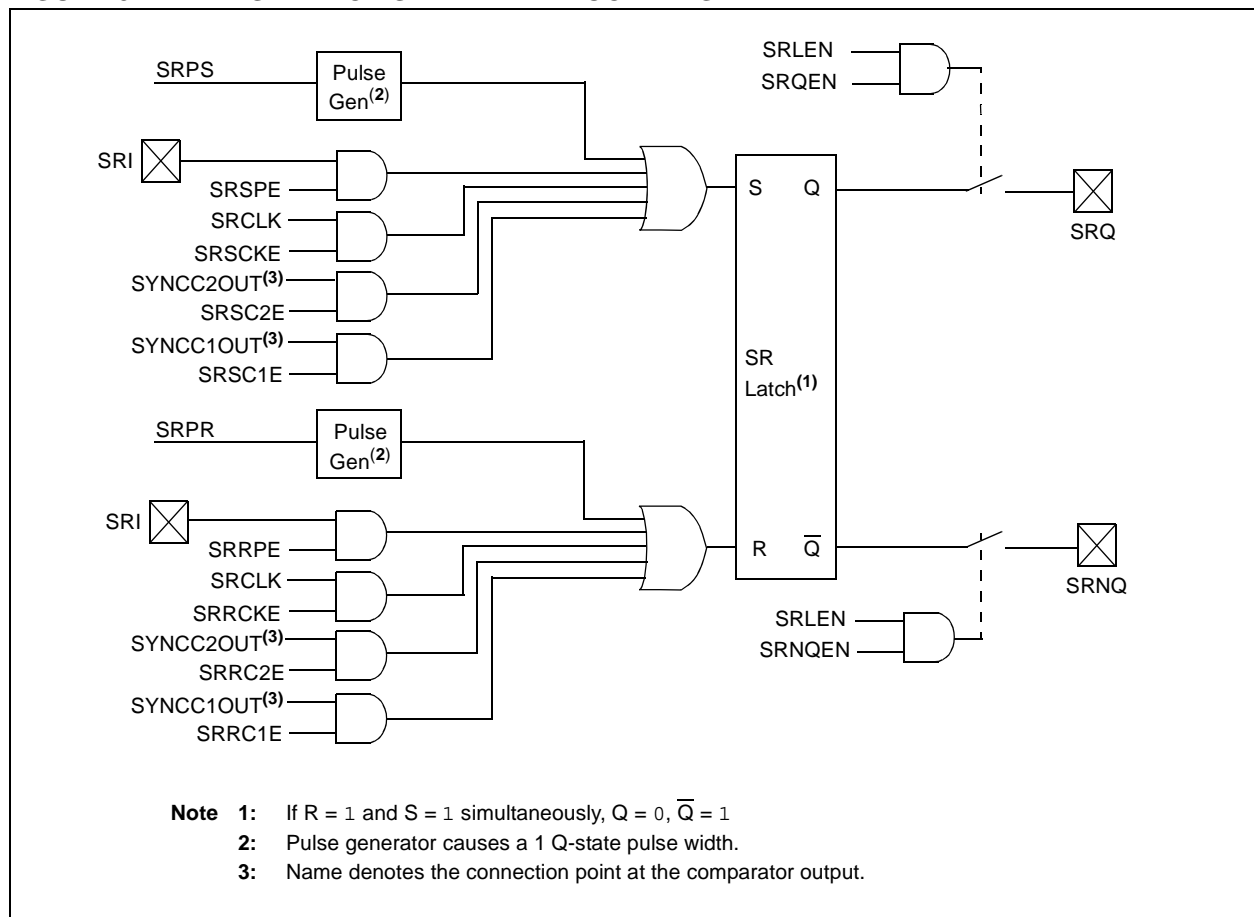
The SRQEN and SRNQEN bits of the SRCON0 register control the Q and  $\bar{Q}$  latch outputs. Both of the SR latch outputs may be directly output to an I/O pin at the same time.

The applicable TRIS bit of the corresponding port must be cleared to enable the port pin output driver.

### 9.3 Effects of a Reset

Upon any device Reset, the SR latch is not initialized. The user's firmware is responsible to initialize the latch output before enabling it to the output pins.

**FIGURE 9-1: SR LATCH SIMPLIFIED BLOCK DIAGRAM**



# PIC16F193X/LF193X

**TABLE 9-1: SRCLK FREQUENCY TABLE**

SRCLK	Divider	Fosc = 32 MHz	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	62.5 kHz	39.0 kHz	31.3 kHz	7.81 kHz	1.95 kHz
110	256	125 kHz	78.1 kHz	62.5 kHz	15.6 kHz	3.90 kHz
101	128	250 kHz	156 kHz	125 kHz	31.25 kHz	7.81 kHz
100	64	500 kHz	313 kHz	250 kHz	62.5 kHz	15.6 kHz
011	32	1 MHz	625 kHz	500 kHz	125 kHz	31.3 kHz
010	16	2 MHz	1.25 MHz	1 MHz	250 kHz	62.5 kHz
001	8	4 MHz	2.5 MHz	2 MHz	500 kHz	125 kHz
000	4	8 MHz	5 MHz	4 MHz	1 MHz	250 kHz

**REGISTER 9-1: SRCON0: SR LATCH CONTROL 0 REGISTER**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/S-0/0	R/S-0/0
SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR
bit 7						bit 0	

**Legend:**

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

bit 7                    **SRLEN:** SR Latch Enable bit  
1 = SR latch is enabled  
0 = SR latch is disabled

bit 6-4                **SRCLK<2:0>:** SR Latch Clock Divider bits  
000 = Generates a 1 FOSC wide pulse every 4th FOSC cycle clock  
001 = Generates a 1 FOSC wide pulse every 8th FOSC cycle clock  
010 = Generates a 1 FOSC wide pulse every 16th FOSC cycle clock  
011 = Generates a 1 FOSC wide pulse every 32th FOSC cycle clock  
100 = Generates a 1 FOSC wide pulse every 64th FOSC cycle clock  
101 = Generates a 1 FOSC wide pulse every 128th FOSC cycle clock  
110 = Generates a 1 FOSC wide pulse every 256th FOSC cycle clock  
111 = Generates a 1 FOSC wide pulse every 512th FOSC cycle clock

bit 3                    **SRQEN:** SR Latch Q Output Enable bit  
**If SRLEN = 1:**  
1 = Q is present on the SRQ pin  
0 = Q is internal only  
**If SRLEN = 0:**  
SR latch is disabled

bit 2                    **SRNQEN:** SR Latch  $\bar{Q}$  Output Enable bit  
**If SRLEN = 1:**  
1 =  $\bar{Q}$  is present on the SRnQ pin  
0 =  $\bar{Q}$  is internal only  
**If SRLEN = 0:**  
SR latch is disabled

bit 1                    **SRPS:** Pulse Set Input of the SR Latch bit  
1 = Pulse input for 1 Q-clock period  
0 = Do not generate pulse. Always reads back '0'.

bit 0                    **SRPR:** Pulse Reset Input of the SR Latch bit  
1 = Pulse input for 1 Q-clock period  
0 = Do not generate pulse. Always reads back '0'.

## REGISTER 9-2: SRCON1: SR LATCH CONTROL 1 REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SRSP E	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E
bit 7							bit 0

### Legend:

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

- bit 7      **SRSP E:** SR Latch Peripheral Set Enable bit  
1 = SRI pin status sets SR Latch  
0 = SRI pin status has no effect on SR Latch
- bit 6      **SRSCKE:** SR Latch Set Clock Enable bit  
1 = Set input of SR latch is pulsed with SRCLK  
0 = Set input of SR latch is not pulsed with SRCLK
- bit 5      **SRSC2E:** SR Latch C2 Set Enable bit  
1 = C2 Comparator output sets SR Latch  
0 = C2 Comparator output has no effect on SR Latch
- bit 4      **SRSC1E:** SR Latch C1 Set Enable bit  
1 = C1 Comparator output sets SR Latch  
0 = C1 Comparator output has no effect on SR Latch
- bit 3      **SRRPE:** SR Latch Peripheral Reset Enable bit  
1 = SRI pin resets SR Latch  
0 = SRI pin has no effect on SR Latch
- bit 2      **SRRCKE:** SR Latch Reset Clock Enable bit  
1 = Reset input of SR latch is pulsed with SRCLK  
0 = Reset input of SR latch is not pulsed with SRCLK
- bit 1      **SRRC2E:** SR Latch C2 Reset Enable bit  
1 = C2 Comparator output resets SR Latch  
0 = C2 Comparator output has no effect on SR Latch
- bit 0      **SRRC1E:** SR Latch C1 Reset Enable bit  
1 = C1 Comparator output resets SR Latch  
0 = C1 Comparator output has no effect on SR Latch

# PIC16F193X/LF193X

---

NOTES:

## 10.0 DEVICE CONFIGURATION

Device Configuration consists of Configuration Word 1 and Configuration Word 2 registers, Code Protection and Device ID.

### 10.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 register at 8007h and Configuration Word 2 register at 8008h.

# PIC16F193X/LF193X

## REGISTER 10-1: CONFIGURATION WORD 1

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN1	BOREN0	$\overline{\text{CPD}}$	$\overline{\text{CP}}$
bit 13						bit 7

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
MCLRE	$\overline{\text{PWRTÉ}}$	WDTE1	WDTE0	FOSC2	FOSC1	FOSC0
bit 6						bit 0

### Legend:

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

bit 13	<b>FCMEN:</b> Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor is enabled 0 = Fail-Safe Clock Monitor is disabled
bit 12	<b>IESO:</b> Internal External Switchover bit 1 = Internal/External Switchover mode is enabled 0 = Internal/External Switchover mode is disabled
bit 11	<b>CLKOUTEN:</b> Clock Out Enable bit 1 = CLKOUT function is disabled. I/O or oscillator function on RA6/CLKOUT 0 = CLKOUT function is enabled on RA6/CLKOUT
bit 10-9	<b>BOREN&lt;1:0&gt;:</b> Brown-out Reset Enable bits <sup>(1)</sup> 11 = BOR enabled 10 = BOR enabled during operation and disabled in Sleep 01 = BOR controlled by SBOREN bit of the PCON register 00 = BOR disabled
bit 8	<b>CPD:</b> Data Code Protection bit <sup>(2)</sup> 1 = Data memory code protection is disabled 0 = Data memory code protection is enabled
bit 7	<b>CP:</b> Code Protection bit <sup>(3)</sup> 1 = Program memory code protection is disabled 0 = Program memory code protection is enabled
bit 6	<b>MCLRE:</b> RE3/ $\overline{\text{MCLR}}$ /VPP Pin Function Select bit <u>If LVP bit = 1:</u> This bit is ignored. <u>If LVP bit = 0:</u> 1 = RE3/ $\overline{\text{MCLR}}$ /VPP pin function is $\overline{\text{MCLR}}$ ; Weak pull-up enabled. 0 = RE3/ $\overline{\text{MCLR}}$ /VPP pin function is digital input; $\overline{\text{MCLR}}$ internally disabled; Weak pull-up under control of WPUE3 bit..
bit 5	<b>PWRTÉ:</b> Power-up Timer Enable bit <sup>(1)</sup> 1 = PWRT disabled 0 = PWRT enabled
bit 4-3	<b>WDTE&lt;1:0&gt;:</b> Watchdog Timer Enable bit 11 = WDT enabled 10 = WDT enabled while running and disabled in Sleep 01 = WDT controlled by the SWDTEN bit in the WDTCON register 00 = WDT disabled

- 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 during an erase.
  - 3: The entire program memory will be erased when the code protection is turned off.



## REGISTER 10-1: CONFIGURATION WORD 1 (CONTINUED)

bit 2-0      **FOSC<2:0>**: Oscillator Selection bits

- 111 = ECH: External Clock, High-Power mode: CLKIN on RA7/OSC1/CLKIN
- 110 = ECM: External Clock, Medium-Power mode: CLKIN on RA7/OSC1/CLKIN
- 101 = ECL: External Clock, Low-Power mode: CLKIN on RA7/OSC1/CLKIN
- 100 = INTOSC oscillator: I/O function on RA7/OSC1/CLKIN
- 011 = EXTRC oscillator: RC function on RA7/OSC1/CLKIN
- 010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT pin and RA7/OSC1/CLKIN
- 001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT pin and RA7/OSC1/CLKIN
- 000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT pin and RA7/OSC1/CLKIN

- 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 during an erase.
  - 3: The entire program memory will be erased when the code protection is turned off.

# PIC16F193X/LF193X

## REGISTER 10-2: CONFIGURATION WORD 2

R/P-1/1	R/P-1/1	U-1	R/P-1/1	R/P-1/1	R/P-1/1	U-1
LVP	DEBUG	—	BORV	STVREN	PLLEN	—
bit 13						bit 7

U-1	R/P-1/1	R/P-1/1	U-1	U-1	R/P-1/1	R/P-1/1
—	VCAPEN1	VCAPEN0	—	—	WRT1	WRT0
bit 6						bit 0

### Legend:

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

- bit 13     **LVP:** Low-Voltage Programming Enable bit<sup>(1)</sup>  
 1 = Low-voltage programming enabled  
 0 = High-voltage on MCLR/VPP must be used for programming
- bit 12     **DEBUG:** In-Circuit Debugger Mode bit  
 1 = In-Circuit Debugger disabled, RB6/ICSPCLK and RB7/ICSPDAT are general purpose I/O pins  
 0 = In-Circuit Debugger enabled, RB6/ICSPCLK and RB7/ICSPDAT are dedicated to the debugger
- bit 11     **Unimplemented:** Read as '1'
- bit 10     **BORV:** Brown-out Reset Voltage Selection bit  
 1 = Brown-out Reset voltage set to 1.9V  
 0 = Brown-out Reset voltage set to 2.7V
- bit 9      **STVREN:** Stack Overflow/Underflow Reset Enable bit  
 1 = Stack Overflow or Underflow will cause a Reset  
 0 = Stack Overflow or Underflow will not cause a Reset
- bit 8      **PLLEN:** PLL Enable bit  
 1 = 4xPLL enabled  
 0 = 4xPLL disabled
- bit 7-6    **Unimplemented:** Read as '1'
- bit 5-4    **VCAPEN<1:0>:** Voltage Regulator Capacitor Enable bits<sup>(2)</sup>  
 00 = VCAP functionality is enabled on RA0  
 01 = VCAP functionality is enabled on RA5  
 10 = VCAP functionality is enabled on RA6  
 11 = No capacitor on VCAP pin
- bit 3-2    **Unimplemented:** Read as '1'
- bit 1-0    **WRT<1:0>:** Flash Memory Self-Write Protection bits  
**4 kW FLASH memory (PIC16F1933/PIC16LF1933 and PIC16F1934/PIC16LF1934 only):**  
 11 = Write protection off  
 10 = 000h to 1FFh write-protected, 200h to FFFh may be modified by EECON control  
 01 = 000h to 7FFh write-protected, 800h to FFFh may be modified by EECON control  
 00 = 000h to FFFh write-protected, no addresses may be modified by EECON control  
**8 kW FLASH memory (PIC16F1936/PIC16LF1936 and PIC16F1937/PIC16LF1937 only):**  
 11 = Write protection off  
 10 = 000h to 1FFh write-protected, 200h to 1FFFh may be modified by EECON control  
 01 = 000h to FFFh write-protected, 1000h to 1FFFh may be modified by EECON control  
 00 = 000h to 1FFFh write-protected, no addresses may be modified by EECON control  
**16 kW FLASH memory (PIC16F1938/PIC16LF1938 and PIC16F1939/PIC16LF1939 only):**  
 11 = Write protection off  
 10 = 000h to 1FFh write-protected, 200h to 3FFFh may be modified by EECON control  
 01 = 000h to 1FFFh write-protected, 2000h to 3FFFh may be modified by EECON control  
 00 = 000h to 3FFFh write-protected, no addresses may be modified by EECON control

**Note 1:** The LVP bit cannot be programmed to '0' when Programming mode is entered via LVP.

**Note 2:** Reads as '11' on PIC16LF193X only.

## 10.2 Code Protection

Code protection is controlled using the  $\overline{\text{CP}}$  bit in Configuration Word 1. When code protection is enabled, all program memory locations (0000h-7FFFh) read as all '0'. Further programming is disabled for the program memory (0000h-7FFFh).

Data memory is protected with its own Code-Protect bit ( $\overline{\text{CPD}}$ ). When data code protection is enabled ( $\overline{\text{CPD}} = 0$ ), all data memory locations read as '0'. Further programming is disabled for the data memory. Data memory can still be programmed and read during program execution.

The user ID locations and Configuration Words can be programmed and read out regardless of the code protection settings.

## 10.3 User ID

Four memory locations (8000h-8003h) 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 reported when using MPLAB<sup>®</sup> IDE. See the "*PIC16193X/PIC16LF193X Memory Programming Specification*" (DS41360A) for more information.

# PIC16F193X/LF193X

---

NOTES:

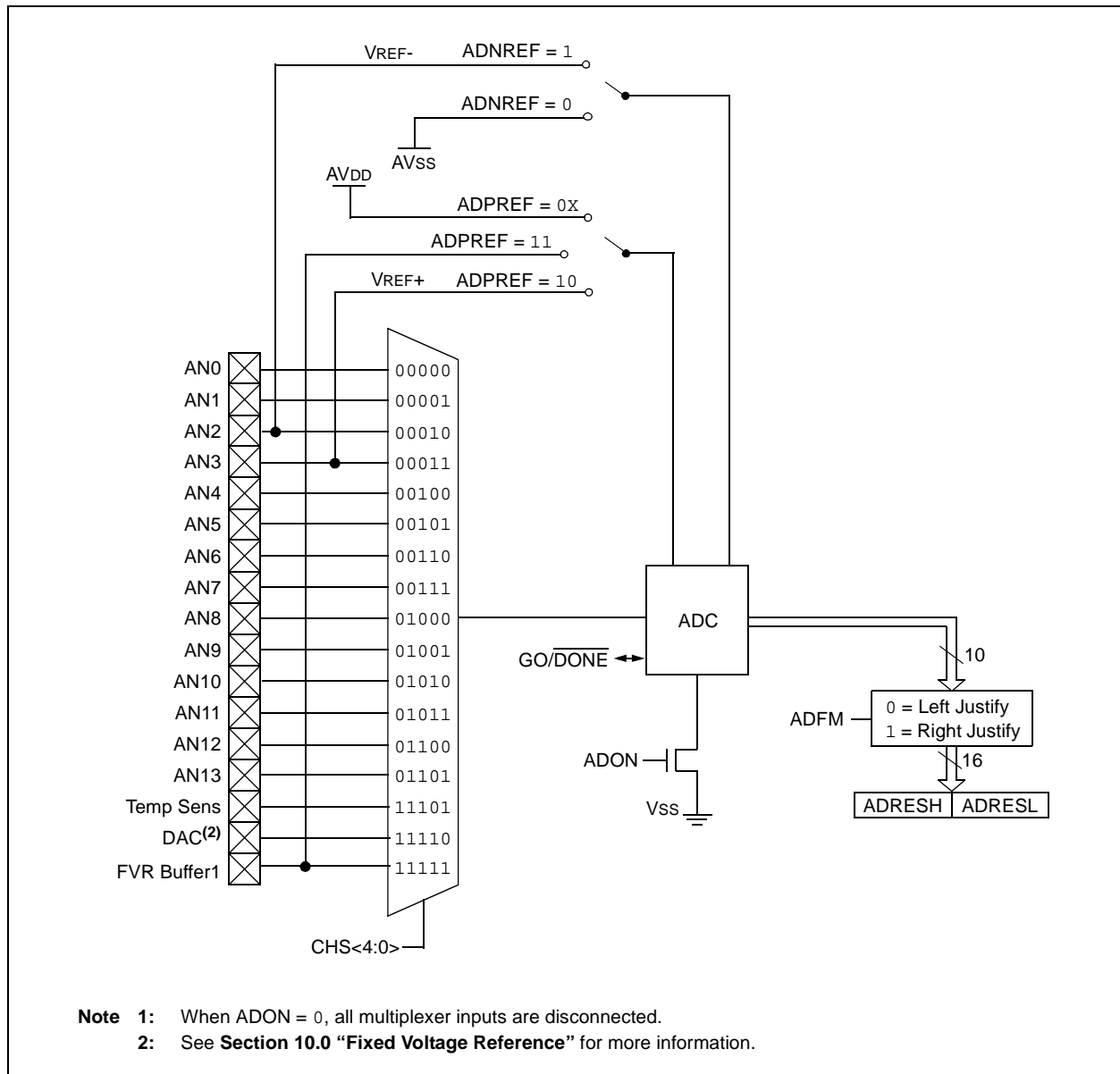
## 11.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result register (ADRES). Figure 11-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

**FIGURE 11-1: ADC BLOCK DIAGRAM**



# PIC16F193X/LF193X

---

## 11.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

### 11.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 6.0 “I/O Ports”** for more information.

**Note:** Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

### 11.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 11.2 “ADC Operation”** for more information.

### 11.1.3 ADC VOLTAGE REFERENCE

The ADPREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+
- AVDD
- FVR (Fixed Voltage Reference)

The ADNREF bits of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF-
- AVSS

See **Section 14.0 “Fixed Voltage Reference”** for more details on the fixed voltage reference.

### 11.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 11-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in **Section 28.0 “Electrical Specifications”** for more information. Table 11-1 gives examples of appropriate ADC clock selections.

**Note:** Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

**TABLE 11-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES**

ADC Clock Period (TAD)		Device Frequency (Fosc) Device Frequency (Fosc)					
ADC Clock Source	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5ns <sup>(2)</sup>	100 ns <sup>(2)</sup>	125 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs
Fosc/4	100	125 ns <sup>(2)</sup>	200 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 μs	4.0 μs
Fosc/8	001	0.5 μs <sup>(2)</sup>	400 ns <sup>(2)</sup>	0.5 μs <sup>(2)</sup>	1.0 μs	2.0 μs	8.0 μs <sup>(3)</sup>
Fosc/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs <sup>(3)</sup>
Fosc/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>
Fosc/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>
FRC	x11	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>

**Legend:** Shaded cells are outside of recommended range.

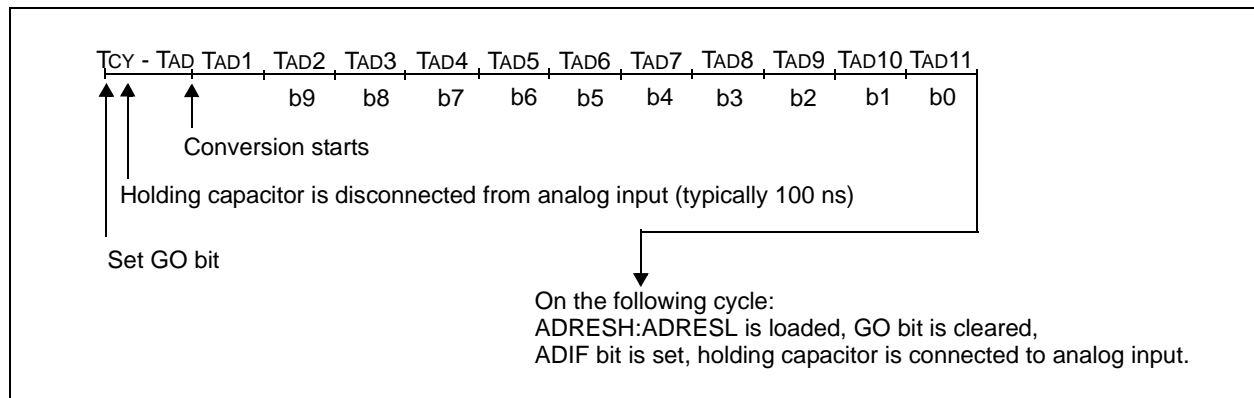
**Note 1:** The FRC source has a typical TAD time of 1.6 μs for VDD.

**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 FRC clock source is only recommended if the conversion will be performed during Sleep.

**FIGURE 11-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**



# PIC16F193X/LF193X

## 11.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

- Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
- 2:** The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

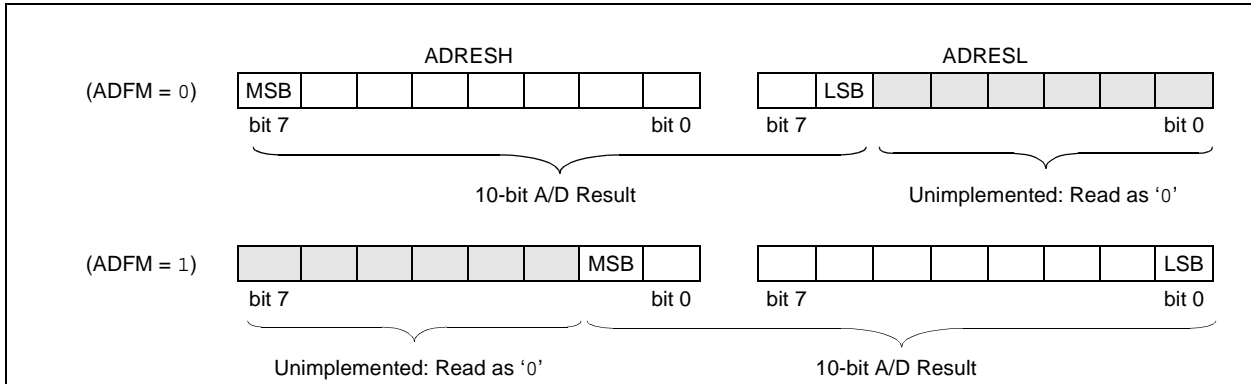
Please refer to **Section 11.1.5 “Interrupts”** for more information.

## 11.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 11-3 shows the two output formats.

**FIGURE 11-3: 10-BIT A/D CONVERSION RESULT FORMAT**





## 11.2 ADC Operation

### 11.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

**Note:** The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 11.2.6 “A/D Conversion Procedure”**.

### 11.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

### 11.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

**Note:** A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

### 11.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

### 11.2.5 SPECIAL EVENT TRIGGER

The Special Event Trigger of the CCP5 module allows periodic ADC measurements without software intervention. When this trigger occurs, the GO/DONE bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

Refer to **Section 19.0 “Capture/Compare/PWM Modules (ECCP1, ECCP2, ECCP3, CCP4, CCP5)”** for more information.

# PIC16F193X/LF193X

## 11.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
  - Disable pin output driver (Refer to the TRIS register)
  - Configure pin as analog (Refer to the ANSEL register)
2. Configure the ADC module:
  - Select ADC conversion clock
  - Configure voltage reference
  - Select ADC input channel
  - Turn on ADC module
3. Configure ADC interrupt (optional):
  - Clear ADC interrupt flag
  - Enable ADC interrupt
  - Enable peripheral interrupt
  - Enable global interrupt<sup>(1)</sup>
4. Wait the required acquisition time<sup>(2)</sup>.
5. Start conversion by setting the GO/DONE bit.
6. Wait for ADC conversion to complete by one of the following:
  - Polling the GO/DONE bit
  - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

**Note 1:** The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

**2:** Refer to **Section 11.3 “A/D Acquisition Requirements”**.

## EXAMPLE 11-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL    ADCON1    ;
MOVLW     B'01110000' ;ADC Frc clock
MOVWF     ADCON1    ;
BANKSEL    TRISA     ;
BSF       TRISA,0   ;Set RA0 to input
BANKSEL    ANSEL     ;
BSF       ANSEL,0   ;Set RA0 to analog
BANKSEL    ADCON0    ;
MOVLW     B'10000001' ;Right justify,
MOVWF     ADCON0    ;Vdd Vref, AN0, On
CALL     SampleTime ;Acquisition delay
BSF      ADCON0,GO  ;Start conversion
BTFSC    ADCON0,GO  ;Is conversion done?
GOTO     $-1        ;No, test again
BANKSEL    ADRESH    ;
MOVF     ADRESH,W   ;Read upper 2 bits
MOVWF    RESULTHI   ;store in GPR space
BANKSEL    ADRESL    ;
MOVF     ADRESL,W   ;Read lower 8 bits
MOVWF    RESULTLO   ;Store in GPR space
```

## 11.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

### REGISTER 11-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

#### Legend:

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

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

bit 6-2    **CHS<4:0>:** Analog Channel Select bits

00000 = AN0  
 00001 = AN1  
 00010 = AN2  
 00011 = AN3  
 00100 = AN4  
 00101 = AN5  
 00110 = AN6  
 00111 = AN7  
 01000 = AN8  
 01001 = AN9  
 01010 = AN10  
 01011 = AN11  
 01100 = AN12  
 01101 = AN13  
 01110 = Reserved. No channel connected.  
 .  
 .  
 .  
 11100 = Reserved. No channel connected.  
 11101 = Temperature Reference from band gap  
 11110 = DAC output (aka CVREF output)  
 11111 = Fixed Voltage Reference (FVR) Buffer 1 Output

bit 1      **GO/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:** ADC Enable bit

1 = ADC is enabled  
 0 = ADC is disabled and consumes no operating current

**Note 1:** See Section 10.0 "Fixed Voltage Reference" for more information.

# PIC16F193X/LF193X

## REGISTER 11-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS2	ADCS1	ADCS0	—	ADNREF	ADPREF1	ADPREF0
bit 7							bit 0

### Legend:

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

- bit 7            **ADFM:** A/D Result Format Select bit  
1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.  
0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.
- bit 6-4        **ADCS<2:0>:** A/D Conversion Clock Select bits  
000 = FOSC/2  
001 = FOSC/8  
010 = FOSC/32  
011 = FRC (clock supplied from a dedicated RC oscillator)  
100 = FOSC/4  
101 = FOSC/16  
110 = FOSC/64  
111 = FRC (clock supplied from a dedicated RC oscillator)
- bit 3            **Unimplemented:** Read as '0'
- bit 2            **ADNREF:** A/D Negative Voltage Reference Configuration bit  
0 = VREF- is connected to AVSS  
1 = VREF- is connected to external VREF-
- bit 1-0        **ADPREF<1:0>:** A/D Positive Voltage Reference Configuration bits  
00 = VREF+ is connected to AVDD  
01 = Reserved  
10 = VREF+ is connected to external VREF+  
11 = VREF+ is connected to internal fixed voltage reference

## REGISTER 11-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
bit 7							bit 0

### Legend:

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

- bit 7-0        **ADRES<9:2>:** ADC Result Register bits  
Upper 8 bits of 10-bit conversion result

# PIC16F193X/LF193X

## REGISTER 11-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES1	ADRES0	—	—	—	—	—	—
bit 7							bit 0

### Legend:

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

bit 7-6            **ADRES<1:0>**: ADC Result Register bits  
Lower 2 bits of 10-bit conversion result

bit 5-0            **Reserved**: Do not use.

## REGISTER 11-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES9	ADRES8
bit 7						bit 0	

### Legend:

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

bit 7-2            **Reserved**: Do not use.

bit 1-0            **ADRES<9:8>**: ADC Result Register bits  
Upper 2 bits of 10-bit conversion result

## REGISTER 11-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							bit 0

### Legend:

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

bit 7-0            **ADRES<7:0>**: ADC Result Register bits  
Lower 8 bits of 10-bit conversion result

# PIC16F193X/LF193X

## 11.3 A/D Acquisition Requirements

For the ADC 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 11-4. 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), refer to Figure 11-4. **The maximum recommended impedance for analog sources is 10 kΩ.** As the

source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (256 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

### EQUATION 11-1: ACQUISITION TIME EXAMPLE

*Assumptions: Temperature = 50°C and external impedance of 10kΩ 5.0V VDD*

$$\begin{aligned}TACQ &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= TAMP + TC + TCOFF \\ &= 2\mu s + TC + [(Temperature - 25^\circ C)(0.05\mu s/^\circ C)]\end{aligned}$$

*The value for TC can be approximated with the following equations:*

$$V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) = V_{CHOLD} \quad ;[1] \text{ } V_{CHOLD} \text{ charged to within } 1/2 \text{ lsb}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{TC}{RC}} \right) = V_{CHOLD} \quad ;[2] \text{ } V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{TC}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) \quad ;\text{combining [1] and [2]}$$

**Note:** Where n = number of bits of the ADC.

*Solving for TC:*

$$\begin{aligned}TC &= -CHOLD(RIC + RSS + RS) \ln(1/511) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.001957) \\ &= 1.12\mu s\end{aligned}$$

*Therefore:*

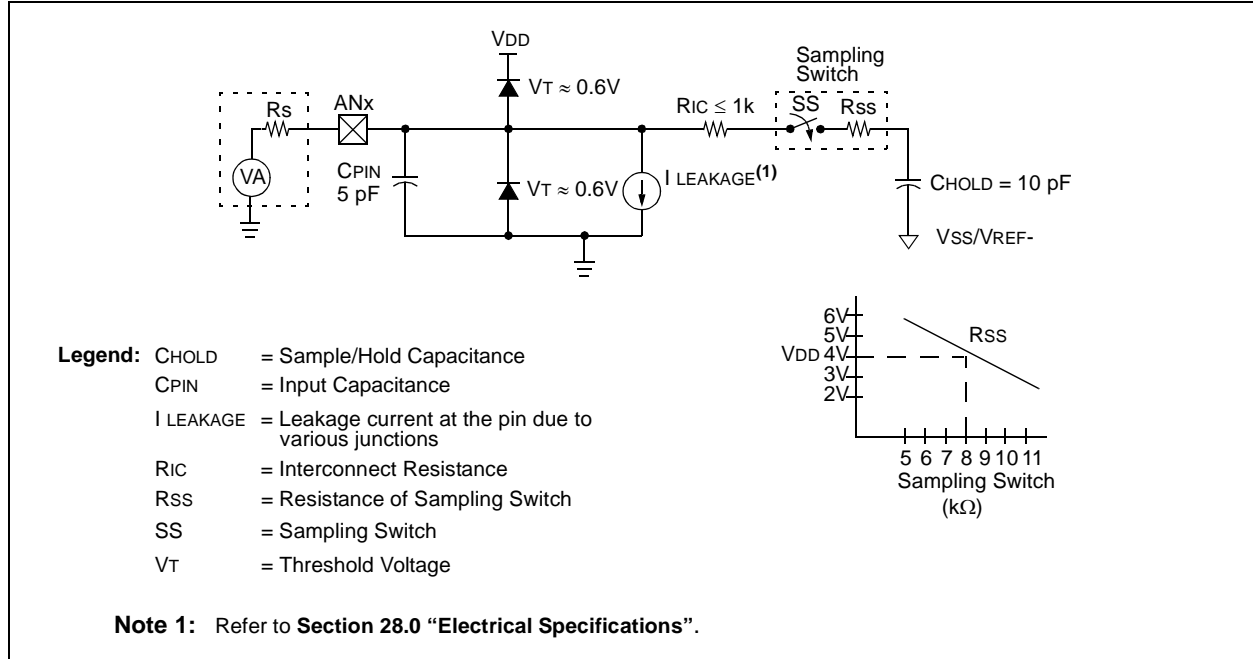
$$\begin{aligned}TACQ &= 2MS + 1.12MS + [(50^\circ C - 25^\circ C)(0.05MS/^\circ C)] \\ &= 4.42MS\end{aligned}$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

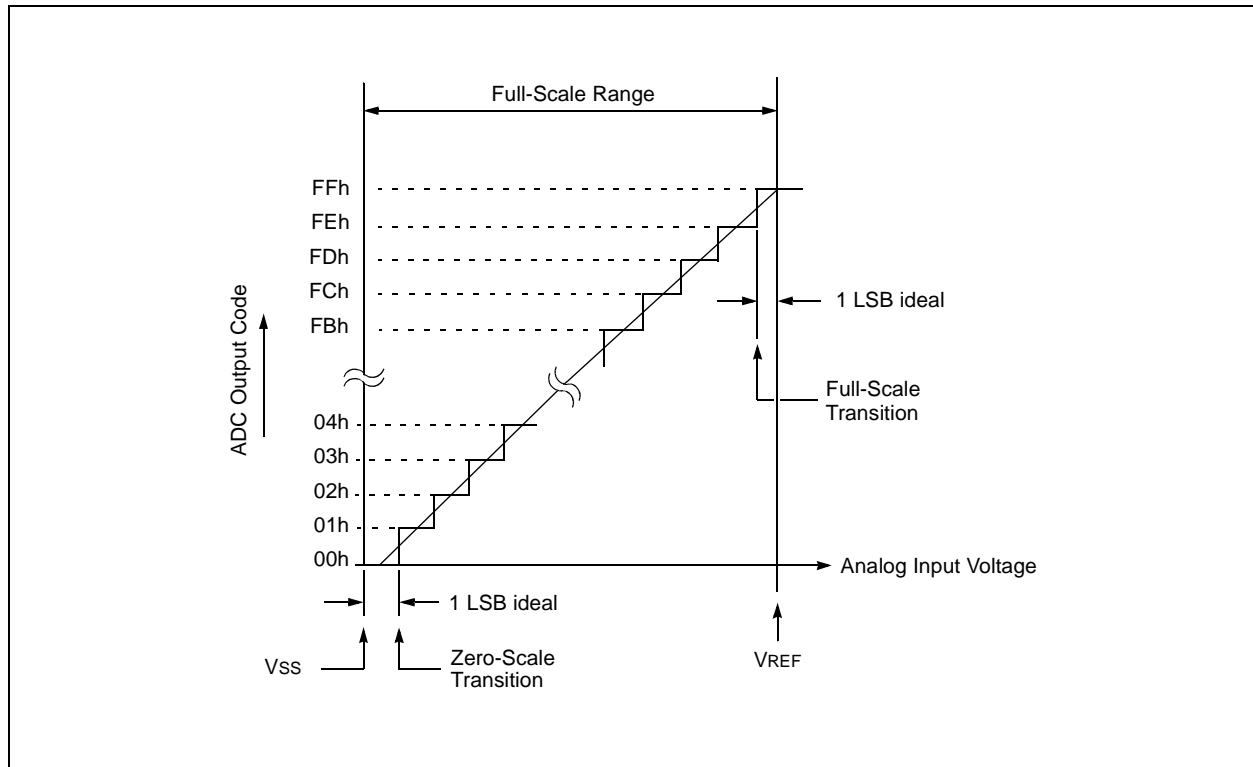
**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**FIGURE 11-4: ANALOG INPUT MODEL**



**FIGURE 11-5: ADC TRANSFER FUNCTION**



# PIC16F193X/LF193X

**TABLE 11-2: SUMMARY OF ASSOCIATED ADC REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	137
ADCON1	ADFM	ADCS2	ADCS1	ADCS0	—	ADNREF	ADPREF1	ADPREF0	138
ADRESH	A/D Result Register High								138
ADRESL	A/D Result Register Low								139
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	86
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91
ANSELE	—	—	—	—	—	ANSE2	ANSE1	ANSE0	101
CCP2CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	184
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISE	—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0	101
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	156
DACCON0	DACEN	DACLPS	DACOE	---	DACPSS1	DACPSS0	---	DACNSS	153
DACCON1	---	---	---	DACR4	DACR3	DACR2	DACR1	DACR0	153

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.



## 12.0 COMPARATOR MODULE

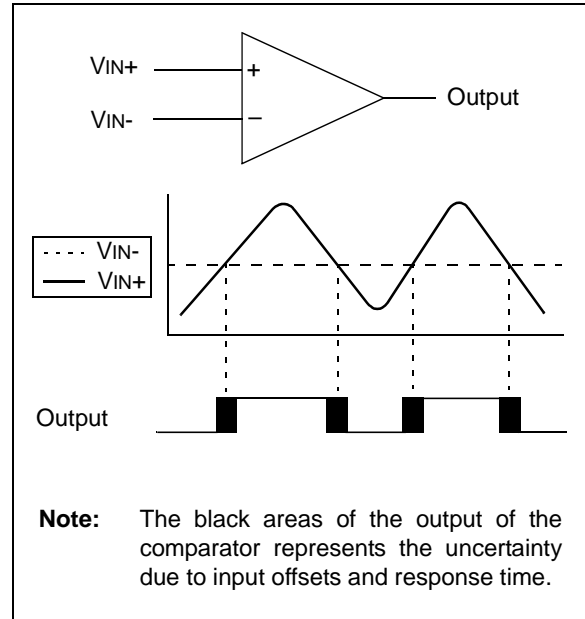
Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Independent comparator control
- Programmable input selection
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Programmable Speed/Power optimization
- PWM shutdown
- Programmable and fixed voltage reference

### 12.1 Comparator Overview

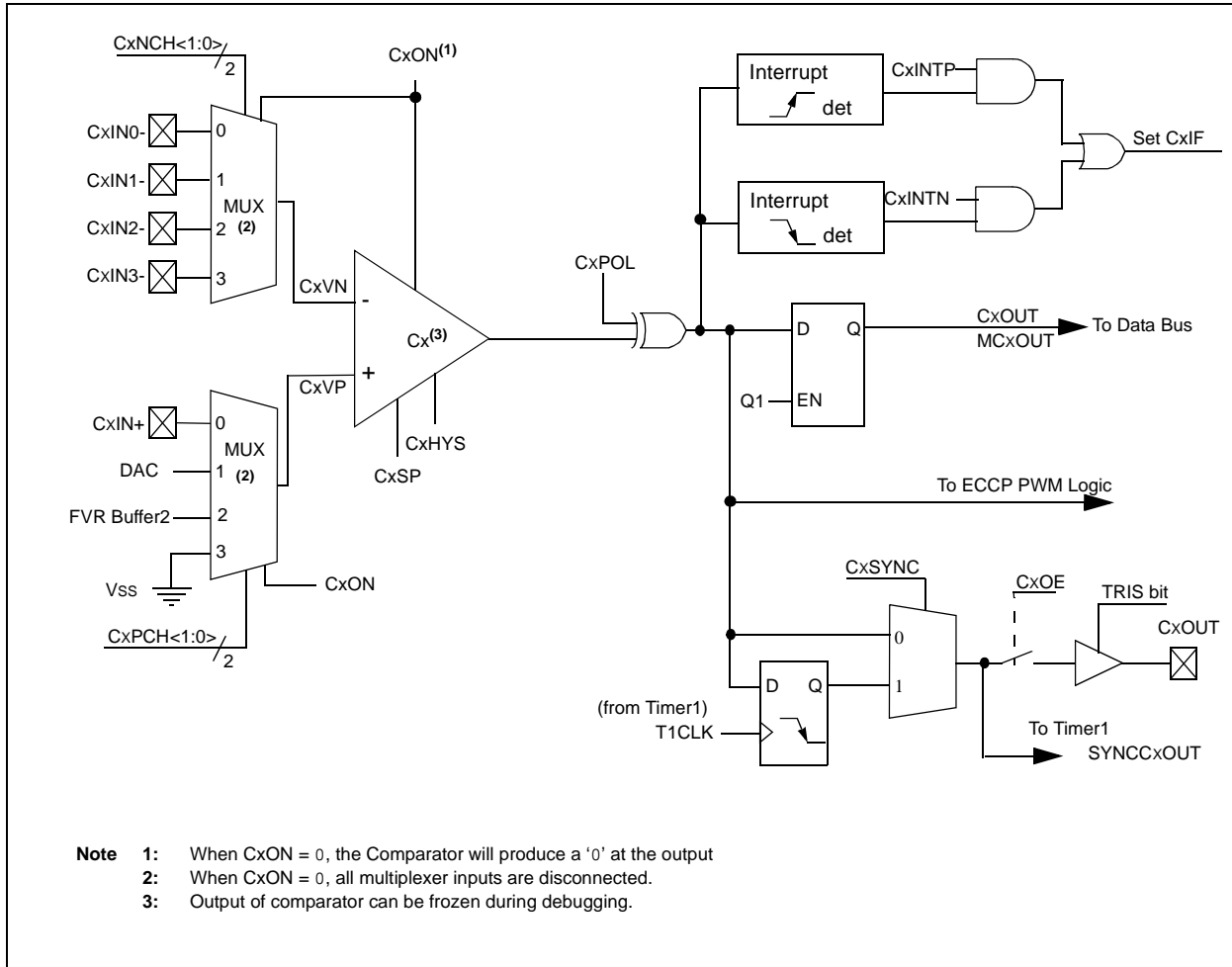
A single comparator is shown in Figure 12-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at  $V_{IN+}$  is less than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog voltage at  $V_{IN+}$  is greater than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital high level.

FIGURE 12-1: SINGLE COMPARATOR



# PIC16F193X/LF193X

**FIGURE 12-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM**



## 12.2 Comparator Control

Each comparator has 2 control registers: CMxCON0 and CMxCON1.

The CMxCON0 registers (see Register 12-1) contain Control and Status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 registers (see Register 12-2) contain Control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

### 12.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

### 12.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRIS bit must be cleared
- CxON bit of the CMxCON0 register must be set

**Note 1:** The CxOE bit overrides the PORT data latch. Setting the CxON has no impact on the port override.

**2:** The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

### 12.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 12-1 shows the output state versus input conditions, including polarity control.

**TABLE 12-1: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS**

Input Condition	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	0
$CxVN < CxVP$	1	1

### 12.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the normal speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

# PIC16F193X/LF193X

## 12.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation.

These hysteresis levels change as a function of the comparator's Speed/Power mode selection.

Table 12-2 shows the hysteresis levels.

**TABLE 12-2: HYSTERESIS LEVELS**

CxSP	CxHYS Enabled	CxHYS Disabled
0	$\pm 3\text{mV}$	$\ll \pm 1\text{mV}$
1	$\pm 20\text{mV}$	$\pm 3\text{mV}$

These levels are approximate.

See **Section 28.0 “Electrical Specifications”** for more information.

## 12.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 16.6 “Timer1 Gate”** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

### 12.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from either comparator, C1 or C2, can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 12-2) and the Timer1 Block Diagram (Figure 16-1) for more information.

## 12.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a Falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

**Note:** Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

## 12.6 Comparator Positive Input Selection

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC
- FVR (Fixed Voltage Reference)
- AVss (Analog Ground)

See **Section 14.0 “Fixed Voltage Reference”** for more information on the fixed voltage reference module.

See **Section 11.0 “Analog-to-Digital Converter (ADC) Module”** for more information on the CVDAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

## 12.7 Comparator Negative Input Selection

The CxNCH<1:0> bits of the CMxCON0 register direct one of four analog pins to the comparator inverting input.

**Note:** To use CxIN+ and CxINx- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

## 12.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section 28.0 “Electrical Specifications”** for more details.

## 12.9 Interaction with ECCP Logic

The C1 and C2 comparators can be used as general purpose comparators. Their outputs can be brought out to the C1OUT and C2OUT pins. However, when the ECCP Auto-Shutdown is active it can use one or both comparators. If auto-restart is also enabled the comparators can be configured as a closed loop analog feedback to the ECCP thereby, creating an analog controlled PWM.

## 12.10 Analog Input Connection Considerations

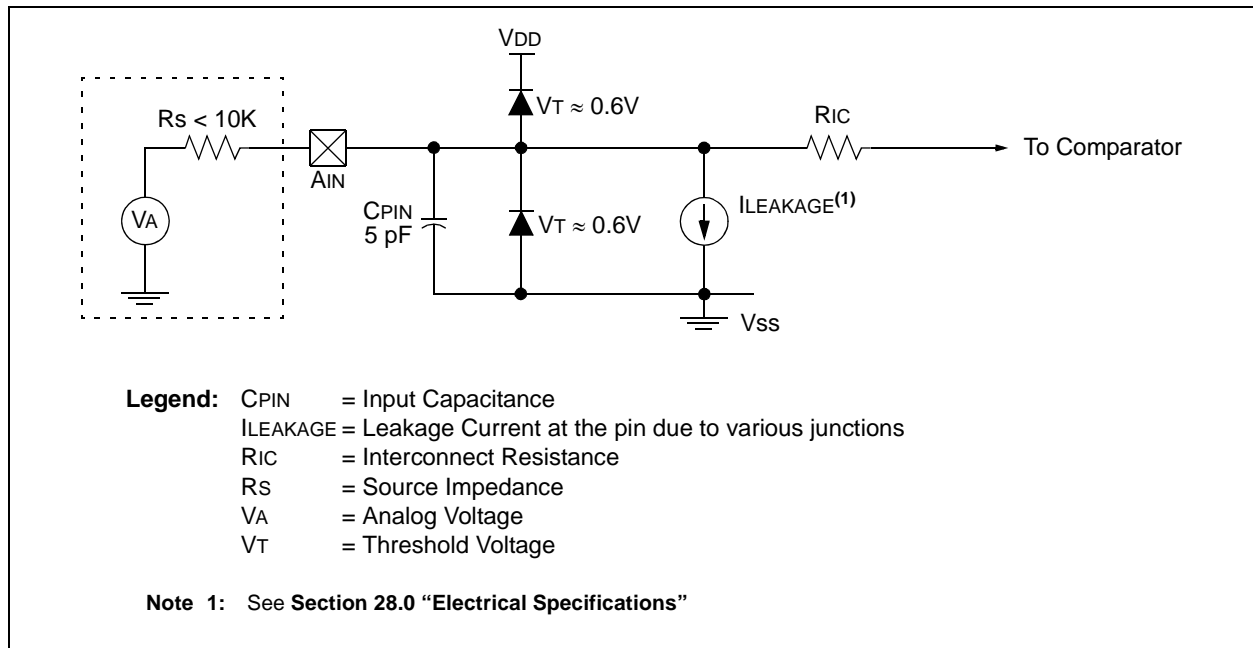
A simplified circuit for an analog input is shown in Figure 12-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range 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. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

**Note 1:** When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

**2:** Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 12-3: ANALOG INPUT MODEL**



# PIC16F193X/LF193X

## REGISTER 12-1: CMxCON0: COMPARATOR X CONTROL REGISTER 0

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit 0

### Legend:

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

- bit 7      **CxON:** Comparator Enable bit  
 1 = Comparator is enabled and consumes no active power  
 0 = Comparator is disabled
- bit 6      **CxOUT:** Comparator Output bit  
If CxPOL = 1 (inverted polarity):  
 1 = CxVP < CxVN  
 0 = CxVP > CxVN  
If CxPOL = 0 (non-inverted polarity):  
 1 = CxVP > CxVN  
 0 = CxVP < CxVN
- bit 5      **CxOE:** Comparator Output Enable bit  
 1 = CxOUT is present on the CxOUT pin. Requires that the associated TRIS bit be cleared to actually drive the pin. Not affected by CxON.  
 0 = CxOUT is internal only
- bit 4      **CxPOL:** Comparator Output Polarity Select bit  
 1 = Comparator output is inverted  
 0 = Comparator output is not inverted
- bit 3      **Unimplemented:** Read as '0'
- bit 2      **CxSP:** Comparator Speed/Power Select bit  
 1 = Comparator operates in normal power, higher speed mode  
 0 = Comparator operates in low-power, low-speed mode
- bit 1      **CxHYS:** Comparator Hysteresis Enable bit  
 1 = Comparator hysteresis enabled  
 0 = Comparator hysteresis disabled
- bit 0      **CxSYNC:** Comparator Output Synchronous Mode bit  
 1 = Comparator output to Timer1 and I/O pin is synchronous to changes on tmr1\_clk. Output updated on the falling edge of tmr1\_clk.  
 0 = Comparator output to Timer1 and I/O pin is asynchronous.  
 Refer to Figure 12-2.

## REGISTER 12-2: CMxCON1: COMPARATOR CX CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH1	CxPCH0	—	—	CxNCH1	CxNCH0
bit 7							bit 0

### Legend:

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

- bit 7      **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits  
 1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit  
 0 = No interrupt flag will be set on a positive going edge of the CxOUT bit
- bit 6      **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits  
 1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit  
 0 = No interrupt flag will be set on a negative going edge of the CxOUT bit
- bit 5-4    **CxPCH<1:0>:** Comparator Positive Input Channel Select bits  
 00 = CxVP connects to CxIN+ pin  
 01 = CxVP connects to CVDAC  
 10 = CxVP connects to FVR Voltage Reference  
 11 = CxVP connects to AVss
- bit 3-2    **Unimplemented:** Read as '0'
- bit 1-0    **CxNCH<1:0>:** Comparator Negative Input Channel Select bits  
 00 = CxVN connects to CxIN0- pin  
 01 = CxVN connects to CxIN1- pin  
 10 = CxVN connects to CxIN2- pin  
 11 = CxVN connects to CxIN3- pin

**Note 1:** Comparator output requires the following three conditions: C2OE = 1, C2ON = 1 and corresponding port TRIS bit = 0.

## REGISTER 12-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7							bit 0

### Legend:

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

- bit 7-2    **Unimplemented:** Read as '0'
- bit 1      **MC2OUT:** Mirror Copy of C2OUT bit
- bit 0      **MC1OUT:** Mirror Copy of C1OUT bit

# PIC16F193X/LF193X

**TABLE 12-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CM1CON0	C1ON	C1OUT	C1OE	C1POL	---	C1SP	C1HYS	C1SYNC	148
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	148
CM1CON1	C1NTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH1	C1NCH0	149
CM2CON1	C2NTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH1	C2NCH0	149
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	149
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	156
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS1	DACPSS0	—	DACNSS	153
DACCON1	—	—	—	DACR4	DACR3	DACR2	DACR1	DACR0	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	93
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	93
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	86
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.



## 13.0 DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with  $V_{DD}$ , with 32 selectable output levels. The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACOUT device pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACCON0 register.

### 13.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACCON1 register.

The DAC output voltage is determined by the following equations:

#### EQUATION 13-1:

$$V_{OUT} = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR\langle 4:0 \rangle}{2^{\wedge} 5} \right) + V_{SOURCE-}$$

$V_{SOURCE+} = V_{DD}, V_{REF+}$  or  $FVR1$

$V_{SOURCE-} = V_{SS}$  or  $V_{REF-}$

### 13.2 Output Clamped to Vss

The DAC output voltage can be set to  $V_{SS}$  with no power consumption by setting the DACEN bit of the DACCON0 register to '0'. This allows the comparator to detect a zero-crossing while not consuming additional current from the DAC.

### 13.3 Output Ratiometric to VDD

The DAC is  $V_{DD}$  derived and therefore, the DAC output changes with fluctuations in  $V_{DD}$ . The tested absolute accuracy of the DAC can be found in **Section 28.0 "Electrical Specifications"**.

### 13.4 Voltage Reference Output

The DAC can be output to the device DACOUT pin by setting the DACOE bit of the DACCON0 register to '1'. Selecting the reference voltage for output on the DACOUT pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACOUT pin when it has been configured for reference voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the voltage reference output for external connections to DACOUT. Figure 13-1 shows an example buffering technique.

### 13.5 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DACCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

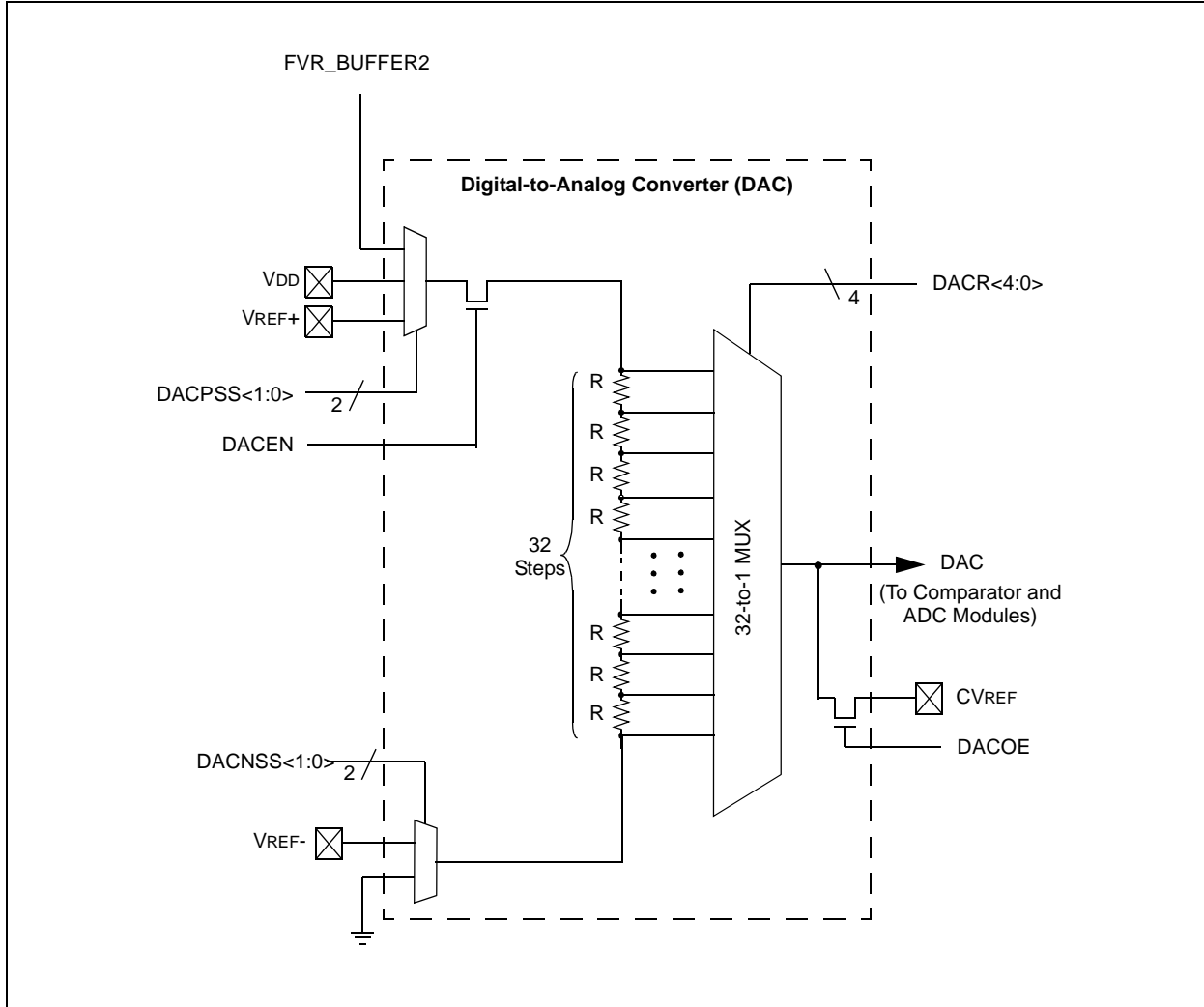
### 13.6 Effects of a Reset

A device Reset affects the following:

- Voltage reference is disabled
- Fixed voltage reference is disabled
- DAC is removed from the DACOUT pin
- The DACR<4:0> range select bits are cleared

# PIC16F193X/LF193X

FIGURE 13-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



# PIC16F193X/LF193X

## REGISTER 13-1: DACCON0: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
DACEN	DACLPS	DACOE	---	DACPSS1	DACPSS0	---	DACNSS
bit 7							bit 0

### Legend:

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

- bit 7      **DACEN:** DAC Enable bit  
0 = DAC is disabled  
1 = DAC is enabled
- bit 6      **DACLPS:** DAC Low-Power Voltage State Select bit  
0 = V<sub>DAC</sub> = DAC Negative reference source selected  
1 = V<sub>DAC</sub> = DAC Positive reference source selected
- bit 5      **DACOE:** DAC Voltage Output Enable bit  
1 = DAC voltage level is also an output on the DACOUT pin  
0 = DAC voltage level is disconnected from the DACOUT pin
- bit 4      **Unimplemented:** Read as '0'
- bit 3-2    **DACPSS<1:0>:** DAC Positive Source Select bits  
00 = V<sub>DD</sub>  
01 = V<sub>REF+</sub>  
10 = FVR1 output  
11 = Reserved, do not use
- bit 1      **Unimplemented:** Read as '0'
- bit 0      **DACNSS:** DAC Negative Source Select bits  
0 = V<sub>SS</sub>  
1 = V<sub>REF-</sub>

## REGISTER 13-2: DACCON1: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
---	---	---	DACR4	DACR3	DACR2	DACR1	DACR0
bit 7							bit 0

### Legend:

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

- bit 7-5    **Unimplemented:** Read as '0'
- bit 4-0    **DACR<4:0>:** DAC Voltage Output Select bits  

$$V_{OUT} = ((V_{SOURCE+}) - (V_{SOURCE-})) * (DACR<4:0> / (2^5)) + V_{SOURCE-}$$

**Note 1:** The output select bits are always right justified to ensure that any number of bits can be used without affecting the register layout

# PIC16F193X/LF193X

---

---

**TABLE 13-1: REGISTERS ASSOCIATED WITH THE DIGITAL-TO-ANALOG CONVERTER**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	156
DACCON0	DACEN	DACLPS	DACOE	---	DACPSS1	DACPSS0	---	DACNSS	153
DACCON1	---	---	---	DACR4	DACR3	DACR2	DACR1	DACR0	153

**Legend:** Shaded cells are not used with the DAC.

## 14.0 FIXED VOLTAGE REFERENCE

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Programmable voltage reference
- LCD bias generator

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

### 14.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC and comparator modules is routed through two independent programmable gain amplifiers. Each amplifier can be configured to amplify the reference voltage by 1x, 2x or 4x.

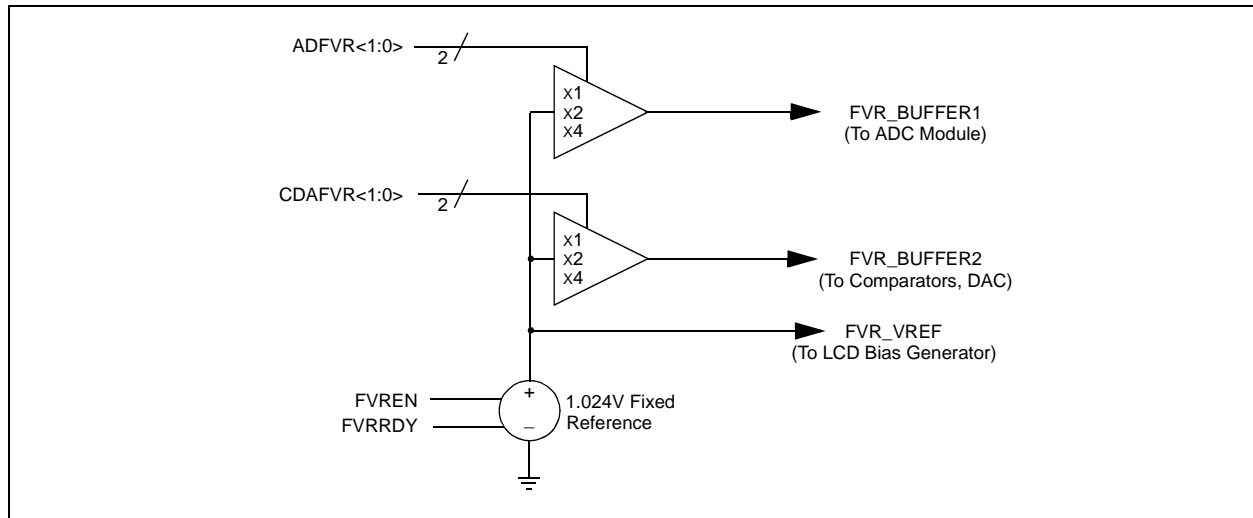
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 11.0 “Analog-to-Digital Converter (ADC) Module”** for additional information on selecting the appropriate input channel.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the comparator module. Reference **Section 12.0 “Comparator Module”** for additional information on selecting the appropriate input channel.

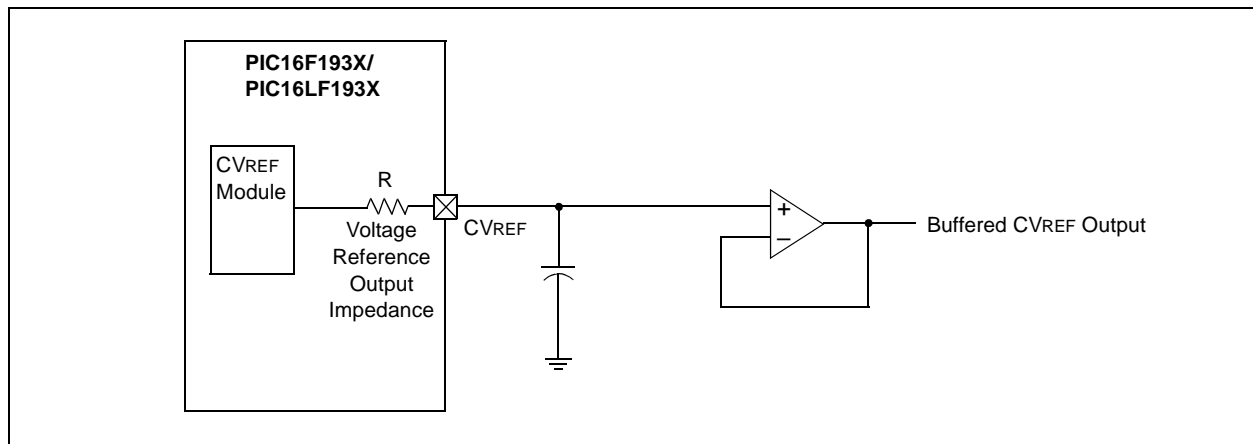
### 14.2 FVR Stabilization Period

When the fixed voltage reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 28.0 “Electrical Specifications”** for the minimum delay requirement.

**FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM**



**FIGURE 14-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE**



# PIC16F193X/LF193X

**REGISTER 14-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER**

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY <sup>(1)</sup>	—	—	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0
bit 7							bit 0

**Legend:**

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

- bit 7      **FVREN:** Fixed Voltage Reference Enable bit  
 0 = Fixed Voltage Reference is disabled  
 1 = Fixed Voltage Reference is enabled
- bit 6      **FVRRDY:** Fixed Voltage Reference Ready Flag bit<sup>(1)</sup>  
 0 = Fixed Voltage Reference output is not active or stable  
 1 = Fixed Voltage Reference output is ready for use
- bit 5-4    **Reserved:** Read as '0'. Maintain these bits clear.
- bit 3-2    **CDAFVR<1:0>:** Comparator and D/A Converter Fixed Voltage Reference Selection bit  
 00 = Comparator and D/A Converter Fixed Voltage Reference Peripheral output is off.  
 01 = Comparator and D/A Converter Fixed Voltage Reference Peripheral output is 1x (1.024V)  
 10 = Comparator and D/A Converter Fixed Voltage Reference Peripheral output is 2x (2.048V)<sup>(2)</sup>  
 11 = Comparator and D/A Converter Fixed Voltage Reference Peripheral output is 4x (4.096V)<sup>(2)</sup>
- bit 1-0    **ADFVR<1:0>:** A/D Converter Fixed Voltage Reference Selection bit  
 00 = A/D Converter Fixed Voltage Reference Peripheral output is off.  
 01 = A/D Converter Fixed Voltage Reference Peripheral output is 1x (1.024V)  
 10 = A/D Converter Fixed Voltage Reference Peripheral output is 2x (2.048V)<sup>(2)</sup>  
 11 = A/D Converter Fixed Voltage Reference Peripheral output is 4x (4.096V)<sup>(2)</sup>

- Note 1:** FVRRDY is always '1' on regulated parts (PIC16F193X).  
**Note 2:** Fixed Voltage Reference output cannot exceed VDD.

**TABLE 14-1: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	156

**Legend:** Shaded cells are not used with the voltage reference.

## 15.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 15-1 is a block diagram of the Timer0 module.

### 15.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

#### 15.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the TMR0CS bit of the OPTION register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

**Note:** The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

#### 15.1.2 8-BIT COUNTER MODE

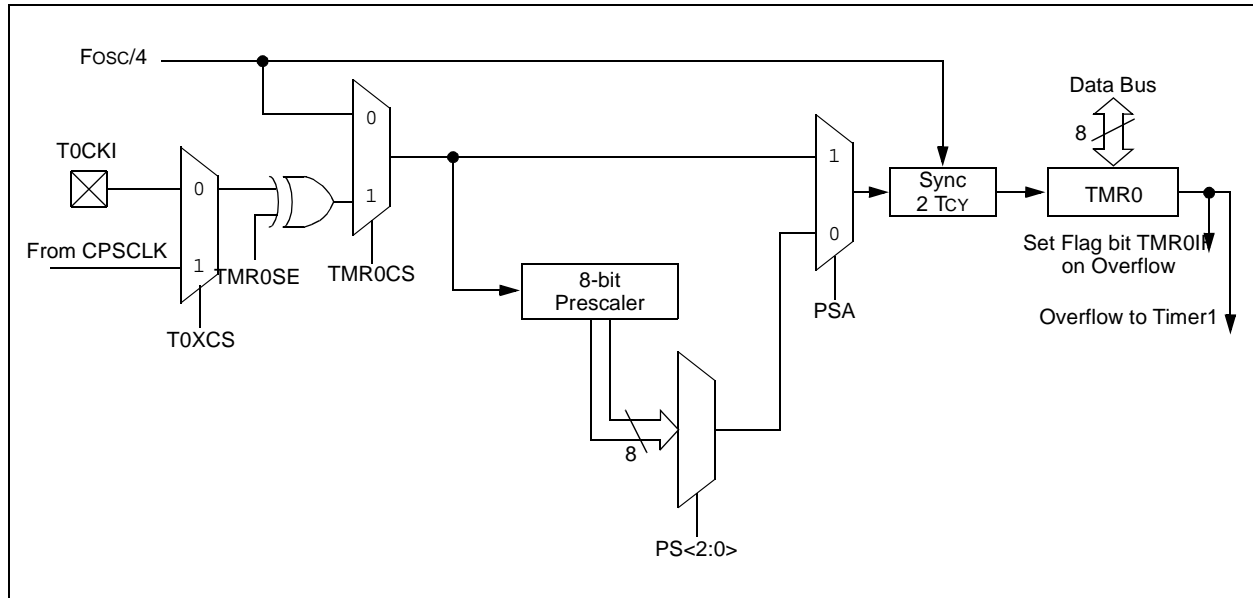
In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin or the Capacitive Sensing Oscillator (CPSCCLK) signal.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION register to '1' and resetting the T0XCS bit in the CPSCON0 register to '0'.

8-Bit Counter Mode using the Capacitive Sensing Oscillator (CPSCCLK) signal is selected by setting the TMR0CS bit in the OPTION register to '1' and setting the T0XCS bit in the CPSCON0 register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMR0SE bit in the OPTION register.

**FIGURE 15-1: BLOCK DIAGRAM OF THE TIMER0**



# PIC16F193X/LF193X

---

## 15.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION register.

**Note:** The Watchdog Timer (WDT) uses its own independent prescaler.

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

## 15.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

**Note:** The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

## 15.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 28.0 “Electrical Specifications”**.

## 15.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.



# PIC16F193X/LF193X

**REGISTER 15-1: OPTION\_REG: OPTION REGISTER**

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

**Legend:**

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

- bit 7            **WPUEN:** Weak Pull-up Enable bit  
1 = All weak pull-ups are disabled (except MCLR, if it is enabled)  
0 = Weak pull-ups are enabled by individual WPUx latch values
- bit 6            **INTEDG:** Interrupt Edge Select bit  
1 = Interrupt on rising edge of RB0/INT pin  
0 = Interrupt on falling edge of RB0/INT pin
- bit 5            **TMR0CS:** Timer0 Clock Source Select bit  
1 = Transition on RA4/T0CKI pin  
0 = Internal instruction cycle clock (FOSC/4)
- bit 4            **TMR0SE:** Timer0 Source Edge Select bit  
1 = Increment on high-to-low transition on RA4/T0CKI pin  
0 = Increment on low-to-high transition on RA4/T0CKI 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	Timer0 Rate
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

**TABLE 15-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	180
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	73
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	51
TMR0	Timer0 Module Register								157*
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

\* Page provides register information.

# PIC16F193X/LF193X

---

NOTES:

## 16.0 TIMER1 MODULE WITH GATE CONTROL

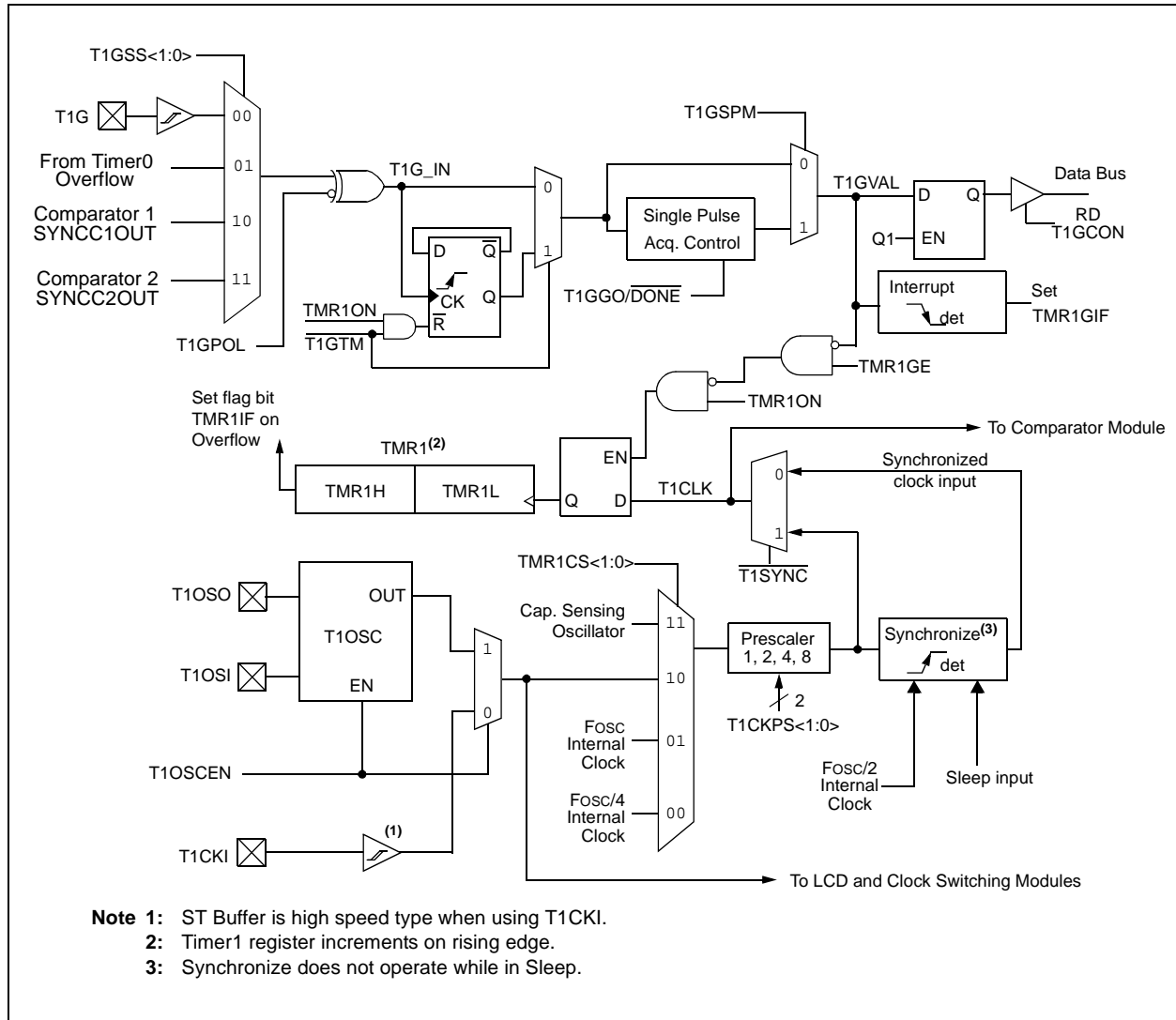
The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Dedicated LP oscillator circuit
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity

- Gate Toggle Mode
- Gate Single-pulse Mode
- Gate Value Status
- Gate Event Interrupt

Figure 16-1 is a block diagram of the Timer1 module.

**FIGURE 16-1: TIMER1 BLOCK DIAGRAM**



# PIC16F193X/LF193X

## 16.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 16-1 displays the Timer1 enable selections.

**TABLE 16-1: TIMER1 ENABLE SELECTIONS**

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

## 16.2 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 16-2 displays the clock source selections.

### 16.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

### 16.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

**TABLE 16-2: CLOCK SOURCE SELECTIONS**

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	1	x	System Clock (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	1	x	Capacitive Sensing Oscillator
1	0	0	External Clocking on T1CKI Pin
1	0	1	Osc.Circuit On T1OSI/T1OSO Pins

## 16.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register 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.

## 16.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

**Note:** The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.

## 16.5 Timer1 Operation in Asynchronous Counter Mode

If control bit  $\overline{T1SYNC}$  of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then 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 16.5.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”).

**Note:** When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

### 16.5.1 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 TMR1H:TMR1L register pair.

## 16.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 Gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 Gate can also be driven by multiple selectable sources.

### 16.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 16-3 for timing details.

**TABLE 16-3: TIMER1 GATE ENABLE SELECTIONS**

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

### 16.6.2 TIMER1 GATE SOURCE SELECTION

The Timer1 Gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 16-4: TIMER1 GATE SOURCES**

T1GSS	Timer1 Gate Source
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
10	Comparator 1 Output SYNCC1OUT (optionally synchronized out)
11	Comparator 2 Output SYNCC2OUT (optionally synchronized out)

# PIC16F193X/LF193X

---

## 16.6.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 Gate Control. It can be used to supply an external source to the Timer1 Gate circuitry.

## 16.6.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 Gate circuitry.

## 16.6.2.3 Comparator C1 Gate Operation

The output resulting from a Comparator 1 operation can be selected as a source for Timer1 Gate Control. The Comparator 1 output (SYNCC1OUT) can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 12.4.1 “Comparator Output Synchronization”**.

## 16.6.2.4 Comparator C2 Gate Operation

The output resulting from a Comparator 2 operation can be selected as a source for Timer1 Gate Control. The Comparator 2 output (SYNCC2OUT) can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 12.4.1 “Comparator Output Synchronization”**.

## 16.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 Gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 16-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

<b>Note:</b> Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.
---

## 16.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/DONE bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/DONE bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/DONE bit. See Figure 16-5 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 Gate source to be measured. See Figure 16-6 for timing details.

## 16.6.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GCON bit is valid even when the Timer1 Gate is not enabled (TMR1GE bit is cleared).

## 16.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 Gate is not enabled (TMR1GE bit is cleared).

## 16.7 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 of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

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

**Note:** The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

## 16.8 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:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- T1SYNC bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1OSCEN bit of the T1CON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

Timer1 oscillator will continue to operate in Sleep regardless of the T1SYNC bit setting.

## 16.9 ECCP/CCP Capture/Compare Time Base

The CCP modules uses the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

For more information, see **Section 19.0 “Capture/Compare/PWM Modules (ECCP1, ECCP2, ECCP3, CCP4, CCP5)”**.

## 16.10 ECCP/CCP Special Event Trigger

When any of the CCP's are configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

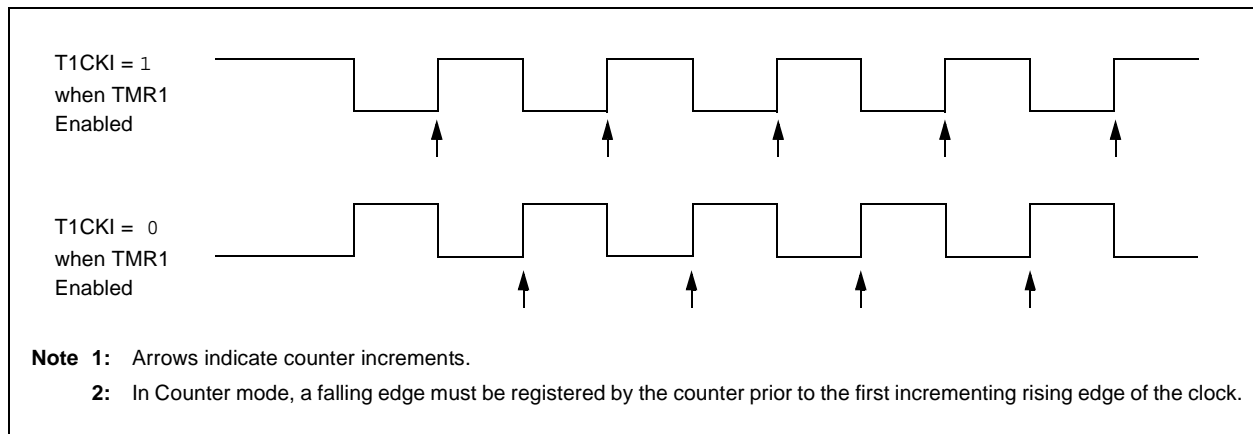
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC/4 to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the CCP, the write will take precedence.

For more information, see **Section 11.2.5 “Special Event Trigger”**.

**FIGURE 16-2: TIMER1 INCREMENTING EDGE**



# PIC16F193X/LF193X

FIGURE 16-3: TIMER1 GATE ENABLE MODE

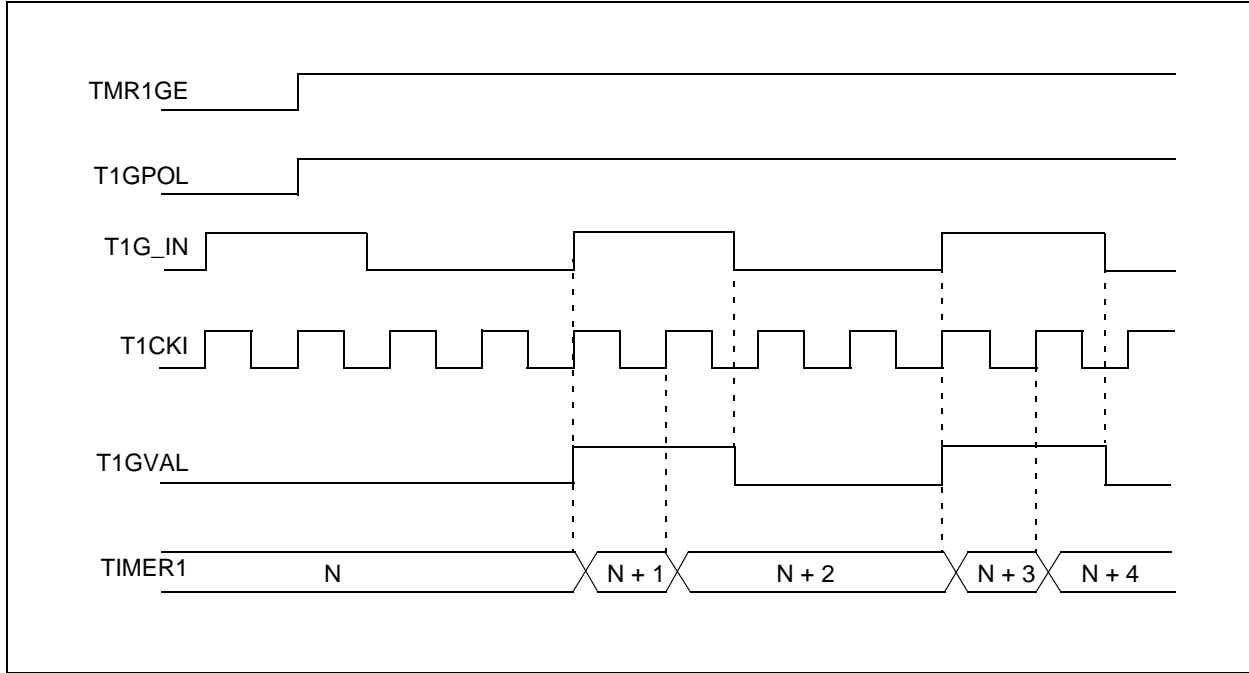
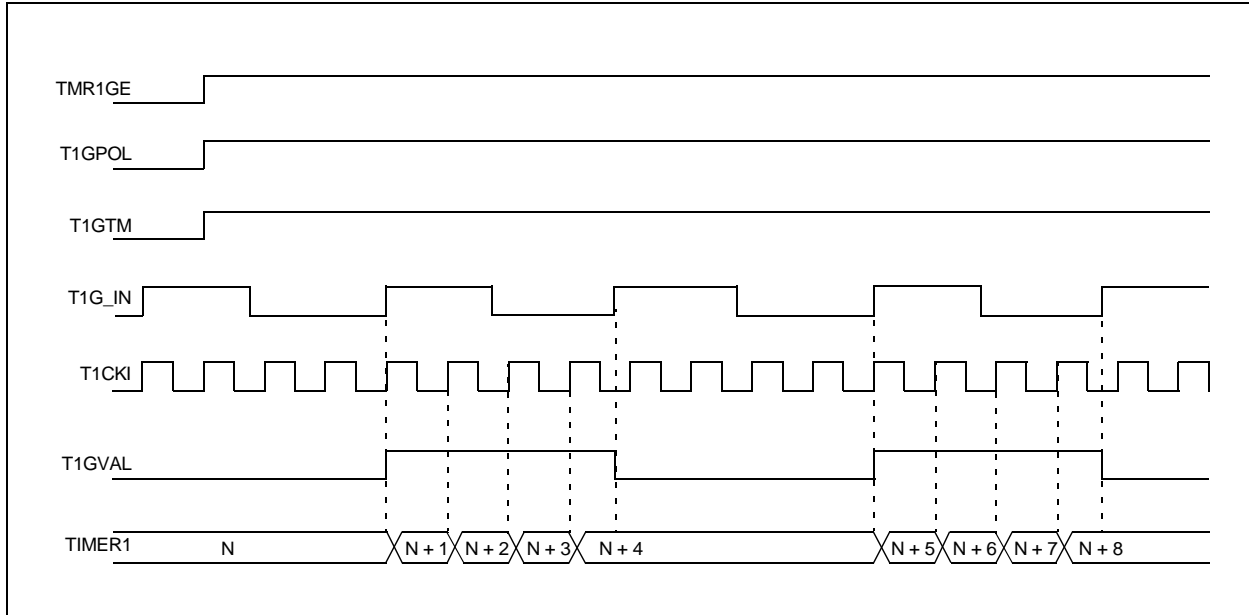
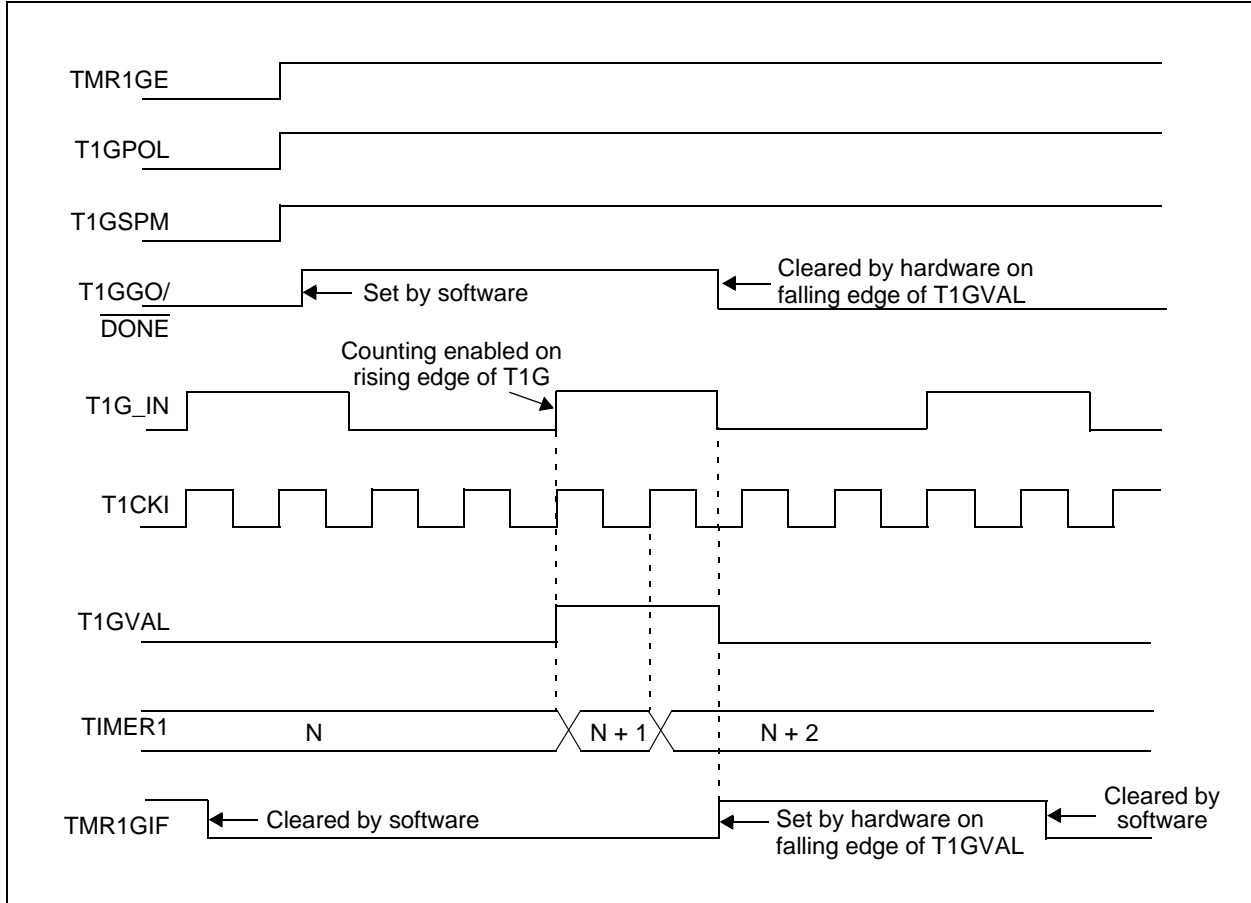


FIGURE 16-4: TIMER1 GATE TOGGLE MODE



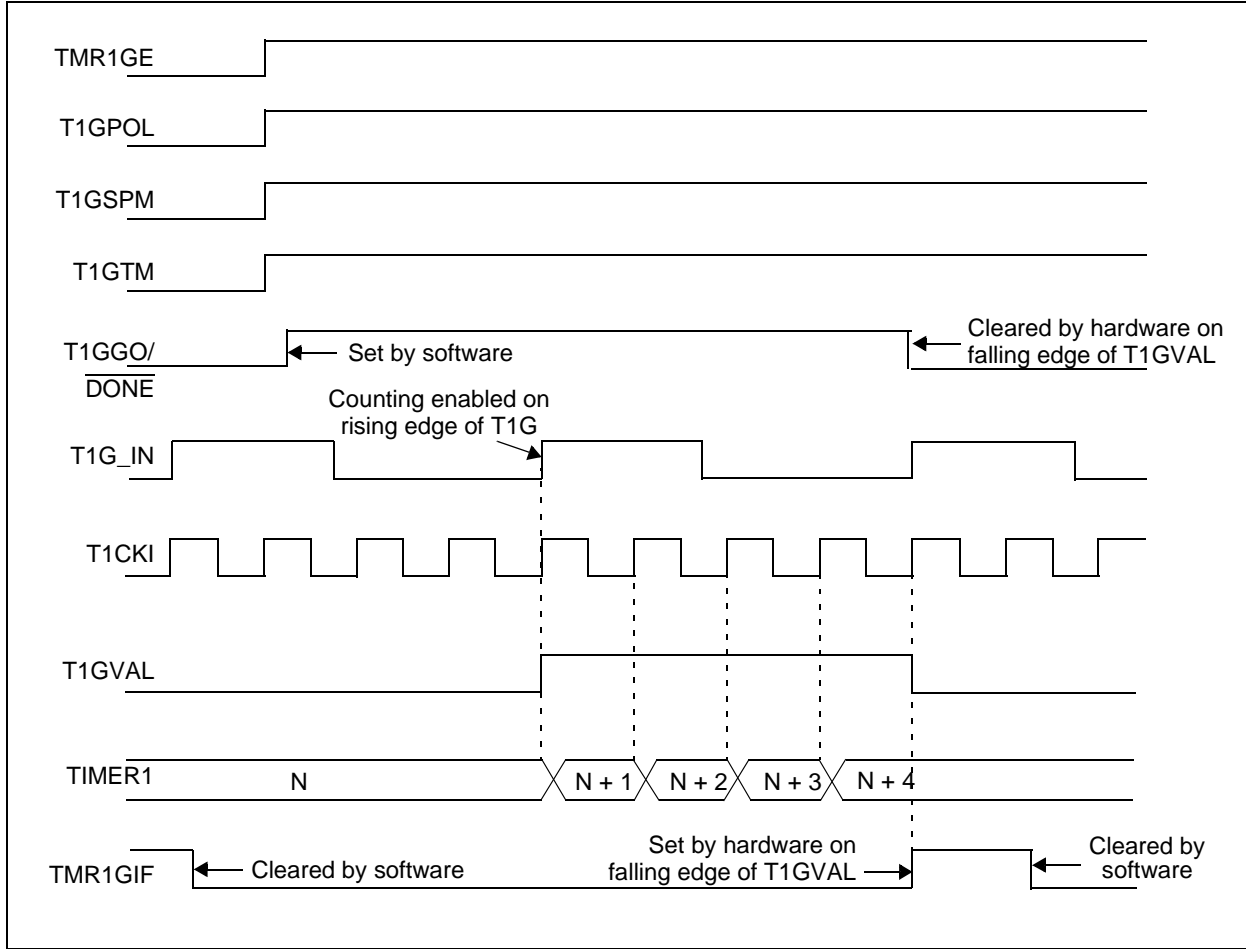


**FIGURE 16-5: TIMER1 GATE SINGLE-PULSE MODE**



# PIC16F193X/LF193X

FIGURE 16-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



## 16.11 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 16-1, is used to control Timer1 and select the various features of the Timer1 module.

**REGISTER 16-1: T1CON: TIMER1 CONTROL REGISTER**

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON
bit 7							bit 0

**Legend:**

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

- bit 7-6      **TMR1CS<1:0>**: Timer1 Clock Source Select bits  
 11 = Timer1 clock source is Capacitive Sensing Oscillator (CAPOSC)  
 10 = Timer1 clock source is pin or oscillator:  
     If **T1OSCEN** = 0:  
         External clock from T1CKI pin (on the rising edge)  
     If **T1OSCEN** = 1:  
         Crystal oscillator on T1OSI/T1OSO pins  
 01 = Timer1 clock source is system clock (Fosc)  
 00 = Timer1 clock source is instruction clock (Fosc/4)
- 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  
 1 = Dedicated Timer1 oscillator circuit enabled  
 0 = Dedicated Timer1 oscillator circuit disabled
- bit 2        **T1SYNC**: Timer1 External Clock Input Synchronization Control bit  
TMR1CS<1:0> = 1X  
 1 = Do not synchronize external clock input  
 0 = Synchronize external clock input with system clock (Fosc)
- TMR1CS<1:0> = 0X  
 This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1X.
- bit 1        **Unimplemented**: Read as '0'
- bit 0        **TMR1ON**: Timer1 On bit  
 1 = Enables Timer1  
 0 = Stops Timer1  
 Clears Timer1 Gate flip-flop

# PIC16F193X/LF193X

## 16.12 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), shown in Register 16-2, is used to control Timer1 Gate.

**REGISTER 16-2: T1GCON: TIMER1 GATE CONTROL REGISTER**

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS1	T1GSS0
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

bit 7 **TMR1GE:** Timer1 Gate Enable bit

If TMR1ON = 0:

This bit is ignored

If TMR1ON = 1:

1 = Timer1 counting is controlled by the Timer1 gate function

0 = Timer1 counts regardless of Timer1 gate function

bit 6 **T1GPOL:** Timer1 Gate Polarity bit

1 = Timer1 gate is active-high (Timer1 counts when gate is high)

0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **T1GTM:** Timer1 Gate Toggle Mode bit

1 = Timer1 Gate Toggle mode is enabled

0 = Timer1 Gate Toggle mode is disabled and toggle flip flop is cleared

Timer1 gate flip-flop toggles on every rising edge.

bit 4 **T1GSPM:** Timer1 Gate Single-Pulse Mode bit

1 = Timer1 gate Single-Pulse mode is enabled and is controlling Timer1 gate

0 = Timer1 gate Single-Pulse mode is disabled

bit 3 **T1GGO/DONE:** Timer1 Gate Single-Pulse Acquisition Status bit

1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge

0 = Timer1 gate single-pulse acquisition has completed or has not been started

This bit is automatically cleared when T1GSPM is cleared.

bit 2 **T1GVAL:** Timer1 Gate Current State bit

Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L.

Unaffected by Timer1 Gate Enable (TMR1GE).

bit 1-0 **T1GSS<1:0>:** Timer1 Gate Source Select bits

00 = Timer1 Gate pin

01 = Timer0 overflow output

10 = Comparator 1 optionally synchronized output (SYNCC1OUT)

11 = Comparator 2 optionally synchronized output (SYNCC2OUT)

# PIC16F193X/LF193X

**TABLE 16-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	184
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	184
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	90
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								165*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								165*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	169
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS1	T1GSS0	170

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

\* Page provides register information.

# PIC16F193X/LF193X

---

NOTES:

## 17.0 TIMER2/4/6 MODULES

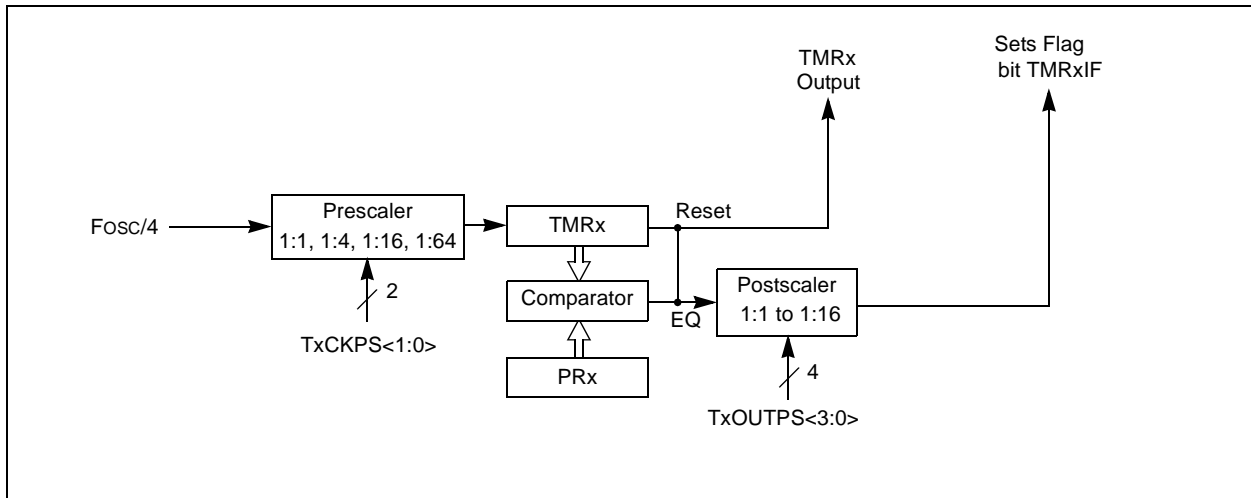
There are three identical Timer2-type modules available. To maintain pre-existing naming conventions, the Timers are called Timer2, Timer4 and Timer6 (also Timer2/4/6).

The Timer2/4/6 modules incorporate the following features:

- 8-bit Timer and Period registers (TMRx and PRx, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMRx match with PRx, respectively
- Optional use as the shift clock for the MSSPx modules (Timer2 only)

See Figure 17-1 for a block diagram of Timer2/4/6.

**FIGURE 17-1: TIMER2/4/6 BLOCK DIAGRAM**



# PIC16F193X/LF193X

---

## 17.1 Timer2/4/6 Operation

The clock input to the Timer2/4/6 modules is the system instruction clock (Fosc/4).

TMRx increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, TxCKPS<1:0> of the TxCON register. The value of TMRx is compared to that of the Period register, PRx, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMRx to 00h on the next cycle and drives the output counter/postscaler (see **Section 17.2 “Timer2/4/6 Interrupt”**).

The TMRx and PRx registers are both directly readable and writable. The TMRx register is cleared on any device Reset, whereas the PRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMRx register
- a write to the TxCON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- MCLR Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

<b>Note:</b> TMRx is not cleared when TxCON is written.
---

## 17.2 Timer2/4/6 Interrupt

Timer2/4/6 can also generate an optional device interrupt. The Timer2/4/6 output signal (TMRx-to-PRx match) provides the input for the 4-bit counter/postscaler. This counter generates the TMRx match interrupt flag which is latched in TMRxIF of the PIRx register. The interrupt is enabled by setting the TMRx Match Interrupt Enable bit, TMRxIE of the PIEx register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, TxOUTPS<3:0>, of the TxCON register.

## 17.3 Timer2/4/6 Output

The unscaled output of TMRx is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSPx modules operating in SPI mode. Additional information is provided in **Section 17.0 “SSP Module Overview”**

## 17.4 Timer2/4/6 Operation During Sleep

The Timerx timers cannot be operated while the processor is in Sleep mode. The contents of the TMRx and PRx registers will remain unchanged while the processor is in Sleep mode.



# PIC16F193X/LF193X

## REGISTER 17-1: TXCON: TIMER2-TYPE TIMER CONTROL REGISTER

U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMRxON	TxCKPS1	TxCKPS0
bit 7							bit 0

### Legend:

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

bit 7	<b>Unimplemented:</b> Read as '0'
bit 6-3	<b>TOUTPS&lt;3:0&gt;:</b> Timer Output Postscaler Select bits 0000 = 1:1 Postscaler 0001 = 1:2 Postscaler 0010 = 1:3 Postscaler 0011 = 1:4 Postscaler 0100 = 1:5 Postscaler 0101 = 1:6 Postscaler 0110 = 1:7 Postscaler 0111 = 1:8 Postscaler 1000 = 1:9 Postscaler 1001 = 1:10 Postscaler 1010 = 1:11 Postscaler 1011 = 1:12 Postscaler 1100 = 1:13 Postscaler 1101 = 1:14 Postscaler 1110 = 1:15 Postscaler 1111 = 1:16 Postscaler
bit 2	<b>TMRxON:</b> Timerx On bit 1 = Timerx is on 0 = Timerx is off
bit 1-0	<b>TxCKPS&lt;1:0&gt;:</b> Timer2-type Clock Prescale Select bits 00 = Prescaler is 1 01 = Prescaler is 4 10 = Prescaler is 16 11 = Prescaler is 64

# PIC16F193X/LF193X

**TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2/4/6**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	184
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	184
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	76
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	79
PR2	Timer2 Module Period Register								173*
TMR2	Holding Register for the 8-bit TMR2 Register								173*
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	175

**Legend:** x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

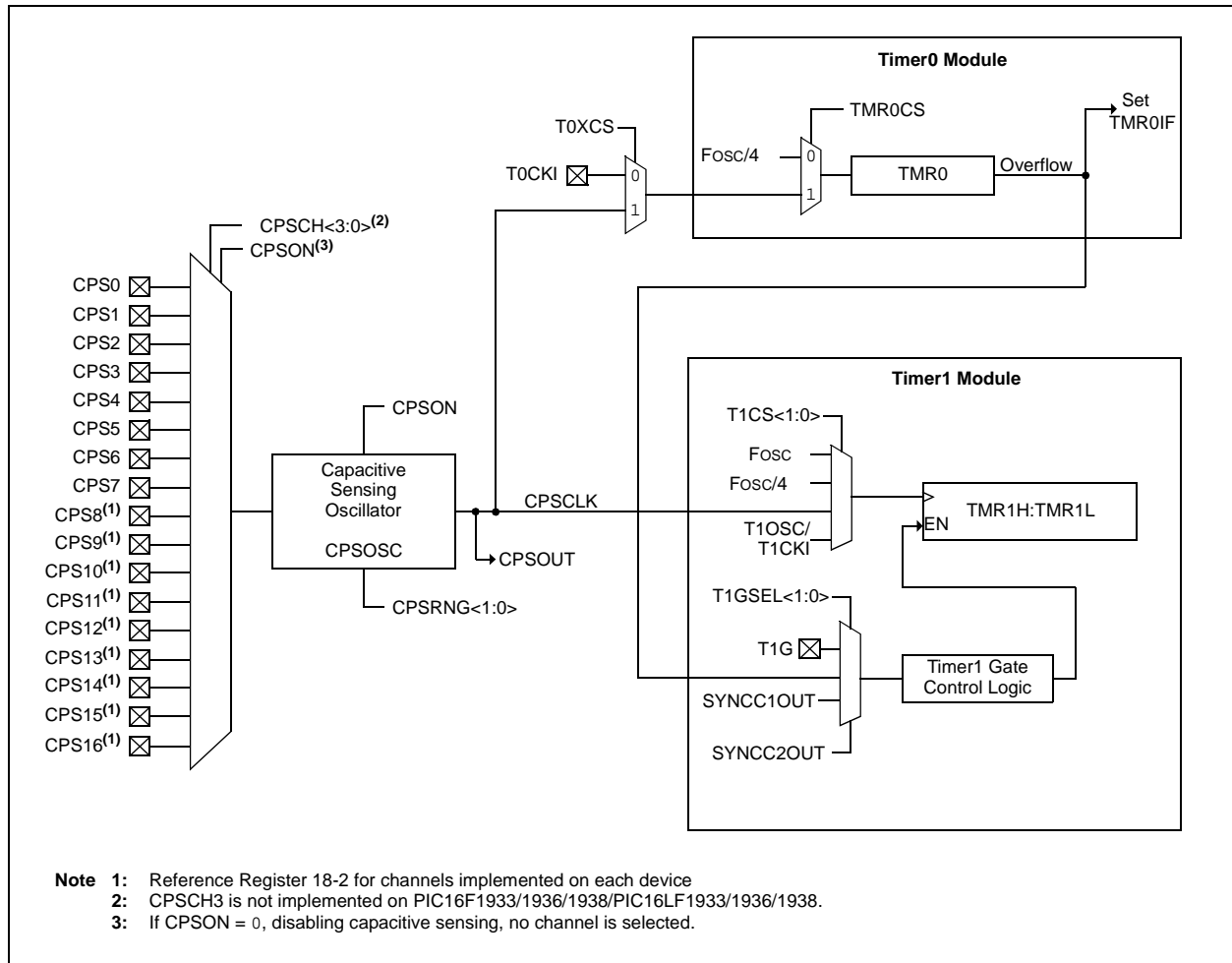
\* Page provides register information.

## 18.0 CAPACITIVE SENSING MODULE

The capacitive sensing module allows for an interaction with an end user without a mechanical interface. In a typical application, the capacitive sensing module is attached to a pad on a Printed Circuit Board (PCB), which is electrically isolated from the end user. When the end user places their finger over the PCB pad, a capacitive load is added, causing a frequency shift in the capacitive sensing module. The capacitive sensing module requires software and at least one timer resource to determine the change in frequency. Key features of this module include:

- Analog MUX for monitoring multiple inputs
- Capacitive sensing oscillator
- Multiple timer resources
- Software control
- Operation during Sleep

**FIGURE 18-1: CAPACITIVE SENSING BLOCK DIAGRAM**



# PIC16F193X/LF193X

## 18.1 Analog MUX

The capacitive sensing module can monitor up to 16 inputs. The capacitive sensing inputs are defined as CPS<15:0>. To determine if a frequency change has occurred the user must:

- Select the appropriate CPS pin by setting the CPSCH<3:0> bits of the CPSCON1 register
- Set the corresponding ANSEL bit
- Set the corresponding TRIS bit
- Run the software algorithm

Selection of the CPSx pin while the module is enabled will cause the capacitive sensing oscillator to be on the CPSx pin. Failure to set the corresponding ANSEL and TRIS bits can cause the capacitive sensing oscillator to stop, leading to false frequency readings.

## 18.2 Capacitive Sensing Oscillator

The capacitive sensing oscillator consists of a constant current source and a constant current sink, to produce a triangle waveform. The CPSOUT bit of the CPSCON0 register shows the status of the capacitive sensing oscillator, whether it is a sinking or sourcing current. The oscillator is designed to drive a capacitive load (single PCB pad) and at the same time, be a clock source to either Timer0 or Timer1. The oscillator has three different current settings as defined by CPRNG<1:0> of the CPSCON0 register. The different current settings for the oscillator serve two purposes:

- Maximize the number of counts in a timer for a fixed time base
- Maximize the count differential in the timer during a change in frequency

## 18.3 Timer resources

To measure the change in frequency of the capacitive sensing oscillator, a fixed time base is required. For the period of the fixed time base, the capacitive sensing oscillator is used to clock either Timer0 or Timer1. The frequency of the capacitive sensing oscillator is equal to the number of counts in the timer divided by the period of the fixed time base.

## 18.4 Fixed Time Base

To measure the frequency of the capacitive sensing oscillator, a fixed time base is required. Any timer resource or software loop can be used to establish the fixed time base. It is up to the end user to determine the method in which the fixed time base is generated.

**Note:** The fixed time base can not be generated by the timer resource that the capacitive sensing oscillator is clocking.

### 18.4.1 TIMER0

To select Timer0 as the timer resource for the capacitive sensing module:

- Set the T0XCS bit of the CPSCON0 register
- Clear the TMR0CS bit of the OPTION register

When Timer0 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer0. Refer to **Section 15.0 “Timer0 Module”** for additional information.

### 18.4.2 TIMER1

To select Timer1 as the timer resource for the capacitive sensing module, set the TMR1CS<1:0> of the T1CON register to ‘11’. When Timer1 is chosen as the timer resource, the capacitive sensing oscillator will be the clock source for Timer1. Because the Timer1 module has a gate control, developing a time base for the frequency measurement can be simplified by using the Timer0 overflow flag.

It is recommend that the Timer0 overflow flag, in conjunction with the Toggle mode of the Timer1 Gate, be used to develop the fixed time base required by the software portion of the capacitive sensing module. Refer to **Section 16.12 “Timer1 Gate Control Register”** for additional information.

**TABLE 18-1: TIMER1 ENABLE FUNCTION**

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	On
1	1	Count Enabled by input

## 18.5 Software Control

The software portion of the capacitive sensing module is required to determine the change in frequency of the capacitive sensing oscillator. This is accomplished by the following:

- Setting a fixed time base to acquire counts on Timer0 or Timer1
- Establishing the nominal frequency for the capacitive sensing oscillator
- Establishing the reduced frequency for the capacitive sensing oscillator due to an additional capacitive load
- Set the frequency threshold

### 18.5.1 NOMINAL FREQUENCY (NO CAPACITIVE LOAD)

To determine the nominal frequency of the capacitive sensing oscillator:

- Remove any extra capacitive load on the selected CPSx pin
- At the start of the fixed time base, clear the timer resource
- At the end of the fixed time base save the value in the timer resource

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator for the given time base. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed time base.

### 18.5.2 REDUCED FREQUENCY (ADDITIONAL CAPACITIVE LOAD)

The extra capacitive load will cause the frequency of the capacitive sensing oscillator to decrease. To determine the reduced frequency of the capacitive sensing oscillator:

- Add a typical capacitive load on the selected CPSx pin
- Use the same fixed time base as the nominal frequency measurement
- At the start of the fixed time base, clear the timer resource
- At the end of the fixed time base save the value in the timer resource

The value of the timer resource is the number of oscillations of the capacitive sensing oscillator with an additional capacitive load. The frequency of the capacitive sensing oscillator is equal to the number of counts on in the timer divided by the period of the fixed time base. This frequency should be less than the value obtained during the nominal frequency measurement.

### 18.5.3 FREQUENCY THRESHOLD

The frequency threshold should be placed midway between the value of nominal frequency and the reduced frequency of the capacitive sensing oscillator. Refer to Application Note AN1103, "*Software Handling for Capacitive Sensing*" (DS01103) for more detailed information on the software required for capacitive sensing module.

**Note:** For more information on general capacitive sensing refer to Application Notes:

- AN1101, "*Introduction to Capacitive Sensing*" (DS01101)
- AN1102, "*Layout and Physical Design Guidelines for Capacitive Sensing*" (DS01102)

## 18.6 Operation during Sleep

The capacitive sensing oscillator will continue to run as long as the module is enabled, independent of the part being in Sleep. In order for the software to determine if a frequency change has occurred, the part must be awake. However, the part does not have to be awake when the timer resource is acquiring counts.

**Note:** Timer0 does not operate when in Sleep, and therefore cannot be used for capacitive sense measurements in Sleep.

# PIC16F193X/LF193X

## REGISTER 18-1: CPSCON0: CAPACITIVE SENSING CONTROL REGISTER 0

R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R-0/0	R/W-0/0
CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS
bit 7							bit 0

### Legend:

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

- bit 7      **CPSON:** Capacitive Sensing Module Enable bit  
 1 = Capacitive sensing module is operating  
 0 = Capacitive sensing module is shut off and consumes no operating current
- bit 6-4    **Unimplemented:** Read as '0'
- bit 3-2    **CPSRNG<1:0>:** Capacitive Sensing Oscillator Range bits  
 00 = Oscillator is off  
 01 = Oscillator is in low range. Charge/discharge current is nominally 0.1  $\mu$ A.  
 10 = Oscillator is in medium range. Charge/discharge current is nominally 1.2  $\mu$ A.  
 11 = Oscillator is in high range. Charge/discharge current is nominally 18  $\mu$ A.
- bit 1      **CPSOUT:** Capacitive Sensing Oscillator Status bit  
 1 = Oscillator is sourcing current (Current flowing out the pin)  
 0 = Oscillator is sinking current (Current flowing into the pin)
- bit 0      **T0XCS:** Timer0 External Clock Source Select bit  
If TMR0CS = 1  
 The T0XCS bit controls which clock external to the core/Timer0 module supplies Timer0:  
 1 = Timer0 clock source is the capacitive sensing oscillator  
 0 = Timer0 clock source is the T0CKI pin  
If TMR0CS = 0  
 Timer0 clock source is controlled by the core/Timer0 module and is FOSC/4

# PIC16F193X/LF193X

**REGISTER 18-2: CPSCON1: CAPACITIVE SENSING CONTROL REGISTER 1**

U-0	U-0	U-0	R/W-0/0 <sup>(1, 2)</sup>	R/W-0/0 <sup>(1)</sup>	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CPSCH4	CPSCH3	CPSCH2	CPSCH1	CPSCH0
bit 7							bit 0

**Legend:**

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

bit 7-4                      **Unimplemented:** Read as '0'  
bit 3-0                      **CPSCCH<3:0>:** Capacitive Sensing Channel Select bits  
If CPSON = 0:  
                                    These bits are ignored. No channel is selected.  
If CPSON = 1:  
                                    0000 = channel 0, (CPS0)  
                                    0001 = channel 1, (CPS1)  
                                    0010 = channel 2, (CPS2)  
                                    0011 = channel 3, (CPS3)  
                                    0100 = channel 4, (CPS4)  
                                    0101 = channel 5, (CPS5)  
                                    0110 = channel 6, (CPS6)  
                                    0111 = channel 7, (CPS7)  
                                    1000 = channel 8, (CPS8<sup>(1)</sup>)  
                                    1001 = channel 9, (CPS9<sup>(1)</sup>)  
                                    1010 = channel 10, (CPS10<sup>(1)</sup>)  
                                    1011 = channel 11, (CPS11<sup>(1)</sup>)  
                                    1100 = channel 12, (CPS12<sup>(1)</sup>)  
                                    1101 = channel 13, (CPS13<sup>(1)</sup>)  
                                    1110 = channel 14, (CPS14<sup>(1)</sup>)  
                                    1111 = channel 15, (CPS15<sup>(1)</sup>)

**Note 1:** These channels are not implemented on the PIC16F1933/1936/1938/PIC16LF1933/1936/1938.  
**Note 2:** This bit is not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938, read as '0'

**TABLE 18-2: SUMMARY OF REGISTERS ASSOCIATED WITH CAPACITIVE SENSING**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	86
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	91
ANSELD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	97
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	73
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	51
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	169
TxCON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMRXON	TXCKPS1	TXCKPS0	175
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	97

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the capacitive sensing module.

# PIC16F193X/LF193X

---

NOTES:



## 19.0 CAPTURE/COMPARE/PWM MODULES (ECCP1, ECCP2, ECCP3, CCP4, CCP5)

This device contains three Enhanced Capture/Compare/PWM (ECCP1, ECCP2, ECCP3) and two standard Capture/Compare/PWM module (CCP4 and CCP5). The CCP4 and CCP5 modules are identical in operation. The ECCP1, ECCP2 and ECCP3 modules may also be referred to as CCP1, CCP2, CCP3, as required.

# PIC16F193X/LF193X

## 19.1 Capture/Compare/PWM

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate a Pulse-Width Modulated signal of varying frequency and duty cycle.

Table 19-1 shows the timer resources required by the CCP module.

**TABLE 19-1: REQUIRED TIMER RESOURCES**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2 or 4 or 6

### REGISTER 19-1: CCPXCON: CCPX CONTROL REGISTER

R/W-00	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxM1 <sup>(1)</sup>	PxM0 <sup>(1)</sup>	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

**Legend:**

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

bit 7-6      **PxM<1:0>**: Enhanced PWM Output Configuration bits<sup>(1)</sup>  
If CCPxM<3:2> = 00, 01, 10:  
xx = PxA assigned as Capture/Compare input; PxB, PxC, PxD assigned as port pins  
If CCPxM<3:2> = 11:  
00 = Single output; PxA modulated; PxB, PxC, PxD assigned as port pins  
01 = Full-Bridge output forward; P1D modulated; P1A active; P1B, P1C inactive  
10 = Half-Bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins  
11 = Full-Bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4      **DCxB<1:0>**: PWM Duty Cycle 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>**: ECCPx Mode Select bits  
0000 = Capture/Compare/PWM off (resets ECCPx module)  
0001 = Reserved  
0010 = Compare mode: toggle output on match  
0011 = Capture mode  
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: initialize ECCPx pin low; set output on compare match (set CCPxIF)  
1001 = Compare mode: initialize ECCPx pin high; clear output on compare match (set CCPxIF)  
1010 = Compare mode: generate software interrupt only; ECCPx pin reverts to I/O state  
1011 = Compare mode: trigger special event (ECCPx resets TMR1 or TMR3, sets CCPxIF bit, ECCP2 trigger also starts A/D conversion if A/D module is enabled)<sup>(1)</sup>

CCP<5:4> only:  
11xx = PWM mode

ECCP<3:1> only:  
1100 = PWM mode: PxA, PxC active-high; PxB, PxD active-high  
1101 = PWM mode: PxA, PxC active-high; PxB, PxD active-low  
1110 = PWM mode: PxA, PxC active-low; PxB, PxD active-high  
1111 = PWM mode: PxA, PxC active-low; PxB, PxD active-low

**Note 1:** These bits are not implemented on CCP<5:4>.

## 19.2 CCP Clock Selection

The PIC16F193X/LF193X allows each individual CCP module to select the timer source that controls the CCP module. Each module has an independent selection.

As the PIC16F193X/LF193X has only one 16-bit timer (Timer1), the Capture and Compare modes of the CCP modules always uses Timer1. As there are three 8-bit

timers with auto-reload (Timer2, Timer4 and Timer6), PWM mode on the CCP modules can use any of these timers.

The following registers are used to select which timer is used:

- CCP Timers Control Register 0 (CCPTMRS0)
- CCP Timers Control Register 1 (CCPTMRS1)

### REGISTER 19-2: CCPTMRS0: CCP TIMERS CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
C4TSEL1	C4TSEL0	C3TSEL1	C3TSEL0	C2TSEL1	C2TSEL0	C1TSEL1	C1TSEL0
bit 7							bit 0

#### Legend:

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

- bit 7-6      **C4TSEL<1:0>**: CCP4 Timer Selection  
 00 = CCP4 is based off Timer 2 in PWM Mode  
 01 = CCP4 is based off Timer 4 in PWM Mode  
 10 = CCP4 is based off Timer 6 in PWM Mode  
 11 = Reserved
- bit 5-4      **C3TSEL<1:0>**: CCP3 Timer Selection  
 00 = CCP3 is based off Timer 2 in PWM Mode  
 01 = CCP3 is based off Timer 4 in PWM Mode  
 10 = CCP3 is based off Timer 6 in PWM Mode  
 11 = Reserved
- bit 3-2      **C2TSEL<1:0>**: CCP2 Timer Selection  
 00 = CCP2 is based off Timer 2 in PWM Mode  
 01 = CCP2 is based off Timer 4 in PWM Mode  
 10 = CCP2 is based off Timer 6 in PWM Mode  
 11 = Reserved
- bit 1-0      **C1TSEL<1:0>**: CCP1 Timer Selection  
 00 = CCP1 is based off Timer 2 in PWM Mode  
 01 = CCP1 is based off Timer 4 in PWM Mode  
 10 = CCP1 is based off Timer 6 in PWM Mode  
 11 = Reserved

# PIC16F193X/LF193X

## REGISTER 19-3: CCPTMRS1: CCP TIMERS CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	C5TSEL1	C5TSEL0
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = bit is unchanged

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

bit 7-2

**Unimplemented:** Read as '0'

bit 1-0

**C5TSEL<1:0>:** CCP5 Timer Selection

00 = CCP5 is based off Timer 2 in PWM Mode

01 = CCP5 is based off Timer 4 in PWM Mode

10 = CCP5 is based off Timer 6 in PWM Mode

11 = Reserved

## 19.3 Capture Mode

In Capture mode, the CCPRxH, CCPRxL register pair captures the 16-bit value of the TMR1 register when an event occurs on pin CCPx. An event is defined as one of the following and is configured by the CCPxM<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value (see Figure 19-1).

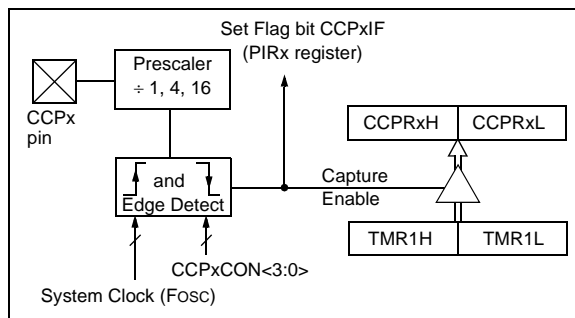
### 19.3.1 CCPX PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Also, the CCPx pin function can be moved to alternative pins using the APFCON register. Refer to **Section 6.1 “Alternate Pin Function”** for more details.

**Note:** If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

**FIGURE 19-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 19.3.2 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.

### 19.3.3 SOFTWARE INTERRUPT MODE

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

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

### 19.3.4 CCP PRESCALER

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

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler (see Example 19-1).

### EXAMPLE 19-1: CHANGING BETWEEN CAPTURE PRESCALERS

```

BANKSEL CCP1CON    ;Set Bank bits to point
                   ;to CCP1CON
CLRF    CCP1CON    ;Turn CCP module off
MOVLW  NEW_CAPT_PS ;Load the W reg with
                   ;the new prescaler
MOVWF  CCP1CON    ;move value and CCP ON
MOVWF  CCP1CON    ;Load CCP1CON with this
                   ;value
    
```

### 19.3.5 CAPTURE DURING SLEEP

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

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

If Timer1 is clocked by an external clock source, then Capture mode will operate as defined in **Section 19.1 “Capture/Compare/PWM”**.

# PIC16F193X/LF193X

**TABLE 19-2: REGISTERS ASSOCIATED WITH CAPTURE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCON	PxM1 <sup>(1)</sup>	PxM0 <sup>(1)</sup>	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
CCPRxL	Capture/Compare/PWM Register x Low Byte (LSB)								187
CCPRxH	Capture/Compare/PWM Register x High Byte (MSB)								187
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	148
CM1CON1	C1INTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH1	C1NCH0	149
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	148
CM2CON1	C2INTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH1	C2NCH0	149
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOCF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	79
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	169
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	170
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								165
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								165
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	97
TRISE	—	—	—	—	TRISE3	TRISE2 <sup>(2)</sup>	TRISE1 <sup>(2)</sup>	TRISE0 <sup>(2)</sup>	101

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture and Compare.

**Note 1:** Applies to ECCP modules only.

## 19.4 Compare Mode

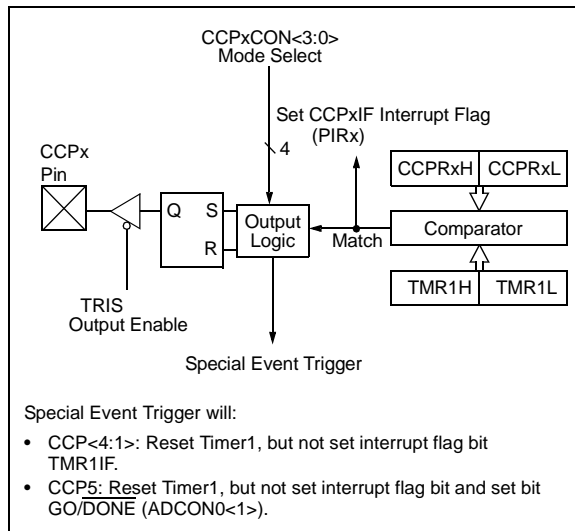
In Compare mode, the 16-bit CCPRx register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCPx module may:

- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

All Compare modes can generate an interrupt.

**FIGURE 19-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 19.4.1 CCPX PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Also, the CCPx pin function can be moved to alternative pins using the APFCON register. Refer to **Section 6.1 “Alternate Pin Function”** for more details.

**Note:** Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

### 19.4.2 TIMER1 MODE SELECTION

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

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

### 19.4.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the CCPx module does not assert control of the CCPx pin (see the CCP1CON register).

### 19.4.4 SPECIAL EVENT TRIGGER

When Special Event Trigger mode is chosen (CCPxM<3:0> = 1011), the CCPx module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled (CCP5 only)

The CCPx module does not assert control of the CCPx pin in this mode (see the CCPxCON register).

The Special Event Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPRxH, CCPRxL register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. The Special Event Trigger output starts an A/D conversion (if the A/D module is enabled). This feature is only available on CCP5). This allows the CCPRxH, CCPRxL register pair to effectively provide a 16-bit programmable period register for Timer1.

**Note 1:** The Special Event Trigger from the CCP module does not set interrupt flag bit TMR1IF of the PIR1 register.

**2:** Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring.

### 19.4.5 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (Fosc) for proper operation. Since Fosc is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

# PIC16F193X/LF193X

**TABLE 19-3: REGISTERS ASSOCIATED WITH COMPARE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCON	PxM1 <sup>(1)</sup>	PxM0 <sup>(1)</sup>	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
CCPRxL	Capture/Compare/PWM Register x Low Byte (LSB)								187
CCPRxH	Capture/Compare/PWM Register x High Byte (MSB)								187
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	148
CM1CON1	C1INTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH1	C1NCH0	149
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	148
CM2CON1	C2INTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH1	C2NCH0	149
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOCF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	79
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	169
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	170
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								165
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								165
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	97
TRISE	—	—	—	—	TRISE3	TRISE2 <sup>(2)</sup>	TRISE1 <sup>(2)</sup>	TRISE0 <sup>(2)</sup>	101

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture and Compare.

**Note 1:** Applies to ECCP modules only.

**2:** These bits are not implemented on PIC16F1933/1936/1938/PIC16LF1933/1936/1938 devices, read as '0'.



## 19.5 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCPx pin. The duty cycle, period and resolution are determined by the following registers:

- PRx
- TxCON
- CCPRxL
- CCPxCON

The ECCP modules have the following additional registers:

- ECCPxAS
- PSTRxCON
- PWMxCON

In Pulse-Width Modulation (PWM) mode, the CCPx module produces up to a 10-bit resolution PWM output on the CCPx pin. Since the CCPx pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCPx pin output driver.

**Note:** Clearing the CCPxCON register will relinquish CCPx control of the CCPx pin.

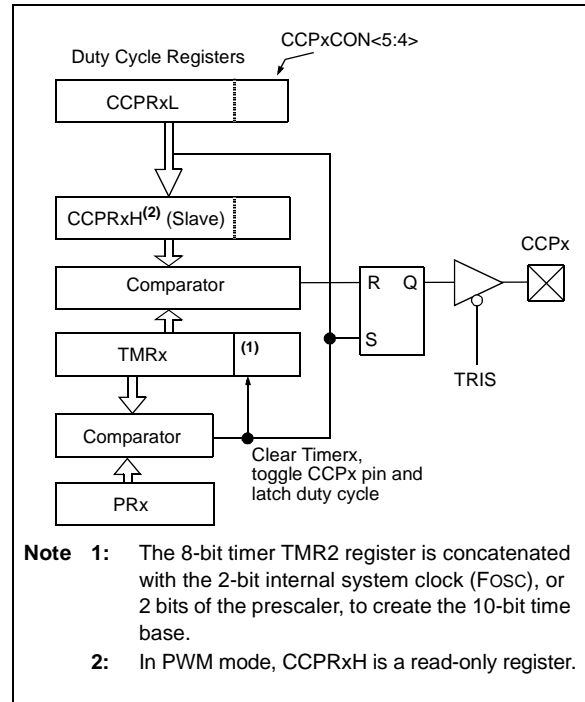
The CCPx module in PWM mode can have the PWM based off of either Timer2, Timer4 or Timer6. This is controlled by the CCPTMRS0 and CCPTMRS1 registers. Reference **Section 19.2 “CCP Clock Selection”** for more information.

Figure 19-3 shows a simplified block diagram of PWM operation.

Figure 19-4 shows a typical waveform of the PWM signal.

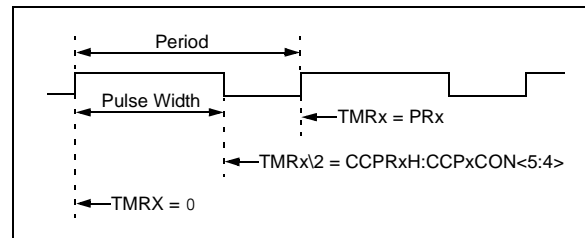
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 19.5.7 “Setup for PWM Operation”**.

**FIGURE 19-3: SIMPLIFIED PWM BLOCK DIAGRAM**



The PWM output (Figure 19-4) has a time base (period) and a time that the output stays high (duty cycle).

**FIGURE 19-4: CCP PWM OUTPUT**



# PIC16F193X/LF193X

## 19.5.1 PWM PERIOD

The PWM period is specified by the PRx register of Timerx. The PWM period can be calculated using the formula of Equation 19-1.

### EQUATION 19-1: PWM PERIOD

$$PWM\ Period = [(PR2x) + 1] \cdot 4 \cdot TOSC \cdot (TMRx\ Prescale\ Value)$$

**Note 1:** TOSC = 1/FOSC

When TMRx is equal to PRx, the following three events occur on the next increment cycle:

- TMRx is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

**Note:** The Timerx postscaler (see **Section 17.1 “Timer2/4/6 Operation”**) is not used in the determination of the PWM frequency.

## 19.5.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPRxL register and DCxB<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSBs and the DCxB<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and DCxB<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PRx and TMRx registers occurs). While using the PWM, the CCPRxH register is read-only.

Equation 19-2 is used to calculate the PWM pulse width.

Equation 19-3 is used to calculate the PWM duty cycle ratio.

### EQUATION 19-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxL:CCPxCON<5:4>) \cdot TOSC \cdot (TMRx\ Prescale\ Value)$$

### EQUATION 19-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PRx + 1)}$$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMRx register is concatenated with either the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timerx prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (see Figure 19-3).

## 19.5.3 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PRx is 255. The resolution is a function of the PRx register value as shown by Equation 19-4.

**EQUATION 19-4: PWM RESOLUTION**

$$Resolution = \frac{\log[4(PRx + 1)]}{\log(2)} \text{ bits}$$

Note: If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

**TABLE 19-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 32 MHz)**

PWM Frequency	1.95 kHz	7.81 kHz	31.25 kHz	125 kHz	250 kHz	333.3 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

**TABLE 19-5: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)**

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

**TABLE 19-6: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)**

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PRx Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

# PIC16F193X/LF193X

---

## 19.5.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMRx register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMRx will continue from its previous state.

## 19.5.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 8.0 “Oscillator Module (With Fail-Safe Clock Monitor)”** for additional details.

## 19.5.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

## 19.5.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Disable the PWM pin (CCPx) output driver(s) by setting the associated TRIS bit(s).
2. Load the PRx register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register and the DCxBx bits of the CCPxCON register, with the PWM duty cycle value.
5. Configure and start Timerx:
  - Clear the TMRxIF interrupt flag bit of the PIRx register. See Note below.
  - Configure the TxCKPS bits of the TxCON register with the Timerx prescale value.
  - Enable Timerx by setting the TMRxON bit of the T2CON register.
6. Enable PWM output pin:
  - Wait until Timerx overflows, TMRxIF bit of the PIR1 register is set. See Note below.
  - Enable the PWM pin (CCPx) output driver(s) by clearing the associated TRIS bit(s).

**Note:** In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

## 19.6 PWM (Enhanced Mode)

The Enhanced PWM mode can generate a PWM signal on up to four different output pins with up to 10-bits of resolution. It can do this through four different PWM output modes:

- Single PWM
- Half-Bridge PWM
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the P1M bits of the CCP1CON register must be set appropriately.

**Note:** The PWM Enhanced mode is available on the Enhanced Capture/Compare/PWM module (CCP1) only.

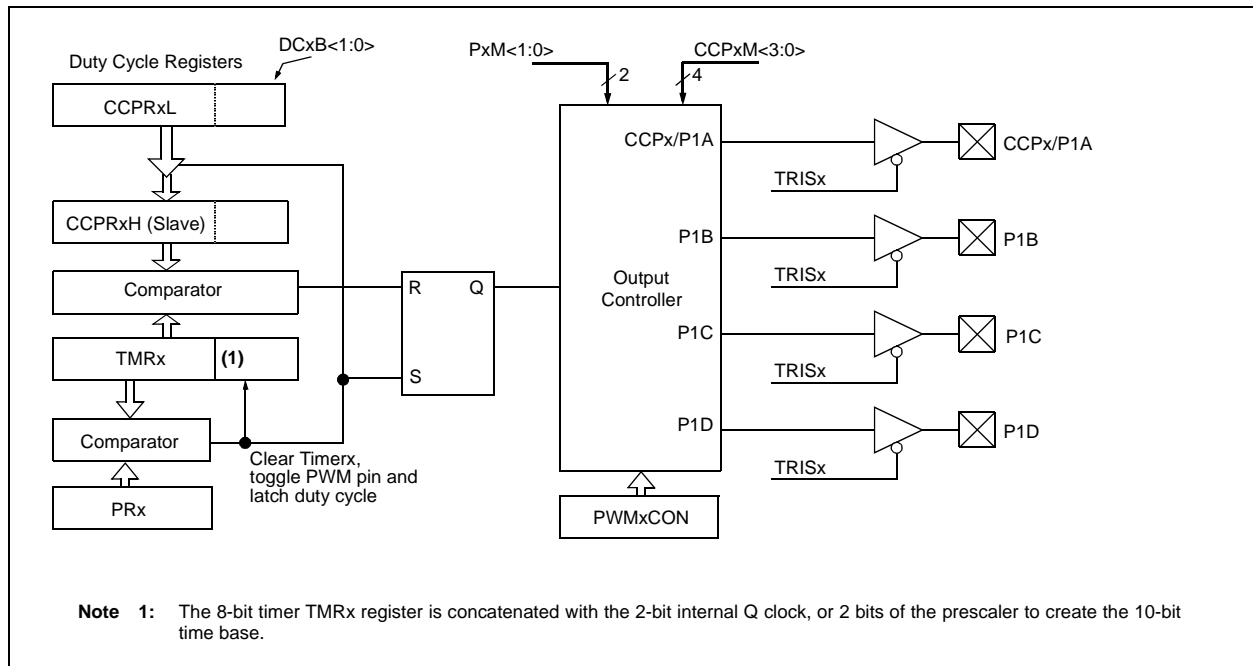
The PWM outputs are multiplexed with I/O pins and are designated P1A, P1B, P1C and P1D. The polarity of the PWM pins is configurable and is selected by setting the CCP1M bits in the CCP1CON register appropriately.

Table 19-7 shows the pin assignments for each Enhanced PWM mode.

Figure 19-5 shows an example of a simplified block diagram of the Enhanced PWM module.

**Note:** To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal.

**FIGURE 19-5: EXAMPLE SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE**



**Note 1:** The TRIS register value for each PWM output must be configured appropriately.

**2:** Clearing the CCPxCON register will relinquish ECCP control of all PWM output pins.

**3:** Any pin not used by an Enhanced PWM mode is available for alternate pin functions.

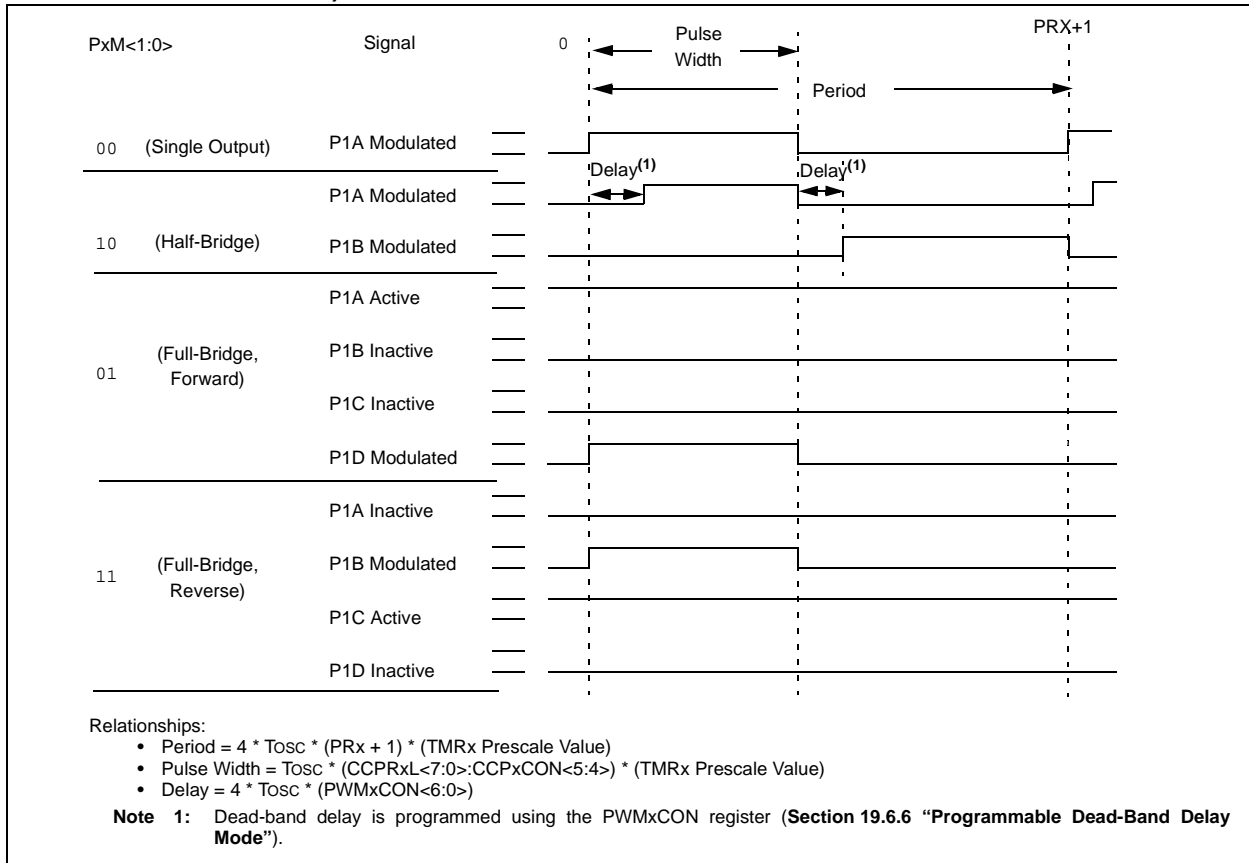
**TABLE 19-7: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES**

ECCP Mode	Pxm<1:0>	CCPx/P1A	P1B	P1C	P1D
Single	00	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>	Yes <sup>(1)</sup>
Half-Bridge	10	Yes	Yes	No	No
Full-Bridge, Forward	01	Yes	Yes	Yes	Yes
Full-Bridge, Reverse	11	Yes	Yes	Yes	Yes

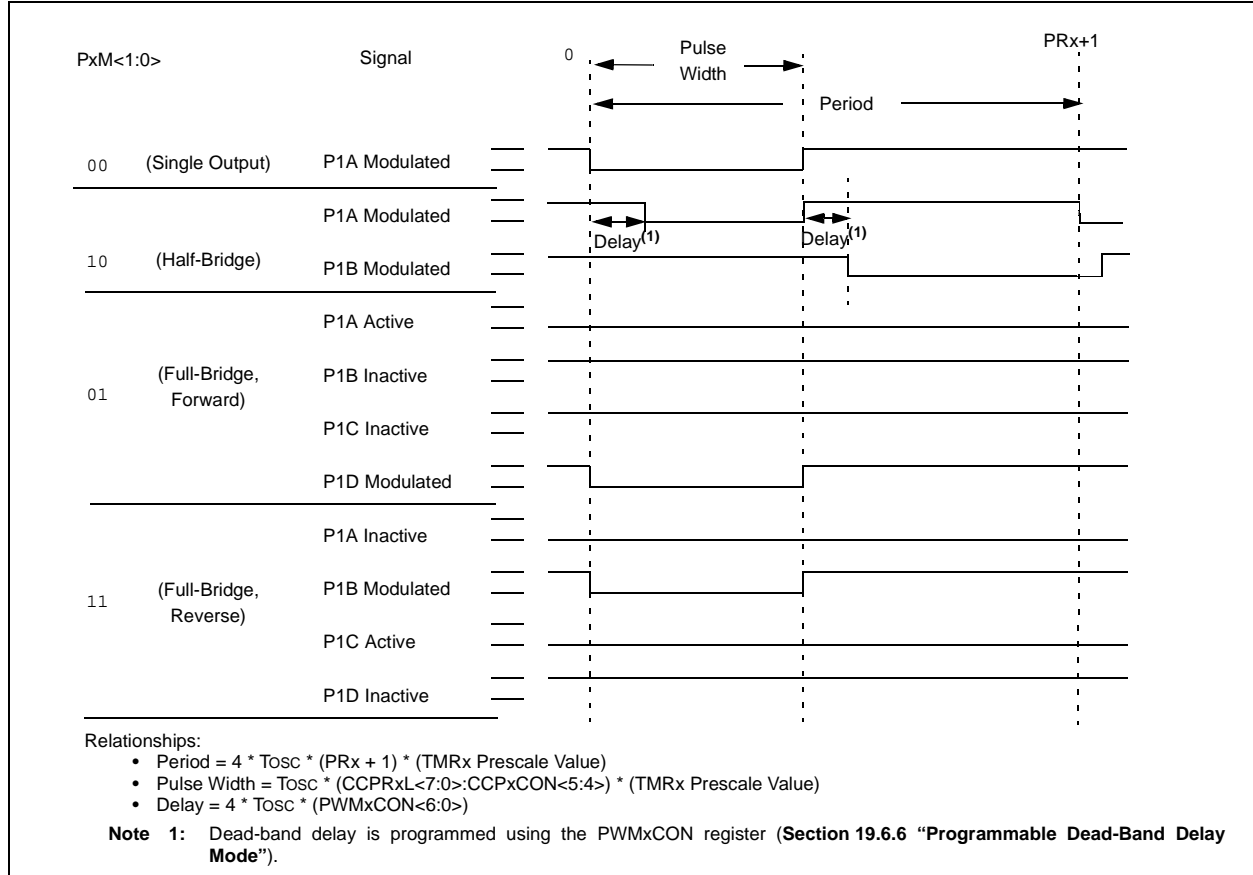
**Note 1:** Pulse Steering enables outputs in Single mode.

# PIC16F193X/LF193X

**FIGURE 19-6: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)**



**FIGURE 19-7: EXAMPLE ENHANCED PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)**



# PIC16F193X/LF193X

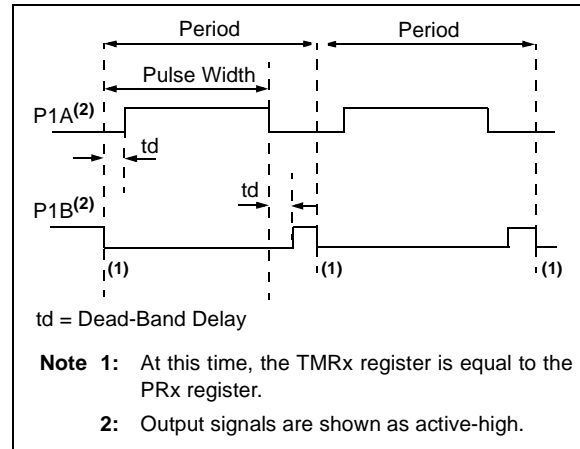
## 19.6.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCPx/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 19-9). This mode can be used for Half-Bridge applications, as shown in Figure 19-9, or for Full-Bridge applications, where four power switches are being modulated with two PWM signals.

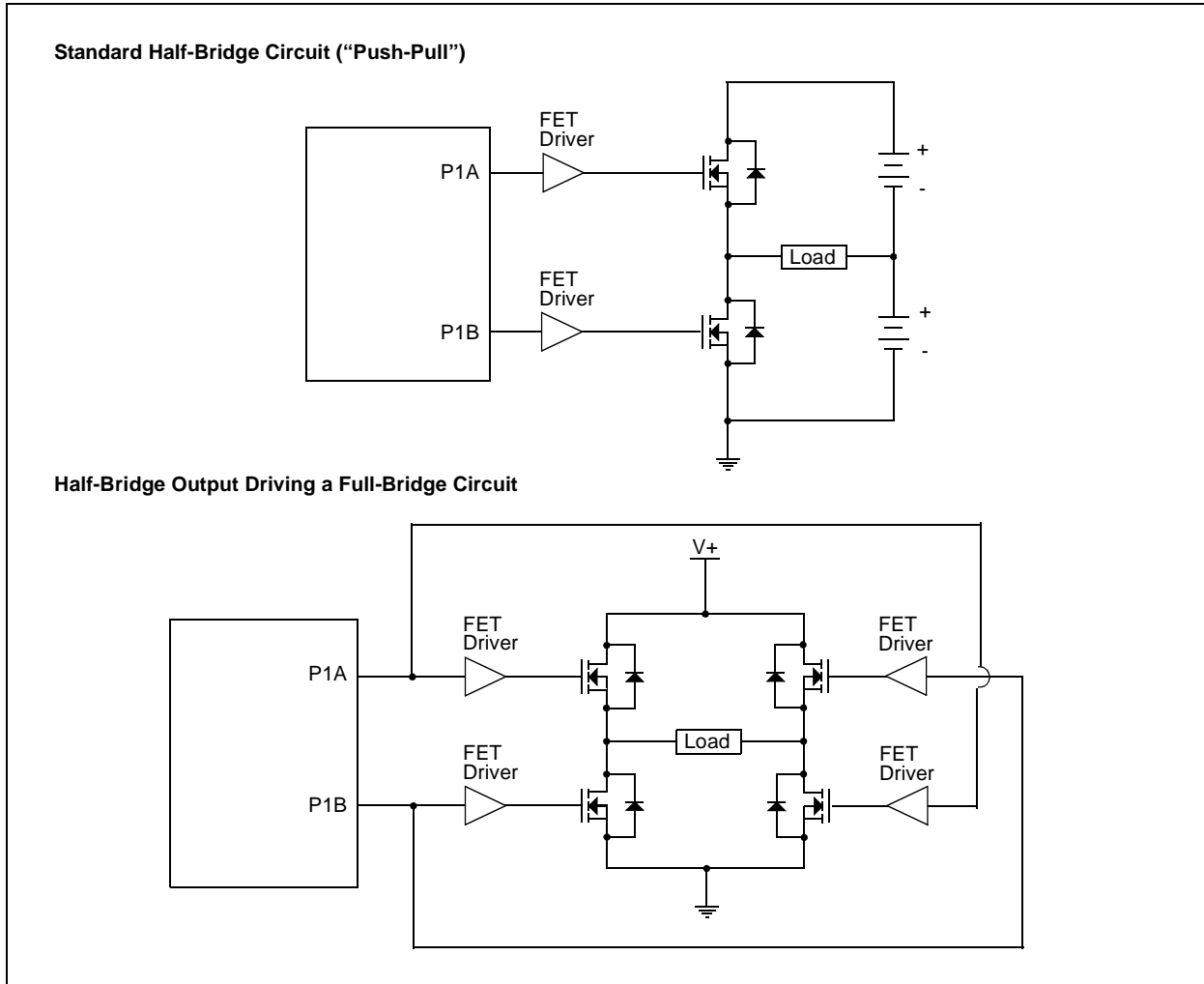
In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in Half-Bridge power devices. The value of the PDC<6:0> bits of the PWMxCON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 19.6.6 “Programmable Dead-Band Delay Mode”** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.

**FIGURE 19-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**



**FIGURE 19-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS**





## 19.6.2 FULL-BRIDGE MODE

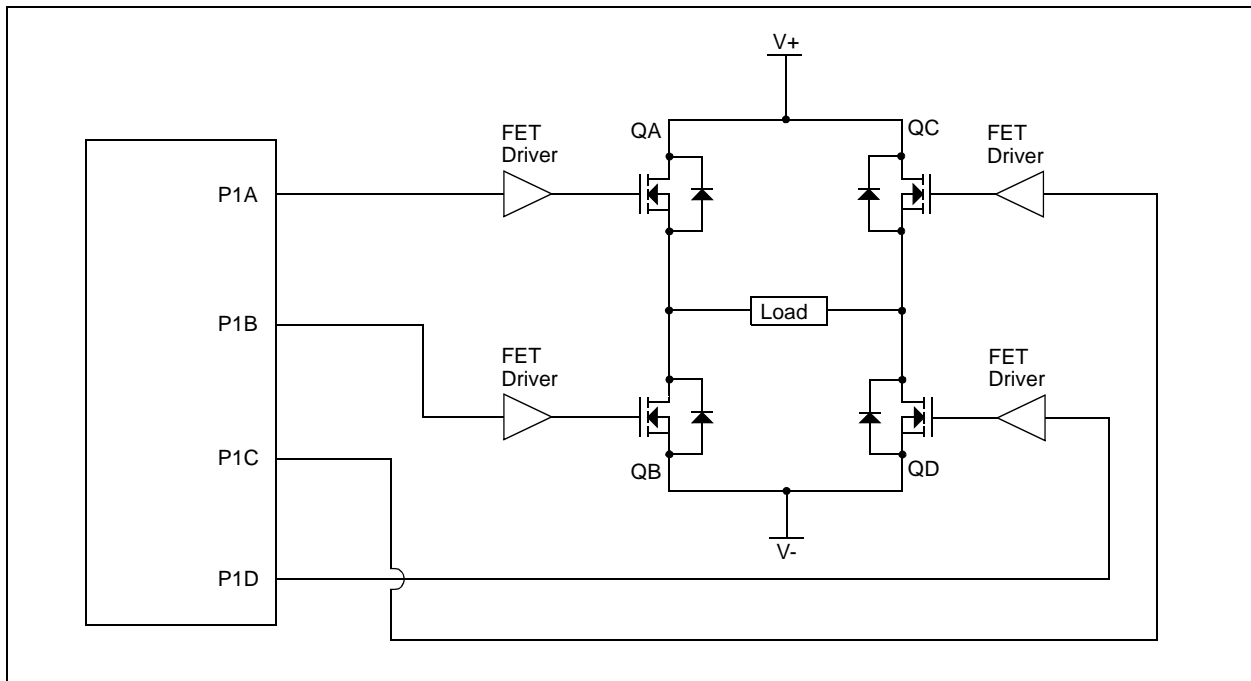
In Full-Bridge mode, all four pins are used as outputs. An example of Full-Bridge application is shown in Figure 19-10.

In the Forward mode, pin CCPx/P1A is driven to its active state, pin P1D is modulated, while P1B and P1C will be driven to their inactive state as shown in Figure 19-11.

In the Reverse mode, P1C is driven to its active state, pin P1B is modulated, while P1A and P1D will be driven to their inactive state as shown Figure 19-11.

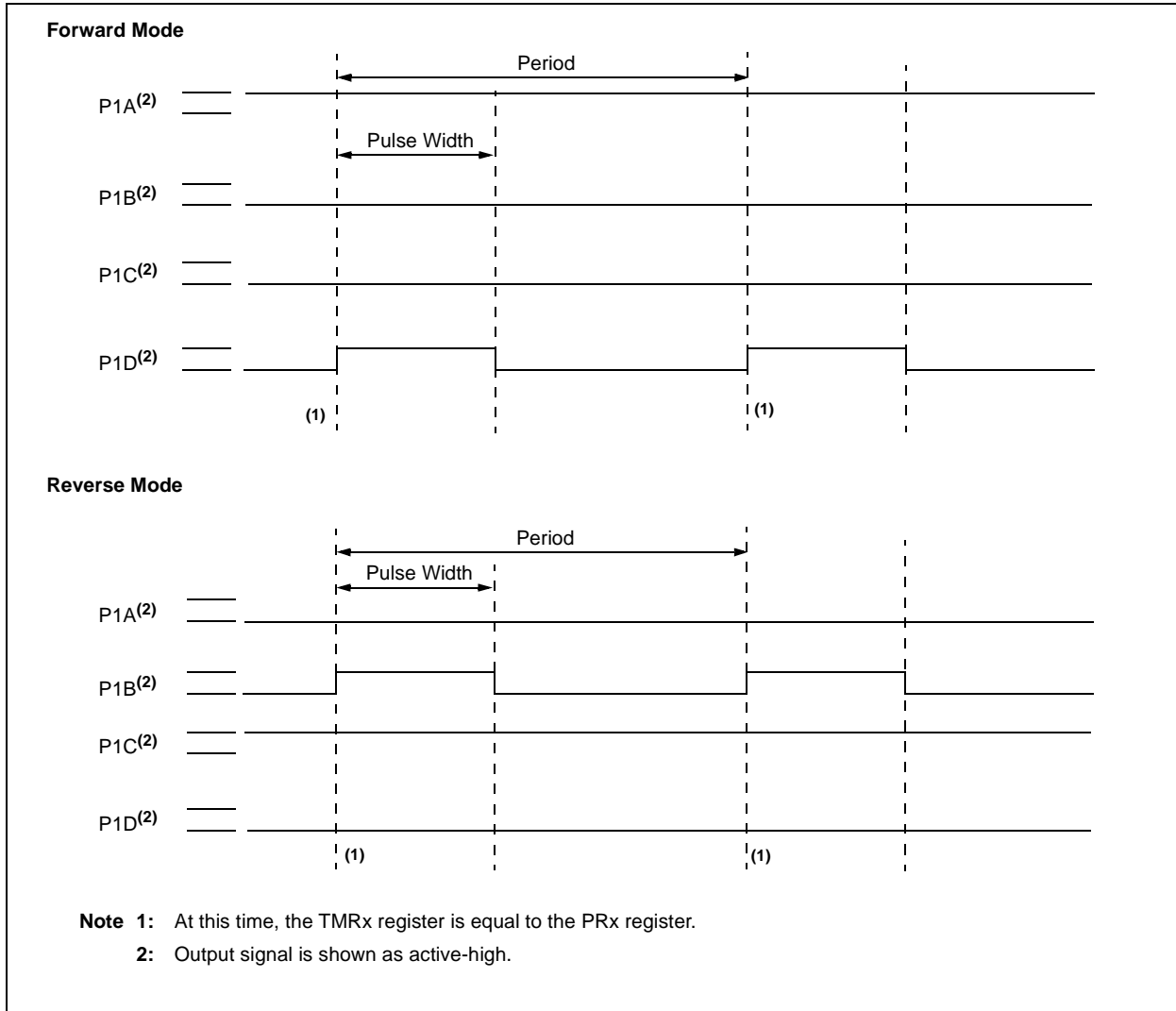
P1A, P1B, P1C and P1D outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the P1A, P1B, P1C and P1D pins as outputs.

**FIGURE 19-10: EXAMPLE OF FULL-BRIDGE APPLICATION**



# PIC16F193X/LF193X

FIGURE 19-11: EXAMPLE OF FULL-BRIDGE PWM OUTPUT



## 19.6.2.1 Direction Change in Full-Bridge Mode

In the Full-Bridge mode, the Pxm1 bit in the CCPxCON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will change to the new direction on the next PWM cycle.

A direction change is initiated in software by changing the Pxm1 bit of the CCPxCON register. The following sequence occurs four Timerx cycles prior to the end of the current PWM period:

- The modulated outputs (P1B and P1D) are placed in their inactive state.
- The associated unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction.
- PWM modulation resumes at the beginning of the next period.

See Figure 19-12 for an illustration of this sequence.

The Full-Bridge mode does not provide dead-band delay. As one output is modulated at a time, dead-band delay is generally not required. There is a situation where dead-band delay is required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn off time of the power switch, including the power device and driver circuit, is greater than the turn on time.

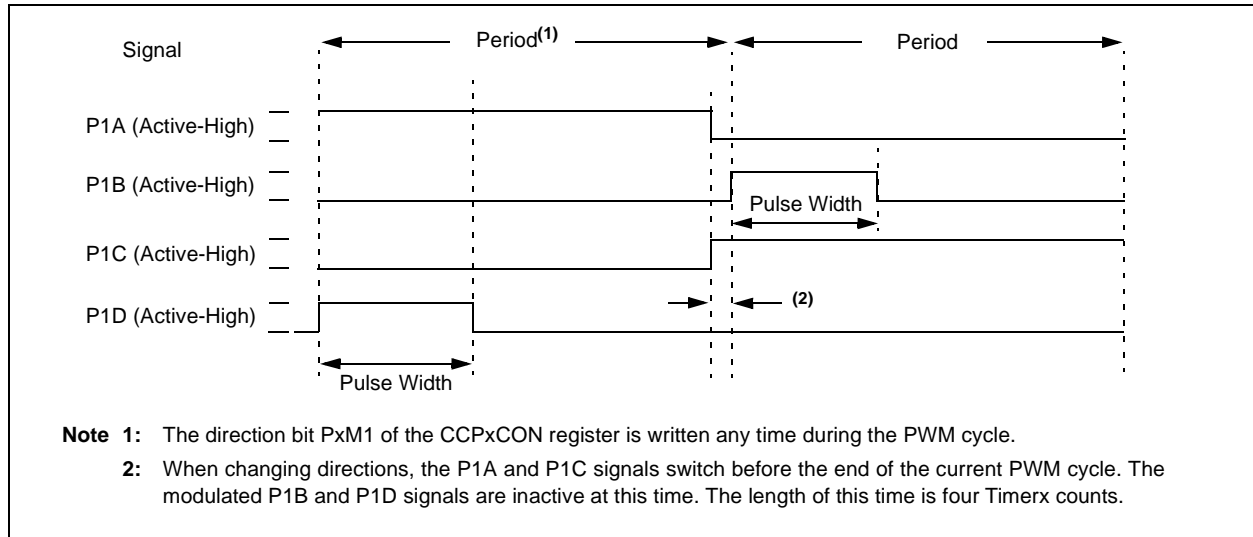
Figure 19-13 shows an example of the PWM direction changing from forward to reverse, at a near 100% duty cycle. In this example, at time t1, the output P1A and P1D become inactive, while output P1C becomes active. Since the turn off time of the power devices is longer than the turn on time, a shoot-through current will flow through power devices QC and QD (see Figure 19-10) for the duration of 't'. The same phenomenon will occur to power devices QA and QB for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce PWM duty cycle for one PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

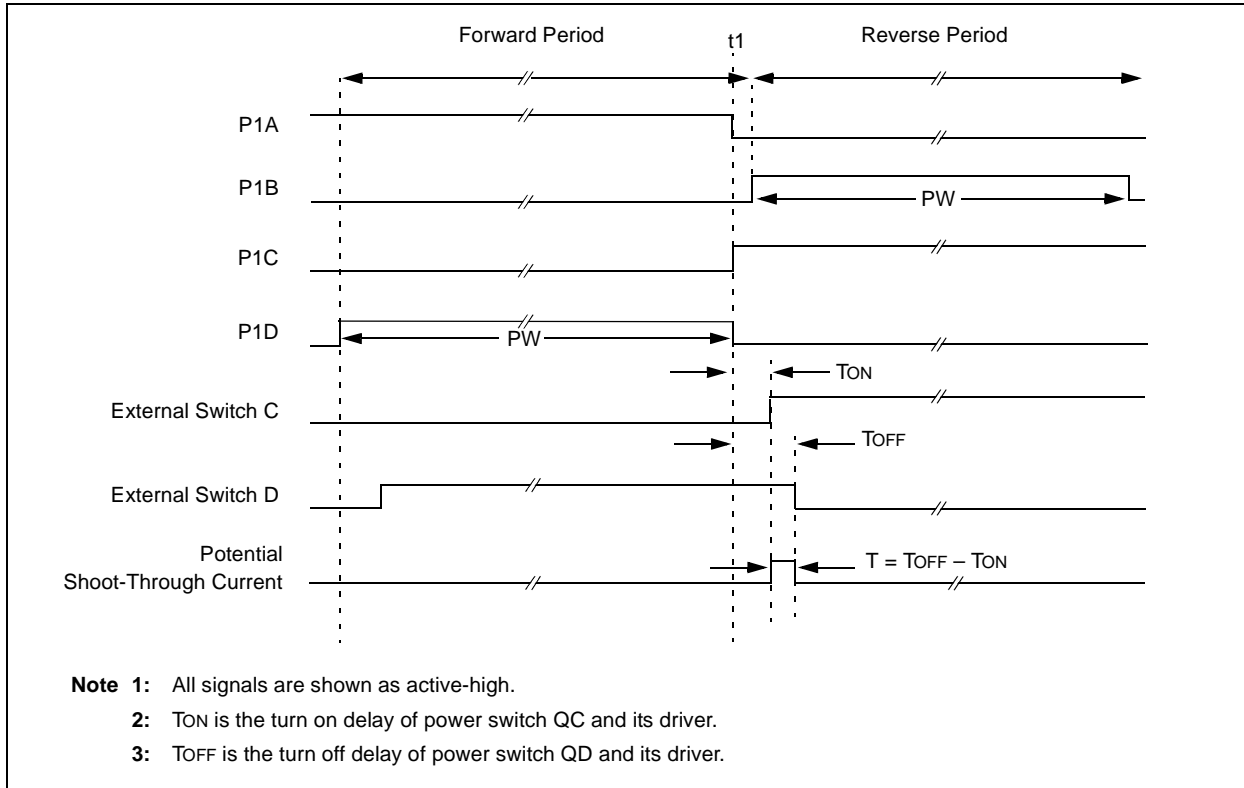
Other options to prevent shoot-through current may exist.

**FIGURE 19-12: EXAMPLE OF PWM DIRECTION CHANGE**



# PIC16F193X/LF193X

FIGURE 19-13: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



## 19.6.3 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

**Note:** When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The CCPxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMRxIF bit of the PIRx register being set as the second PWM period begins.

## 19.6.4 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the CCPxAS<2:0> bits of the CCPxAS register. A shutdown event may be generated by:

- A logic '0' on the INT pin
- Comparator Cx
- Setting the CCPxASE bit in firmware

A shutdown condition is indicated by the CCPxASE (Auto-Shutdown Event Status) bit of the CCPxAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:

The CCPxASE bit is set to '1'. The CCPxASE will remain set until cleared in firmware or an auto-restart occurs (see **Section 19.6.5 "Auto-Restart Mode"**).

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [P1A/P1C] and [P1B/P1D]. The state of each pin pair is determined by the PSSxAC and PSSxBD bits of the CCPxAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

# PIC16F193X/LF193X

## REGISTER 19-4: CCPxAS: CCPX AUTO-SHUTDOWN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPxASE	CCPxAS2	CCPxAS1	CCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
bit 7							bit 0

### Legend:

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

- bit 7      **CCPxASE:** CCPx Auto-Shutdown Event Status bit  
 1 = A shutdown event has occurred; CCPx outputs are in shutdown state  
 0 = CCPx outputs are operating
- bit 6-4    **CCxPAS<2:0>:** CCPx Auto-Shutdown Source Select bits  
 000 = Auto-shutdown is disabled  
 001 = Comparator C1 output low<sup>(1)</sup>  
 010 = Comparator C2 output low<sup>(1)</sup>  
 011 = Either Comparator C1 or C2 low<sup>(1)</sup>  
 100 = VIL on INT pin  
 101 = VIL on INT pin or Comparator C1 low<sup>(1)</sup>  
 110 = VIL on INT pin or Comparator C2 low<sup>(1)</sup>  
 111 = VIL on INT pin or Comparator C1 or Comparator C2 low<sup>(1)</sup>
- bit 3-2    **PSSxACx:** Pins P1A and P1C Shutdown State Control bits  
 00 = Drive pins P1A and P1C to '0'  
 01 = Drive pins P1A and P1C to '1'  
 1x = Pins P1A and P1C tri-state
- bit 1-0    **PSSxBDx:** Pins P1B and P1D Shutdown State Control bits  
 00 = Drive pins P1B and P1D to '0'  
 01 = Drive pins P1B and P1D to '1'  
 1x = Pins P1B and P1D tri-state

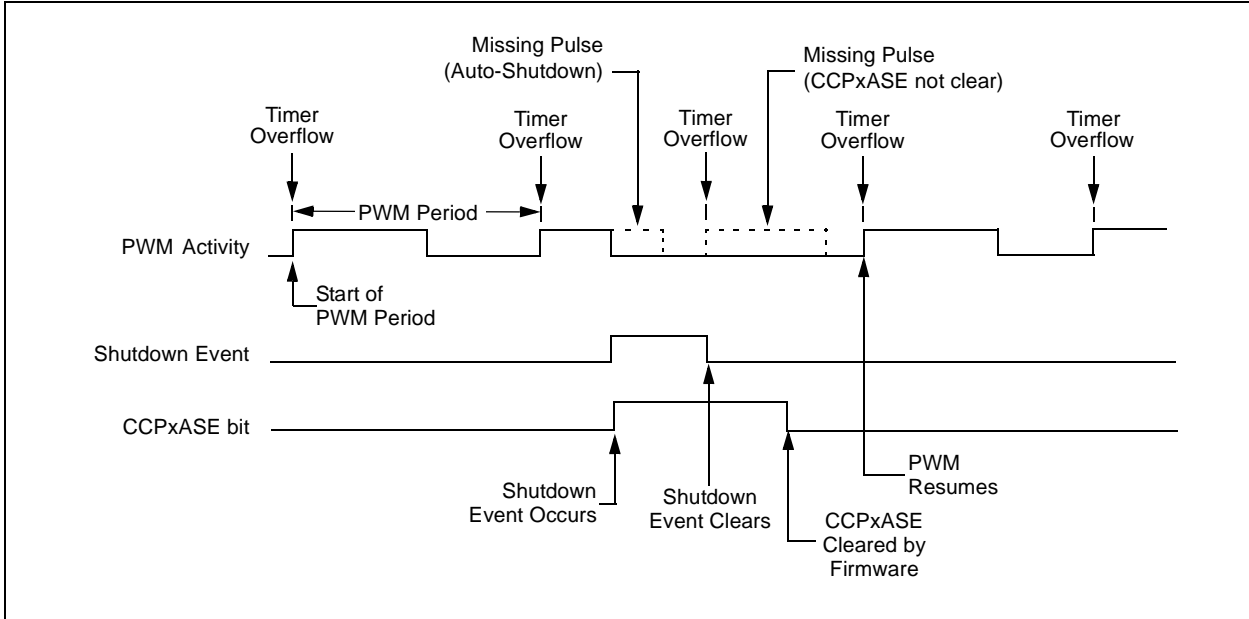
**Note 1:** If CxSYNC is enabled, the shutdown will be delayed by Timer1.

**Note 1:** The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.

**2:** Writing to the CCPxASE bit is disabled while an auto-shutdown condition persists.

**3:** Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart) the PWM signal will always restart at the beginning of the next PWM period.

**FIGURE 19-14: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PXRSEN = 0)**

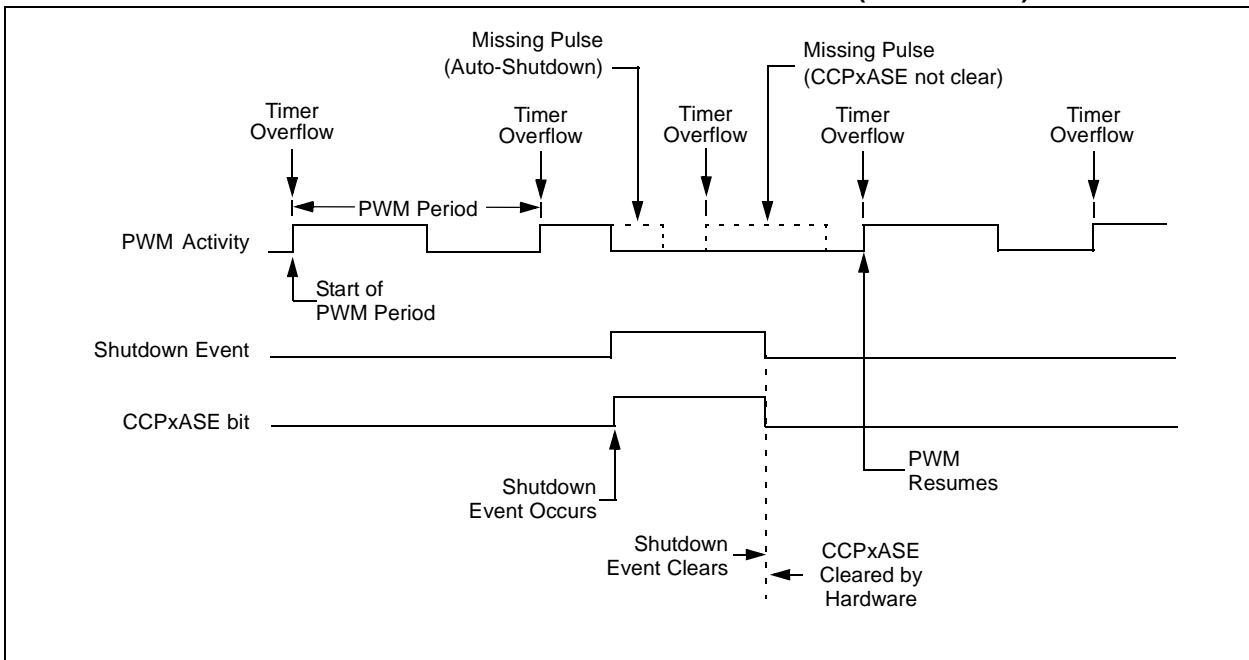


## 19.6.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PXRSEN bit in the PWMxCON register.

If auto-restart is enabled, the CCPxASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the CCPxASE bit will be cleared via hardware and normal operation will resume.

**FIGURE 19-15: PWM AUTO-SHUTDOWN WITH AUTO-RESTART (PXRSEN = 1)**



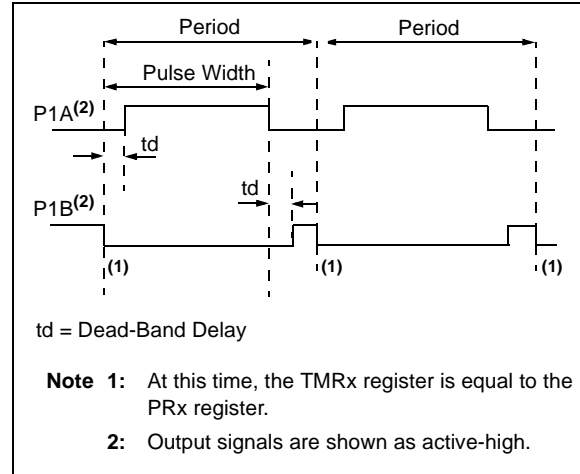
# PIC16F193X/LF193X

## 19.6.6 PROGRAMMABLE DEAD-BAND DELAY MODE

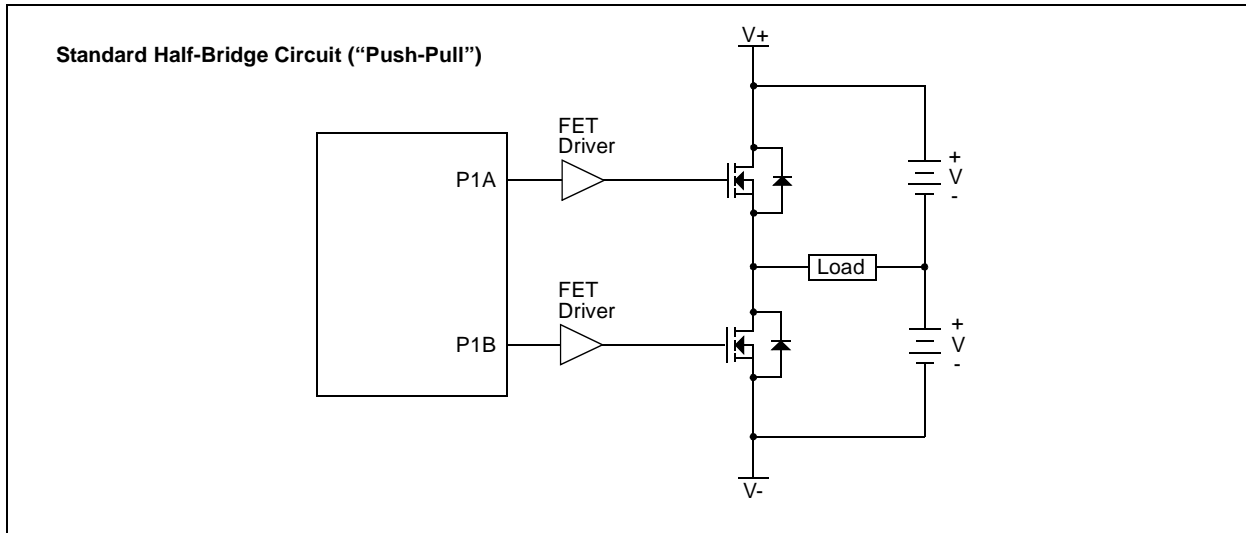
In Half-Bridge applications where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 19-16 for illustration. The lower seven bits of the associated PWMxCON register (Register 19-5) sets the delay period in terms of microcontroller instruction cycles ( $T_{CY}$  or  $4 T_{OSC}$ ).

**FIGURE 19-16: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**



**FIGURE 19-17: EXAMPLE OF HALF-BRIDGE APPLICATIONS**





## REGISTER 19-5: PWMxCON: ENHANCED PWM CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0
bit 7							bit 0

### Legend:

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

- bit 7      **PxRSEN:** PWM Restart Enable bit  
 1 = Upon auto-shutdown, the CCPxASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically  
 0 = Upon auto-shutdown, CCPxASE must be cleared in software to restart the PWM
- bit 6-0    **PxDC<6:0>:** PWM Delay Count bits  
 PxDCx = Number of Fosc/4 (4 \* Tosc) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it transitions active

**Note 1:** Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.

# PIC16F193X/LF193X

## 19.6.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can be simultaneously available on multiple pins.

Once the Single Output mode is selected (CCPxM<3:2> = 11 and PxM<1:0> = 00 of the CCPxCON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STRx<D:A> bits of the PSTRxCON register, as shown in Table 19-7.

**Note:** The associated TRIS bits must be set to output ('0') to enable the pin output driver in order to see the PWM signal on the pin.

While the PWM Steering mode is active, CCPxM<1:0> bits of the CCPxCON register select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to PWM Steering mode as described in **Section 19.6.4 “Enhanced PWM Auto-shutdown mode”**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

### REGISTER 19-6: PSTRXCON: PULSE STEERING CONTROL REGISTER<sup>(1)</sup>

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1	
—	—	—	STRxSYNC	STRxD	STRxC	STRxB	STRxA	
bit 7								bit 0

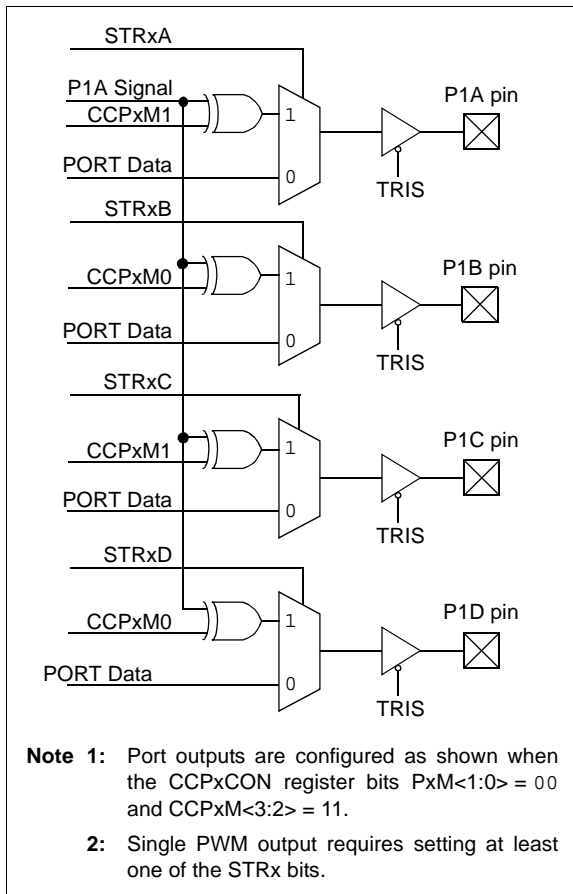
**Legend:**

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

- bit 7-5      **Unimplemented:** Read as '0'
- bit 4      **STRxSYNC:** Steering Sync bit
  - 1 = Output steering update occurs on next PWM period
  - 0 = Output steering update occurs at the beginning of the instruction cycle boundary
- bit 3      **STRxD:** Steering Enable bit D
  - 1 = P1D pin has the PWM waveform with polarity control from CCPxM<1:0>
  - 0 = P1D pin is assigned to port pin
- bit 2      **STRxC:** Steering Enable bit C
  - 1 = P1C pin has the PWM waveform with polarity control from CCPxM<1:0>
  - 0 = P1C pin is assigned to port pin
- bit 1      **STRxB:** Steering Enable bit B
  - 1 = P1B pin has the PWM waveform with polarity control from CCPxM<1:0>
  - 0 = P1B pin is assigned to port pin
- bit 0      **STRxA:** Steering Enable bit A
  - 1 = P1A pin has the PWM waveform with polarity control from CCPxM<1:0>
  - 0 = P1A pin is assigned to port pin

**Note 1:** The PWM Steering mode is available only when the CCPxCON register bits CCPxM<3:2> = 11 and PxM<1:0> = 00.

**FIGURE 19-18: SIMPLIFIED STEERING BLOCK DIAGRAM**



# PIC16F193X/LF193X

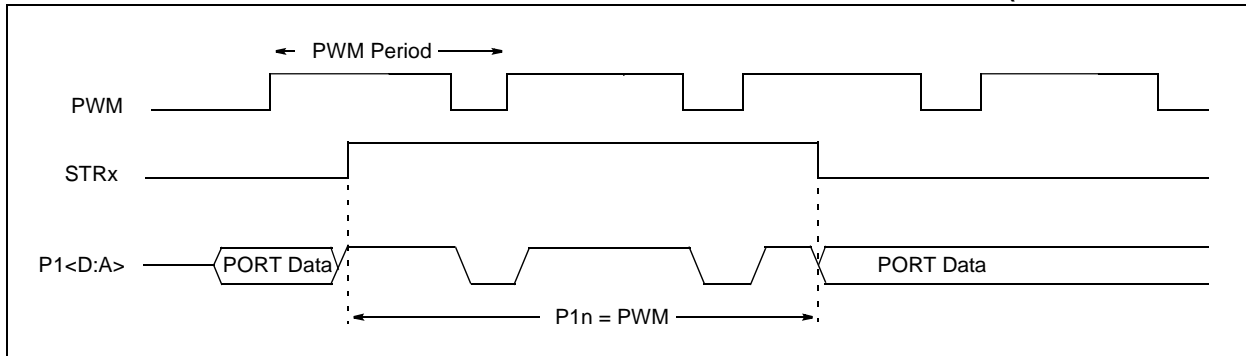
## 19.6.7.1 Steering Synchronization

The STRxSYNC bit of the PSTRxCON register gives the user two selections of when the steering event will happen. When the STRxSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRxCON register. In this case, the output signal at the P1<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

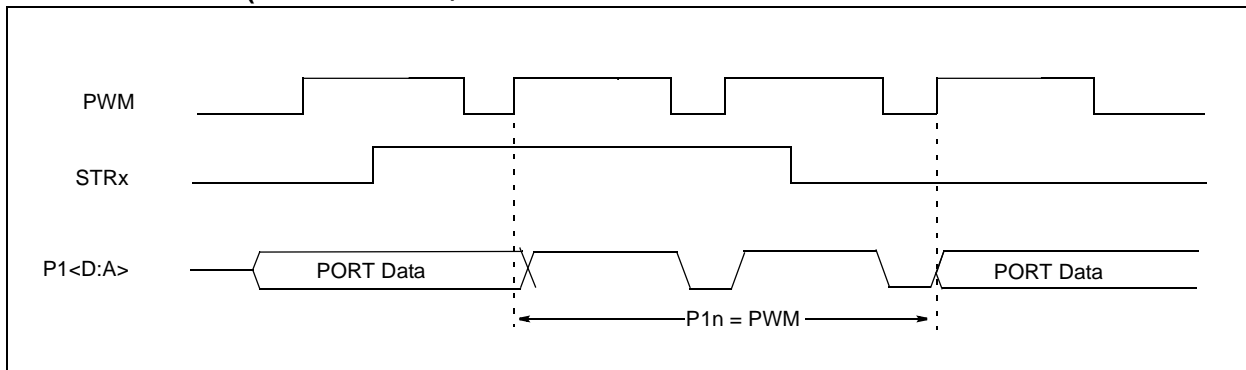
When the STRxSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 19-19 and 19-20 illustrate the timing diagrams of the PWM steering depending on the STRXSYNC setting.

**FIGURE 19-19: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRXSYNC = 0)**



**FIGURE 19-20: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRXSYNC = 1)**



# PIC16F193X/LF193X

**TABLE 19-8: REGISTERS ASSOCIATED WITH PWM**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCON	PxM1 <sup>(1)</sup>	PxM0 <sup>(1)</sup>	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	184
CCPxAS	CCPxASE	CCPxAS2	CCPxAS1	CCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0	204
CCPTMRS0	C4TSEL1	C4TSEL0	C3TSEL1	C3TSEL0	C2TSEL1	C2TSEL0	C1TSEL1	C1TSEL0	185
CCPTMRS1	—	—	—	—	—	—	C5TSEL1	C5TSEL0	186
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFE	TMR0IF	INTF	IOCFIF	73
PRx	Timerx Period Register								173*
PSTRxCON	—	—	—	STRxSYNC	STRxD	STRxC	STRxB	STRxA	208
PWMxCON	PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0	207
TxCON	—	TxOUTPS3	TxOUTPS2	TxOUTPS1	TxOUTPS0	TMRxON	TxCKPS1	TxCKPS0	175
TMRx	Timerx Module Register								173
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	91
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	97

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

**Note 1:** Applies to ECCP modules only.

\* Page provides register information.

# PIC16F193X/LF193X

---

NOTES:

## 20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

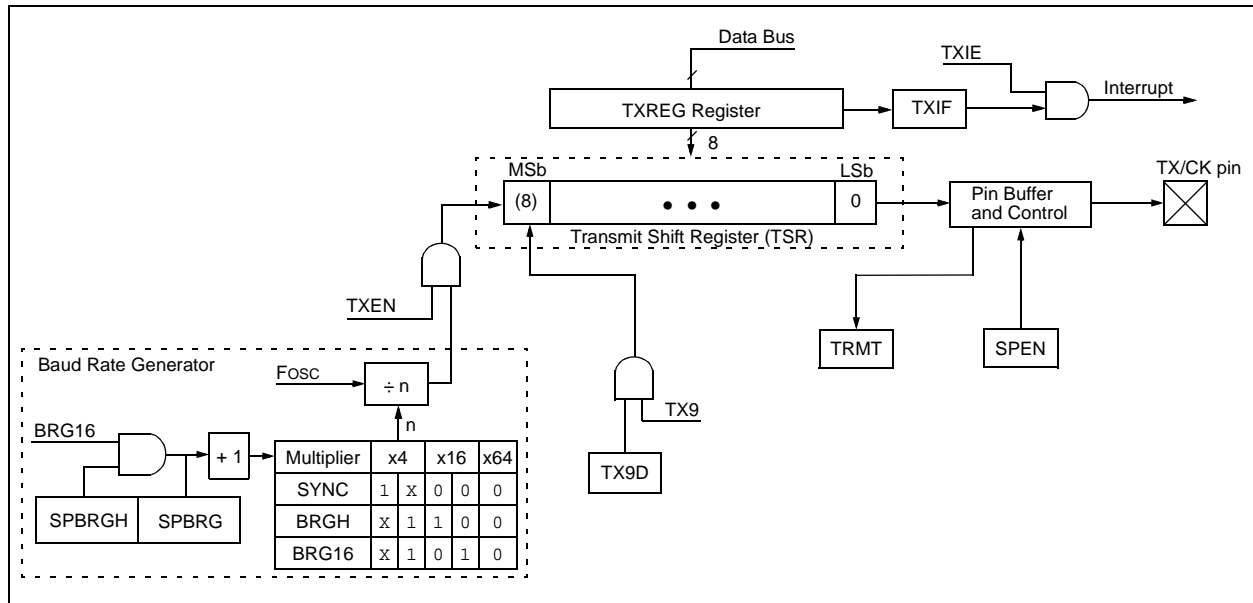
- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes
- Sleep operation

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

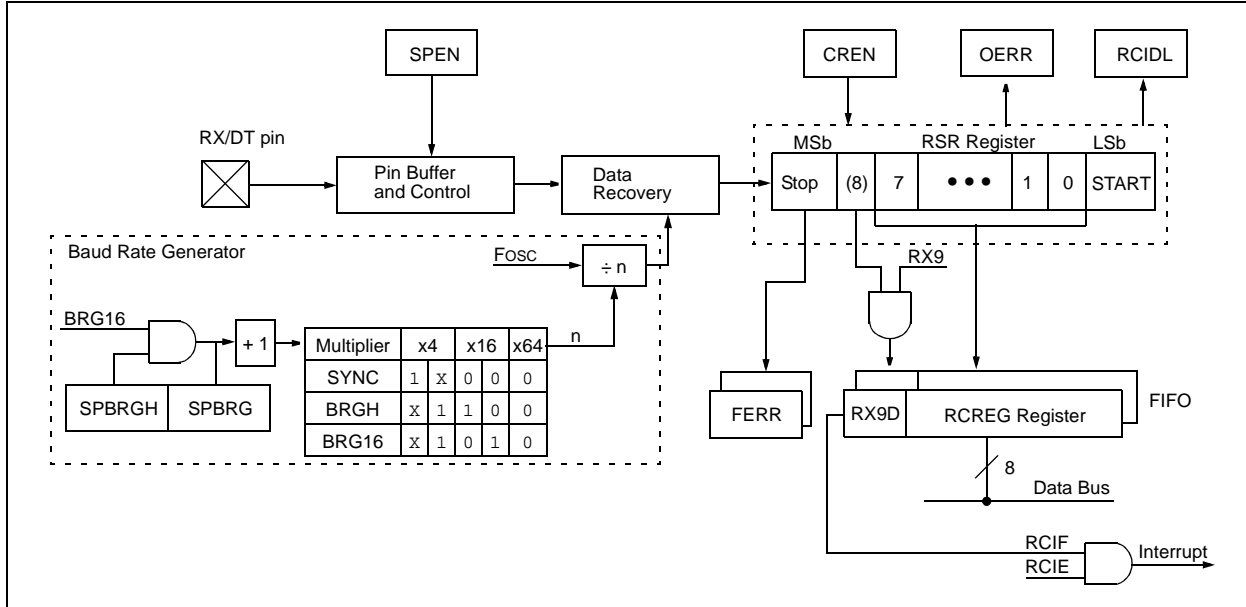
Block diagrams of the EUSART transmitter and receiver are shown in Figure 20-1 and Figure 20-2.

**FIGURE 20-1: EUSART TRANSMIT BLOCK DIAGRAM**



# PIC16F193X/LF193X

**FIGURE 20-2: EUSART RECEIVE BLOCK DIAGRAM**



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These registers are detailed in Register 20-1, Register 20-2 and Register 20-3, respectively.

When the receiver or transmitter section is not enabled then the corresponding RX or TX pin may be used for general purpose input and output.



## 20.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V<sub>OH</sub> mark state which represents a '1' data bit, and a V<sub>OL</sub> space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is 8 bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 20-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

### 20.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 20-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

#### 20.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output.

**Note 1:** The TXIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

#### 20.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one T<sub>cy</sub> immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

#### 20.1.1.3 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

# PIC16F193X/LF193X

## 20.1.1.4 TSR Status

The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

**Note:** The TSR register is not mapped in data memory, so it is not available to the user.

## 20.1.1.5 Transmitting 9-Bit Characters

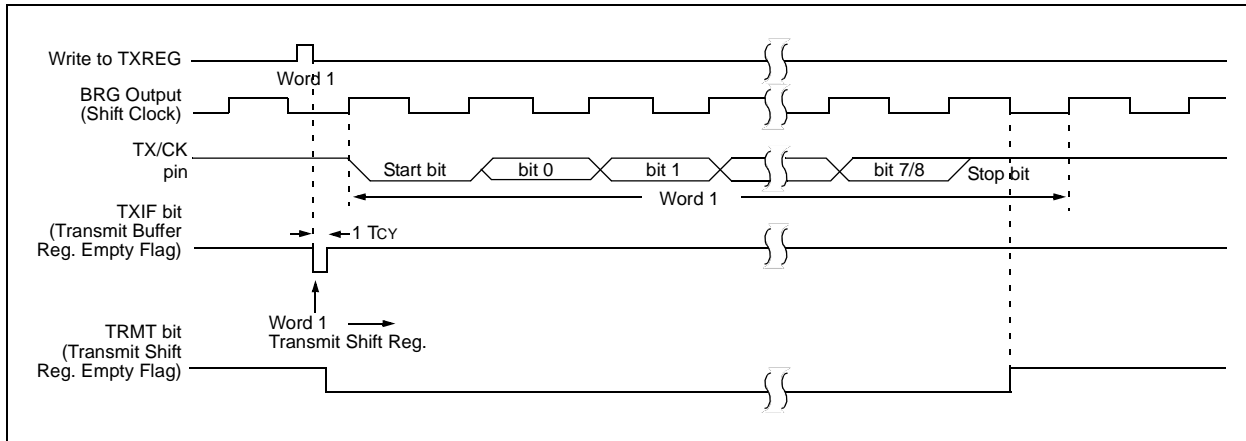
The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set the EUSART will shift 9 bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the 8 Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 20.1.2.7 “Address Detection”** for more information on the address mode.

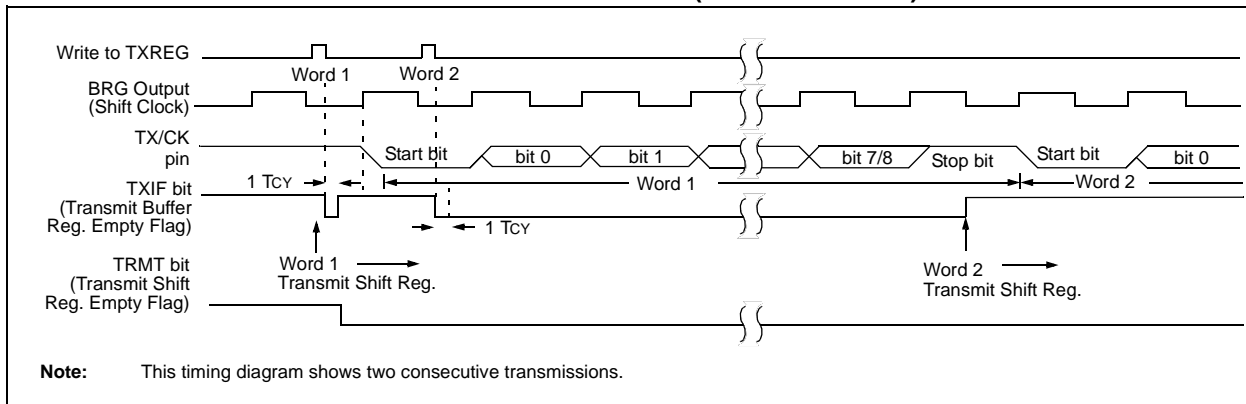
## 20.1.1.6 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 20.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the 8 Least Significant data bits are an address when the receiver is set for address detection.
4. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
5. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
6. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
7. Load 8-bit data into the TXREG register. This will start the transmission.

**FIGURE 20-3: ASYNCHRONOUS TRANSMISSION**



**FIGURE 20-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**



**Note:** This timing diagram shows two consecutive transmissions.

# PIC16F193X/LF193X

**TABLE 20-1: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	225*
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	225*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXREG	EUSART Transmit Data Register								215*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, – = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

\* Page provides register information.

# PIC16F193X/LF193X

## 20.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 20-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 8 or 9 bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

### 20.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the RX/DT I/O pin as an input.

**Note:** When the SPEN bit is set the TX/CK I/O pin is automatically configured as an output, regardless of the state of the corresponding TRIS bit and whether or not the EUSART transmitter is enabled. The PORT latch is disconnected from the output driver so it is not possible to use the TX/CK pin as a general purpose output.

### 20.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 20.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

**Note:** If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 20.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

### 20.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE interrupt enable bit of the PIE1 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

## 20.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

<b>Note:</b> If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.
--

## 20.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

## 20.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift 9 bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

## 20.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

# PIC16F193X/LF193X

## 20.1.2.8 Asynchronous Reception Set-up:

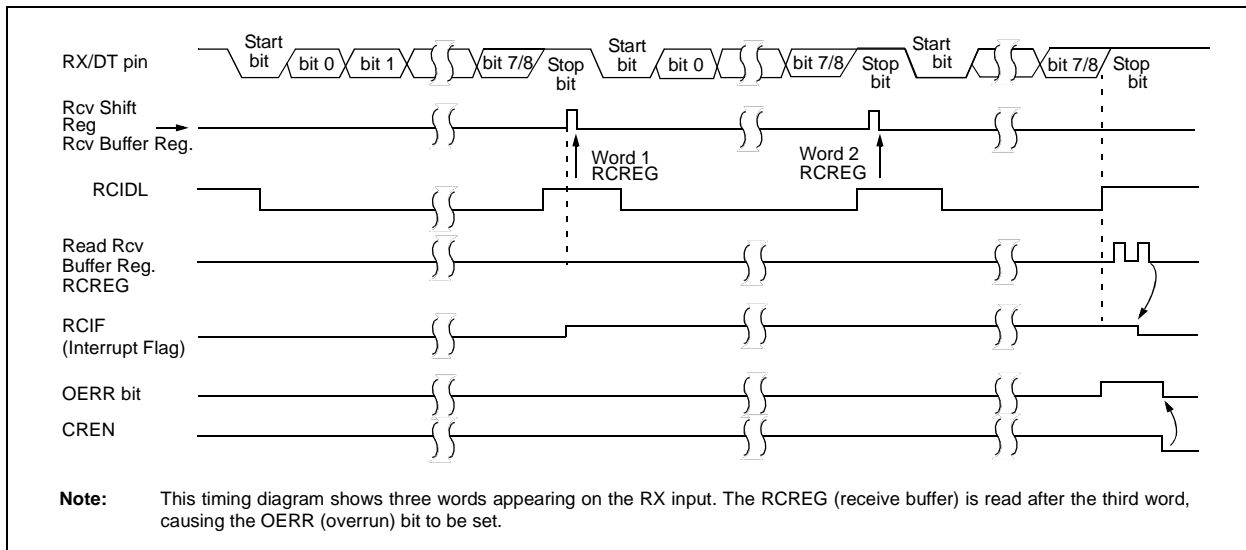
1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 20.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Enable reception by setting the CREN bit.
6. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
7. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
8. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register.
9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

## 20.1.2.9 9-bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 20.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. Enable 9-bit reception by setting the RX9 bit.
5. Enable address detection by setting the ADDEN bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
8. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
9. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

**FIGURE 20-5: ASYNCHRONOUS RECEPTION**



# PIC16F193X/LF193X

**TABLE 20-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCREG	EUSART Receive Data Register								218*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	225*
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	225*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Asynchronous Reception.

\* Page provides register information.

# PIC16F193X/LF193X

## 20.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as VDD or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 8.5 “Internal Clock Modes”** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 20.3.1 “Auto-Baud Detect”**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

### REGISTER 20-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

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

#### Legend:

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

- bit 7      **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<sup>(1)</sup>  
 1 = Transmit enabled  
 0 = Transmit disabled
- bit 4      **SYNC:** EUSART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode
- bit 3      **SENDB:** Send Break Character bit  
Asynchronous mode:  
 1 = Send Sync Break on next transmission (cleared by hardware upon completion)  
 0 = Sync Break transmission completed  
Synchronous mode:  
 Don't care
- 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:** Ninth bit of Transmit Data  
 Can be address/data bit or a parity bit.

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



# PIC16F193X/LF193X

**REGISTER 20-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER<sup>(1)</sup>**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

**Legend:**

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

- bit 7      **SPEN:** Serial Port Enable bit  
             1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)  
             0 = Serial port disabled (held in Reset)
- bit 6      **RX9:** 9-bit Receive Enable bit  
             1 = Selects 9-bit reception  
             0 = Selects 8-bit reception
- bit 5      **SREN:** Single Receive Enable bit  
             Asynchronous mode:  
             Don't care  
             Synchronous mode – Master:  
             1 = Enables single receive  
             0 = Disables single receive  
             This bit is cleared after reception is complete.  
             Synchronous mode – Slave  
             Don't care
- bit 4      **CREN:** Continuous Receive Enable bit  
             Asynchronous mode:  
             1 = Enables receiver  
             0 = Disables receiver  
             Synchronous mode:  
             1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)  
             0 = Disables continuous receive
- bit 3      **ADDEN:** Address Detect Enable bit  
             Asynchronous mode 9-bit (RX9 = 1):  
             1 = Enables address detection, enable interrupt and load 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  
             Asynchronous mode 8-bit (RX9 = 0):  
             Don't care
- 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:** Ninth bit of Received Data  
             This can be address/data bit or a parity bit and must be calculated by user firmware.

# PIC16F193X/LF193X

## REGISTER 20-3: BAUDCON: BAUD RATE CONTROL REGISTER

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

### Legend:

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

- bit 7      **ABDOVF:** Auto-Baud Detect Overflow bit  
Asynchronous mode:  
 1 = Auto-baud timer overflowed  
 0 = Auto-baud timer did not overflow  
Synchronous mode:  
 Don't care
- bit 6      **RCIDL:** Receive Idle Flag bit  
Asynchronous mode:  
 1 = Receiver is Idle  
 0 = Start bit has been received and the receiver is receiving  
Synchronous mode:  
 Don't care
- bit 5      **Unimplemented:** Read as '0'
- bit 4      **SCKP:** Synchronous Clock Polarity Select bit  
Asynchronous mode:  
 1 = Transmit inverted data to the RB7/TX/CK pin  
 0 = Transmit non-inverted data to the RB7/TX/CK pin  
Synchronous mode:  
 1 = Data is clocked on rising edge of the clock  
 0 = Data is clocked on falling edge of the clock
- bit 3      **BRG16:** 16-bit Baud Rate Generator bit  
 1 = 16-bit Baud Rate Generator is used  
 0 = 8-bit Baud Rate Generator is used
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **WUE:** Wake-up Enable bit  
Asynchronous mode:  
 1 = Receiver is waiting for a falling edge. No character will be received, byte RCIF will be set. WUE will automatically clear after RCIF is set.  
 0 = Receiver is operating normally  
Synchronous mode:  
 Don't care
- bit 0      **ABDEN:** Auto-Baud Detect Enable bit  
Asynchronous mode:  
 1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)  
 0 = Auto-Baud Detect mode is disabled  
Synchronous mode:  
 Don't care

## 20.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH, SPBRG register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 20-3 contains the formulas for determining the baud rate. Example 20-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 20-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRG register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

### EXAMPLE 20-1: CALCULATING BAUD RATE ERROR

For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64([SPBRGH:SPBRG] + 1)}$$

Solving for SPBRGH:SPBRG:

$$X = \frac{F_{OSC}}{\text{Desired Baud Rate} \cdot 64} - 1$$

$$= \frac{16000000}{9600 \cdot 64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

# PIC16F193X/LF193X

**TABLE 20-3: BAUD RATE FORMULAS**

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64 (n+1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16 (n+1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{osc}/[4 (n+1)]$
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

**Legend:** x = Don't care, n = value of SPBRGH, SPBRG register pair

**TABLE 20-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	225*
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	225*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

\* Page provides register information.

# PIC16F193X/LF193X

**TABLE 20-5: BAUD RATES FOR ASYNCHRONOUS MODES**

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.64k	2.12	16	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

# PIC16F193X/LF193X

**TABLE 20-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

# PIC16F193X/LF193X

**TABLE 20-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

# PIC16F193X/LF193X

## 20.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCON register starts the auto-baud calibration sequence (Figure 20-6). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Table 20-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH, SPBRG register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRG register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 20-6. During ABD, both the SPBRGH and SPBRG registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH

and SPBRG registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

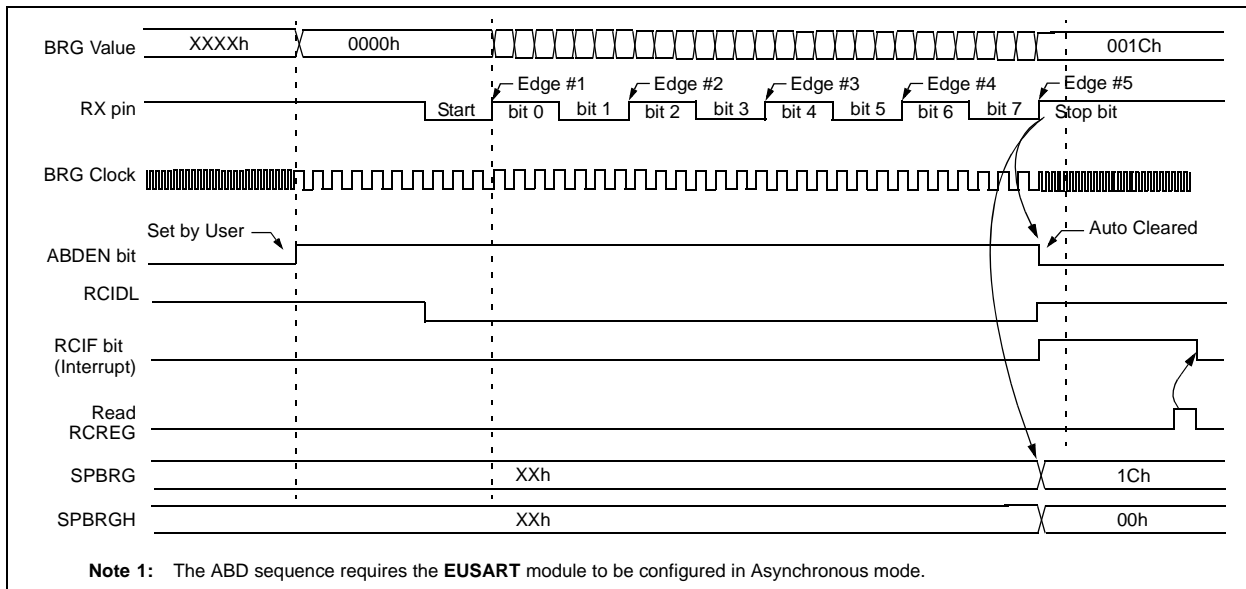
- Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see Section 20.3.3 "Auto-Wake-up on Break").
- 2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.
- 3:** During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRG register pair.

**TABLE 20-6: BRG COUNTER CLOCK RATES**

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

**Note:** During the ABD sequence, SPBRG and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

**FIGURE 20-6: AUTOMATIC BAUD RATE CALIBRATION**





## 20.3.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRG register pair. After the ABDOVF has been set, the counter continues to count until the fifth rising edge is detected on the RX pin. Upon detecting the fifth RX edge, the hardware will set the RCIF interrupt flag and clear the ABDEN bit of the BAUDCON register. The RCIF flag can be subsequently cleared by reading the RCREG register. The ABDOVF flag of the BAUDCON register can be cleared by software directly.

To terminate the auto-baud process before the RCIF flag is set, clear the ABDEN bit then clear the ABDOVF bit of the BAUDCON register. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

## 20.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 20-7), and asynchronously if the device is in Sleep mode (Figure 20-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

## 20.3.3.1 Special Considerations

### Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be 10 or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

### Oscillator Startup Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

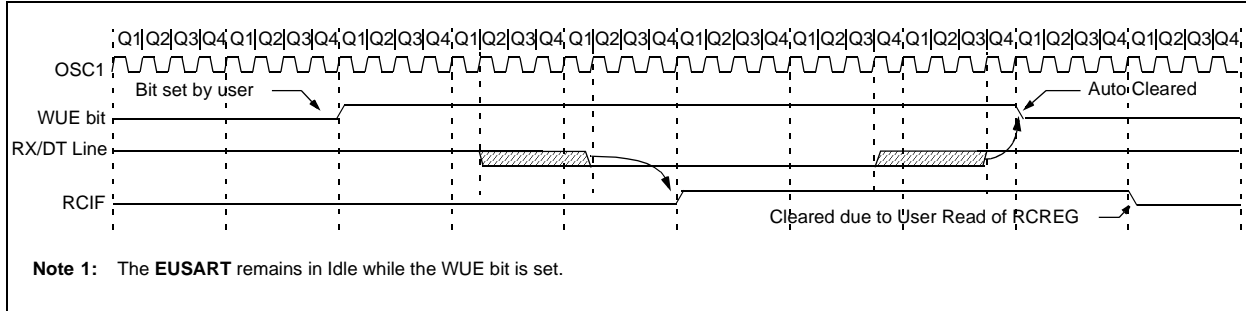
### WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

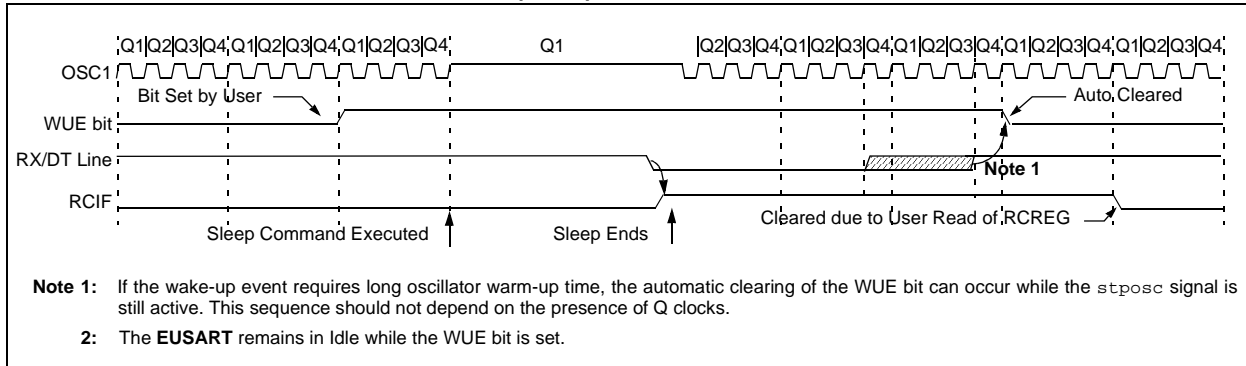
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

# PIC16F193X/LF193X

**FIGURE 20-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**



**FIGURE 20-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



## 20.3.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 20-9 for the timing of the Break character sequence.

### 20.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.

5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

## 20.3.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

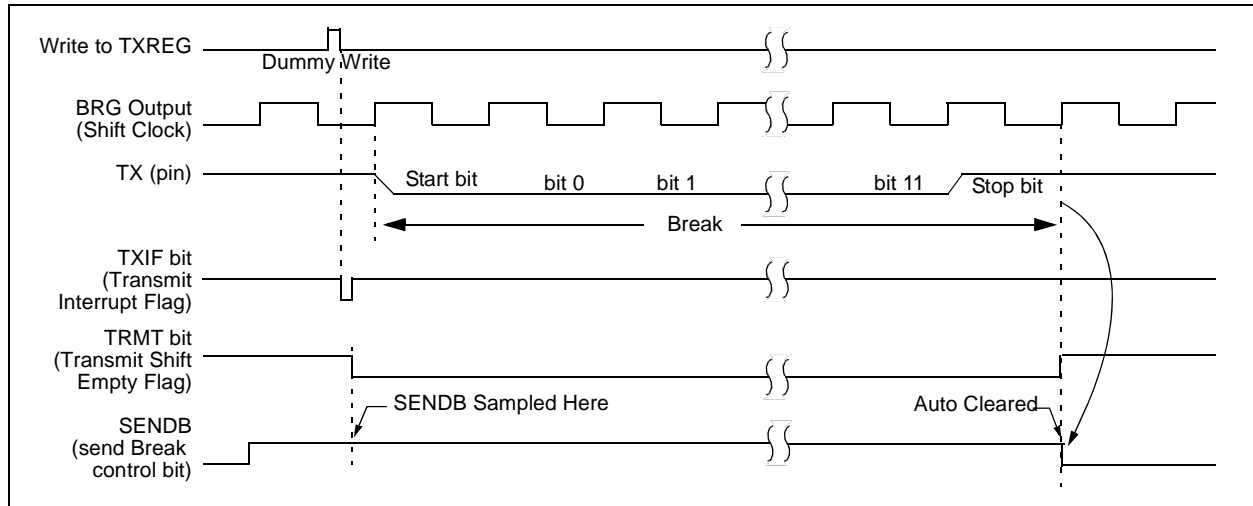
A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 20.3.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCON register before placing the EUSART in Sleep mode.

**FIGURE 20-9: SEND BREAK CHARACTER SEQUENCE**



# PIC16F193X/LF193X

## 20.4 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

### 20.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

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

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

#### 20.4.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

#### 20.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock.

Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

#### 20.4.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

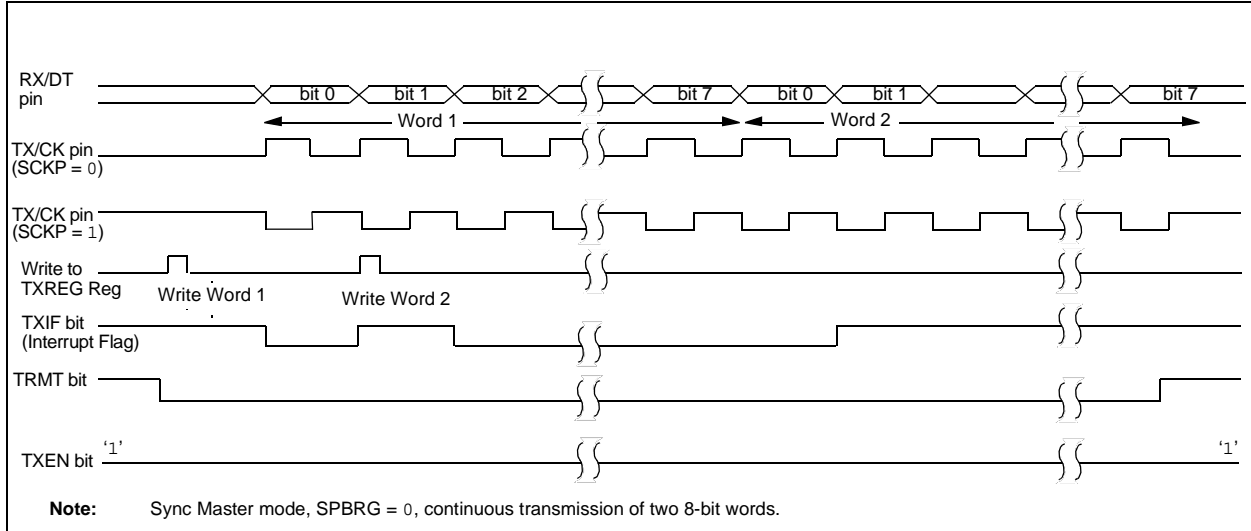
Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

**Note:** The TSR register is not mapped in data memory, so it is not available to the user.

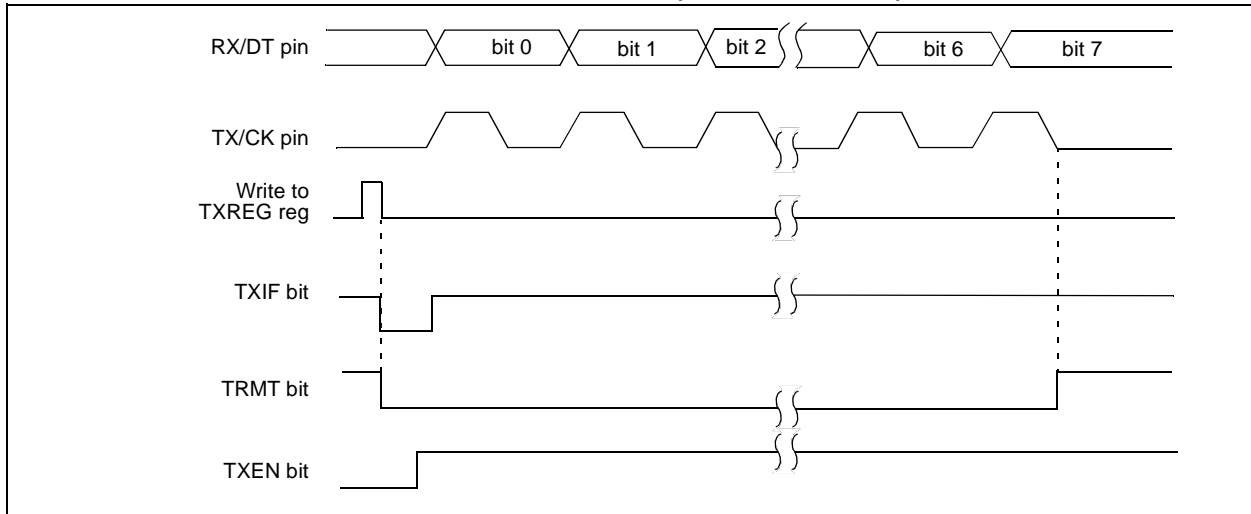
#### 20.4.1.4 Synchronous Master Transmission Set-up:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 20.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

**FIGURE 20-10: SYNCHRONOUS TRANSMISSION**



**FIGURE 20-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



**TABLE 20-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	225*
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	225*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXREG	EUSART Transmit Data Register								215*
TXSTA	CSRC	TX9	TXEN	SYNC	SENCB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Transmission.

\* Page provides register information.

# PIC16F193X/LF193X

---

## 20.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

## 20.4.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

## 20.4.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is

set then the error condition is cleared by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

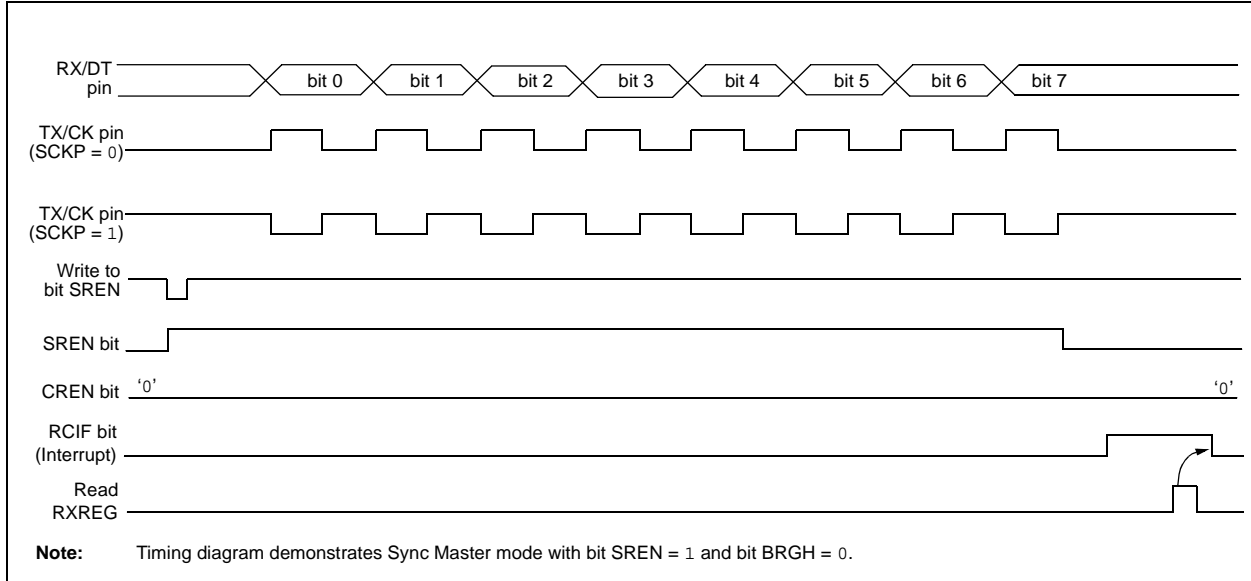
## 20.4.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the 8 Least Significant bits from the RCREG.

## 20.4.1.9 Synchronous Master Reception Set-up:

1. Initialize the SPBRGH, SPBRG register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
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, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set bit RX9.
6. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
7. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the 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 an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

**FIGURE 20-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



**TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCREG	EUSART Receive Data Register								218*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	225*
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	225*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXSTA	CSRC	TX9	TXEN	SYNC	SENCB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

\* Page provides register information.

# PIC16F193X/LF193X

## 20.4.2 SYNCHRONOUS SLAVE MODE

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

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

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

### 20.4.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 20.4.1.3 “Synchronous Master Transmission”**), 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:

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

### 20.4.2.2 Synchronous Slave Transmission Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the CREN and SREN bits.
3. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit transmission is desired, set the TX9 bit.
5. Enable transmission by setting the TXEN bit.
6. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
7. Start transmission by writing the Least Significant 8 bits to the TXREG register.

**TABLE 20-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXREG	EUSART Transmit Data Register								215*
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

\* Page provides register information.



## 20.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 20.4.1.5 “Synchronous Master Reception”**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a “don’t care” in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

## 20.4.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
3. If 9-bit reception is desired, set the RX9 bit.
4. Set the CREN bit to enable reception.
5. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
6. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
7. Retrieve the 8 Least Significant bits from the receive FIFO by reading the RCREG register.
8. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

**TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
RCREG	EUSART Receive Data Register								218*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	223
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	222

**Legend:** x = unknown, – = unimplemented read as ‘0’. Shaded cells are not used for Synchronous Slave Reception.

\* Page provides register information.

# PIC16F193X/LF193X

---

## 20.5 EUSART Operation During Sleep

The EUSART will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

### 20.5.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Reception (see **Section 20.4.2.4 “Synchronous Slave Reception Set-up:”**).
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set. Thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the GIE global interrupt enable bit of the INTCON register is also set, then the Interrupt Service Routine at address 004h will be called.

### 20.5.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Transmission (see **Section 20.4.2.2 “Synchronous Slave Transmission Set-up:”**).
- The TXIF interrupt flag must be cleared by writing the output data to the TXREG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXREG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TXREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the GIE global interrupt enable bit is also set then the Interrupt Service Routine at address 0004h will be called.

## 21.0 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 PIC16F193X/LF193X device, the module drives the panels of up to four commons and up to 24 segments. The LCD module 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 common pins:
  - Static (1 common)
  - 1/2 multiplex (2 commons)
  - 1/3 multiplex (3 commons)
  - 1/4 multiplex (4 commons)

- Segment pins up to:
  - 16 (PIC16F1933/1936/1938/  
PIC16LF1933/1936/1938)
  - 24 (PIC16F1934/1937/1939/  
PIC16LF1934/1937/1939)
- Static, 1/2 or 1/3 LCD Bias

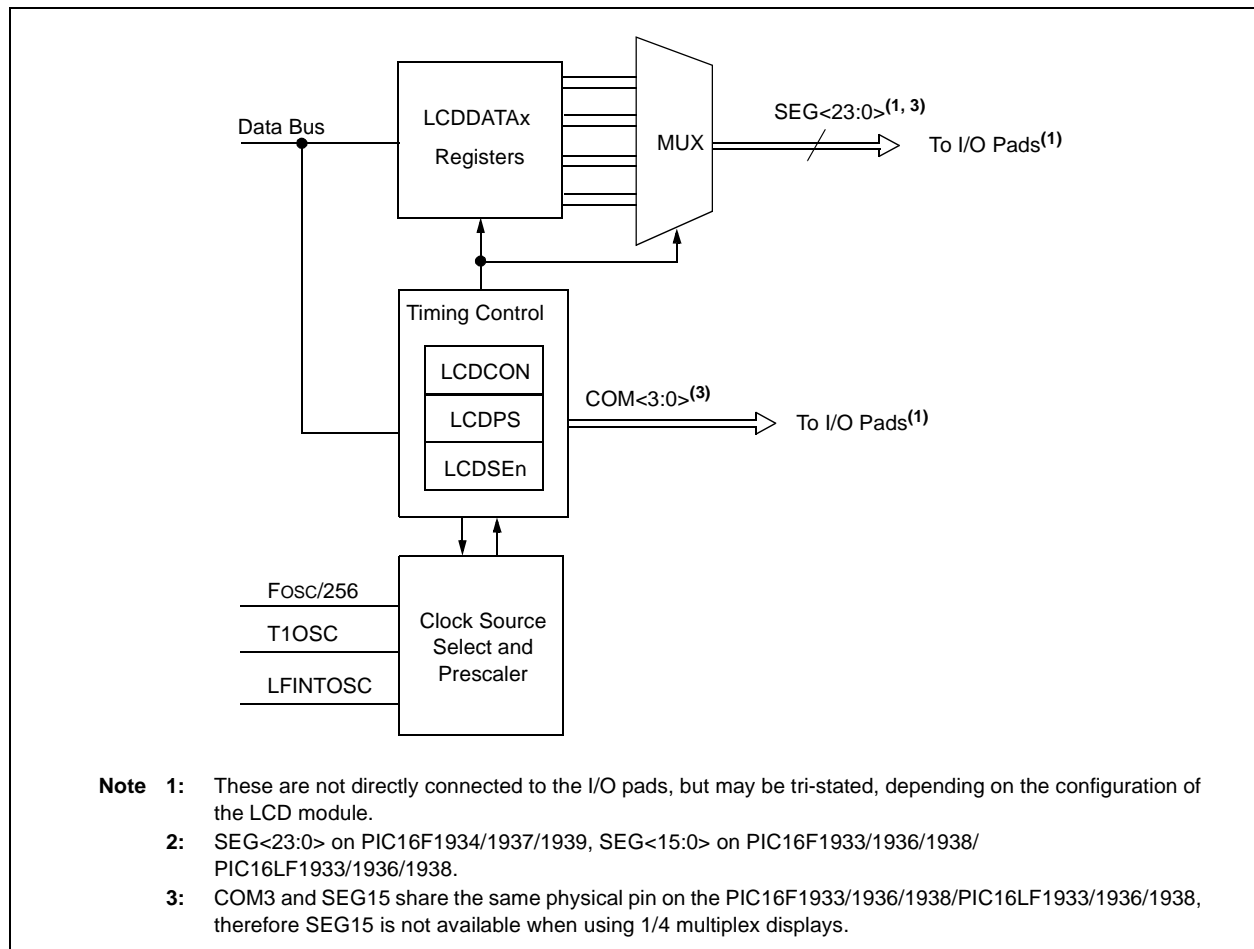
**Note:** COM3 and SEG15 share the same physical pin on the PIC16F1933/1936/1938/  
PIC16LF1933/1936/1938, therefore SEG15 is not available when using 1/4 multiplex displays.

### 21.1 LCD Registers

The module contains the following registers:

- LCD Control register (LCDCON)
- LCD Phase register (LCDPS)
- LCD Reference Ladder register (LCDRL)
- LCD Contrast Control register (LDCST)
- LCD Reference Voltage Control register (LCDREF)
- Up to 3 LCD Segment Enable registers (LCDSEn)
- Up to 12 LCD data registers (LCDDATAn)

**FIGURE 21-1: LCD DRIVER MODULE BLOCK DIAGRAM**



# PIC16F193X/LF193X

**TABLE 21-1: LCD SEGMENT AND DATA REGISTERS**

Device	# of LCD Registers	
	Segment Enable	Data
PIC16F1933/1936/1938/ PIC16LF1933/1936/1938	2	8
PIC16F1934/1937/1939/ PIC16LF1934/1937/1939	3	12

The LCDCON register (Register 21-1) controls the operation of the LCD driver module. The LCDPS register (Register 21-2) configures the LCD clock source prescaler and the type of waveform; Type-A or Type-B. The LCDSE registers (Register 21-5) configure the functions of the port pins.

The following LCDSE registers are available:

- LCDSE0 SE<7:0>
- LCDSE1 SE<15:8>
- LCDSE2 SE<23:16><sup>(1)</sup>

**Note 1:** PIC16F1934/1937/1939/  
PIC16LF1934/1937/1939 only.

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 SEG<7:0>COM0
- LCDDATA1 SEG<15:8>COM0
- LCDDATA2 SEG<23:16>COM0<sup>(1)</sup>
- LCDDATA3 SEG<7:0>COM1
- LCDDATA4 SEG<15:8>COM1
- LCDDATA5 SEG<23:16>COM1<sup>(1)</sup>
- LCDDATA6 SEG<7:0>COM2
- LCDDATA7 SEG<15:8>COM2
- LCDDATA8 SEG<23:16>COM2<sup>(1)</sup>
- LCDDATA9 SEG<7:0>COM3
- LCDDATA10 SEG<15:8>COM3
- LCDDATA11 SEG<23:16>COM3<sup>(1)</sup>

**Note 1:** PIC16F1934/1937/1939/  
PIC16LF1934/1937/1939 only.

As an example, LCDDATAx is detailed in Register 21-6.

Once the module is configured, the LCDEN bit of the LCDCON register is used to enable or disable the LCD module. The LCD panel can also operate during Sleep by clearing the SLPEN bit of the LCDCON register.

# PIC16F193X/LF193X

## REGISTER 21-1: LCDCON: LIQUID CRYSTAL DISPLAY (LCD) CONTROL REGISTER

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

### Legend:

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

- 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 the WA bit of the LCDPS register = 0 (must be cleared in software)  
 0 = No LCD write error
- bit 4      **Unimplemented:** Read as '0'
- bit 3-2    **CS<1:0>:** Clock Source Select bits  
 00 = Fosc/256  
 01 = T1OSC (Timer1)  
 1x = LFINTOSC (31 kHz)
- bit 1-0    **LMUX<1:0>:** Commons Select bits

LMUX<1:0>	Multiplex	Maximum Number of Pixels		Bias
		PIC16F1933/1936/1938/ PIC16LF1933/1936/1938	PIC16F1934/1937/1939/ PIC16LF1934/1937/1939	
00	Static (COM0)	16	24	Static
01	1/2 (COM<1:0>)	32	48	1/2 or 1/3
10	1/3 (COM<2:0>)	48	72	1/2 or 1/3
11	1/4 (COM<3:0>)	60 <sup>(1)</sup>	96	1/3

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

# PIC16F193X/LF193X

## REGISTER 21-2: LCDPS: LCD PHASE REGISTER

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1
WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0
bit 7							bit 0

### Legend:

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

- bit 7      **WFT:** Waveform Type bit  
 1 = Type-B phase changes on each frame boundary  
 0 = Type-A 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 Selection 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

# PIC16F193X/LF193X

## REGISTER 21-3: LCDREF: LCD REFERENCE VOLTAGE CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—
bit 7							bit 0

### Legend:

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

- bit 7      **LCDIRE:** LCD Internal Reference Enable bit  
1 = Internal LCD Reference is enabled and connected to the Internal Contrast Control circuit  
0 = Internal LCD Reference is disabled
- bit 6      **LCDIRS:** LCD Internal Reference Source bit  
If LCDIRE = 1:  
0 = Internal LCD Contrast Control is powered by VDD  
1 = Internal LCD Contrast Control is powered by a 3.072V output of the FVR.  
If LCDIRE = 0:  
Internal LCD Contrast Control is unconnected. LCD bandgap buffer is disabled.
- bit 5      **LCDIRI:** LCD Internal Reference Ladder Idle Enable bit  
Allows the Internal FVR buffer to shut down when the LCD Reference Ladder is in power mode 'B'  
1 = When the LCD Reference Ladder is in power mode 'B', the LCD Internal FVR buffer is disabled.  
0 = The LCD Internal FVR Buffer ignores the LCD Reference Ladder Power mode.
- bit 4      **Unimplemented:** Read as '0'
- bit 3      **VLCD3PE:** VLCD3 Pin Enable bit  
1 = The VLCD3 pin is connected to the internal bias voltage LCDBIAS3<sup>(1)</sup>  
0 = The VLCD3 pin is not connected
- bit 2      **VLCD2PE:** VLCD2 Pin Enable bit  
1 = The VLCD2 pin is connected to the internal bias voltage LCDBIAS2<sup>(1)</sup>  
0 = The VLCD2 pin is not connected
- bit 1      **VLCD1PE:** VLCD1 Pin Enable bit  
1 = The VLCD1 pin is connected to the internal bias voltage LCDBIAS1<sup>(1)</sup>  
0 = The VLCD1 pin is not connected
- bit 0      **Unimplemented:** Read as '0'

**Note 1:** Normal pin controls of TRISx and ANSELx are unaffected.

# PIC16F193X/LF193X

## REGISTER 21-4: LCDCST: LCD CONTRAST CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	LCDCST2	LCDCST1	LCDCST0
bit 7							bit 0

### Legend:

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

bit 7-3      **Unimplemented:** Read as '0'

bit 2-0      **LCDCST<2:0>:** LCD Contrast Control bits

Selects the resistance of the LCD contrast control resistor ladder

Bit Value = Resistor ladder

000 = Minimum Resistance (Maximum contrast). Resistor ladder is shorted.

001 = Resistor ladder is at 1/7th of maximum resistance

010 = Resistor ladder is at 2/7th of maximum resistance

011 = Resistor ladder is at 3/7th of maximum resistance

100 = Resistor ladder is at 4/7th of maximum resistance

101 = Resistor ladder is at 5/7th of maximum resistance

110 = Resistor ladder is at 6/7th of maximum resistance

111 = Resistor ladder is at maximum resistance (Minimum contrast).



# PIC16F193X/LF193X

## REGISTER 21-5: LCDSEn: LCD SEGMENT ENABLE REGISTERS

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

### Legend:

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

bit 7-0      **SEn:** Segment Enable bits  
 1 = Segment function of the pin is enabled  
 0 = I/O function of the pin is enabled

## REGISTER 21-6: LCDDATAx: LCD DATA REGISTERS

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

### Legend:

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

bit 7-0      **SEGx-COMy:** Pixel On bits  
 1 = Pixel on (dark)  
 0 = Pixel off (clear)

# PIC16F193X/LF193X

## 21.2 LCD Clock Source Selection

The LCD module has 3 possible clock sources:

- $F_{osc}/256$
- T1OSC
- LFINTOSC

The first clock source is the system clock divided by 256 ( $F_{osc}/256$ ). 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 LP<3:0> of the LCDPS register are used to set the LCD frame clock rate.

The second clock source is the T1OSC. 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 bit of the T1CON register should be set.

The third clock source is the 31 kHz 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.

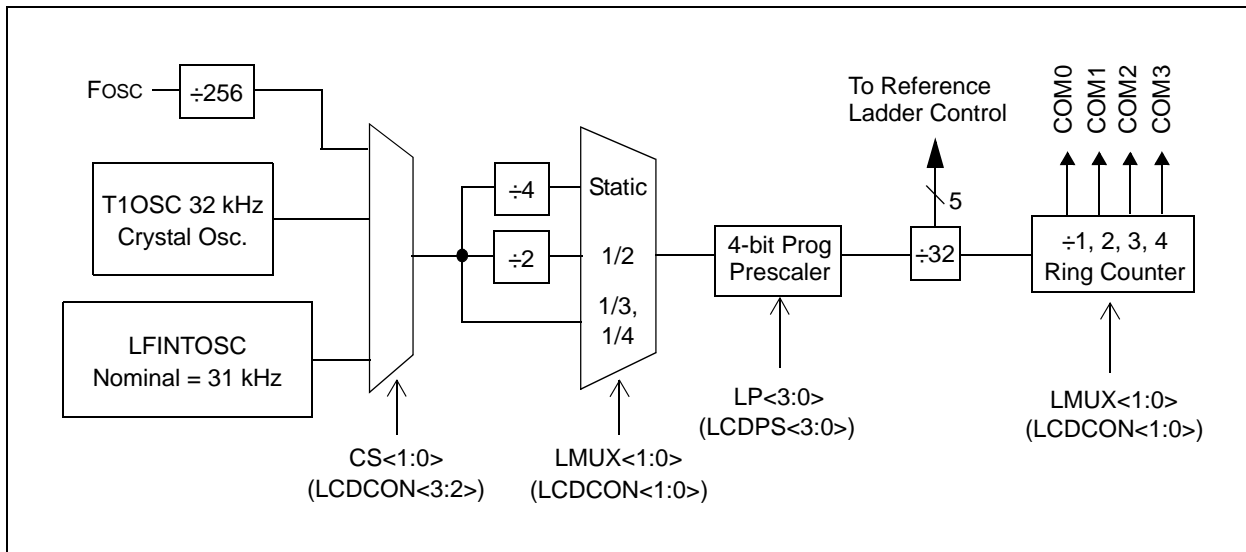
Using bits CS<1:0> of the LCDCON register can select any of these clock sources.

### 21.2.1 LCD PRESCALER

A 4-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 of the LCDPS register, which determine the prescaler assignment and prescale ratio.

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

FIGURE 21-2: LCD CLOCK GENERATION



# PIC16F193X/LF193X

## 21.3 LCD Bias Voltage Generation

The LCD module can be configured for one of three bias types:

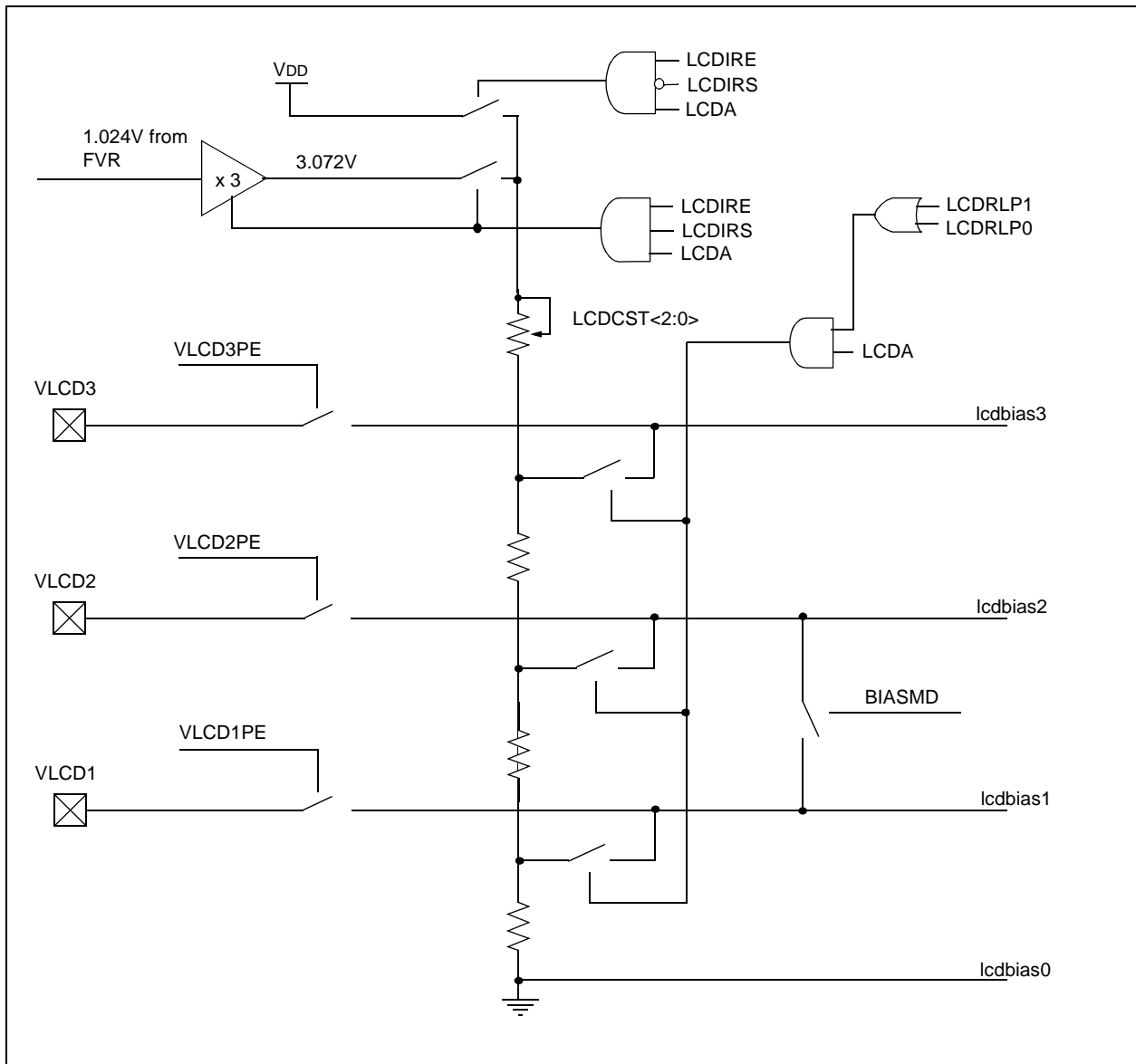
- Static Bias (2 voltage levels: VSS and VLCD)
- 1/2 Bias (3 voltage levels: VSS, 1/2 VLCD and VLCD)
- 1/3 Bias (4 voltage levels: VSS, 1/3 VLCD, 2/3 VLCD and VLCD)

TABLE 21-2: LCD BIAS VOLTAGES

	Static Bias	1/2 Bias	1/3 Bias
LCD Bias 0	VSS	VSS	VSS
LCD Bias 1	—	1/2 VDD	1/3 VDD
LCD Bias 2	—	1/2 VDD	2/3 VDD
LCD Bias 3	VLCD3	VLCD3	VLCD3

So that the user is not forced to place external components and use up to three pins for bias voltage generation, internal contrast control and an internal reference ladder are provided internally to the PIC16F193X/LF193X. Both of these features may be used in conjunction with the external VLCD<3:1> pins, to provide maximum flexibility. Refer to Figure 21-3.

FIGURE 21-3: LCD BIAS VOLTAGE GENERATION BLOCK DIAGRAM



# PIC16F193X/LF193X

---

## 21.4 LCD Bias Internal Reference Ladder

The internal reference ladder can be used to divide the LCD bias voltage two or three equally spaced voltages that will be supplied to the LCD segment pins. To create this, the reference ladder consists of three matched resistors. Refer to Figure 21-3.

### 21.4.1 BIAS MODE INTERACTION

When in 1/2 Bias mode (BIASMD = 1), then the middle resistor of the ladder is shorted out so that only two voltages are generated. The current consumption of the ladder is higher in this mode, with the one resistor removed.

**TABLE 21-3: LCD INTERNAL LADDER POWER MODES (1/3 BIAS)**

Power Mode	Nominal Resistance of Entire Ladder	Nominal I <sub>DD</sub>
Low	3 Mohm	1 $\mu$ A
Medium	300 kohm	10 $\mu$ A
High	30 kohm	100 $\mu$ A

### 21.4.2 POWER MODES

The internal reference ladder may be operated in one of three power modes. This allows the user to trade off LCD contrast for power in the specific application. The larger the LCD glass, the more capacitance is present on a physical LCD segment, requiring more current to maintain the same contrast level.

Three different power modes are available, LP, MP and HP. The internal reference ladder can also be turned off for applications that wish to provide an external ladder or to minimize power consumption. Disabling the internal reference ladder results in all of the ladders being disconnected, allowing external voltages to be supplied.

Whenever the LCD module is inactive (LCDA = 0), the internal reference ladder will be turned off.

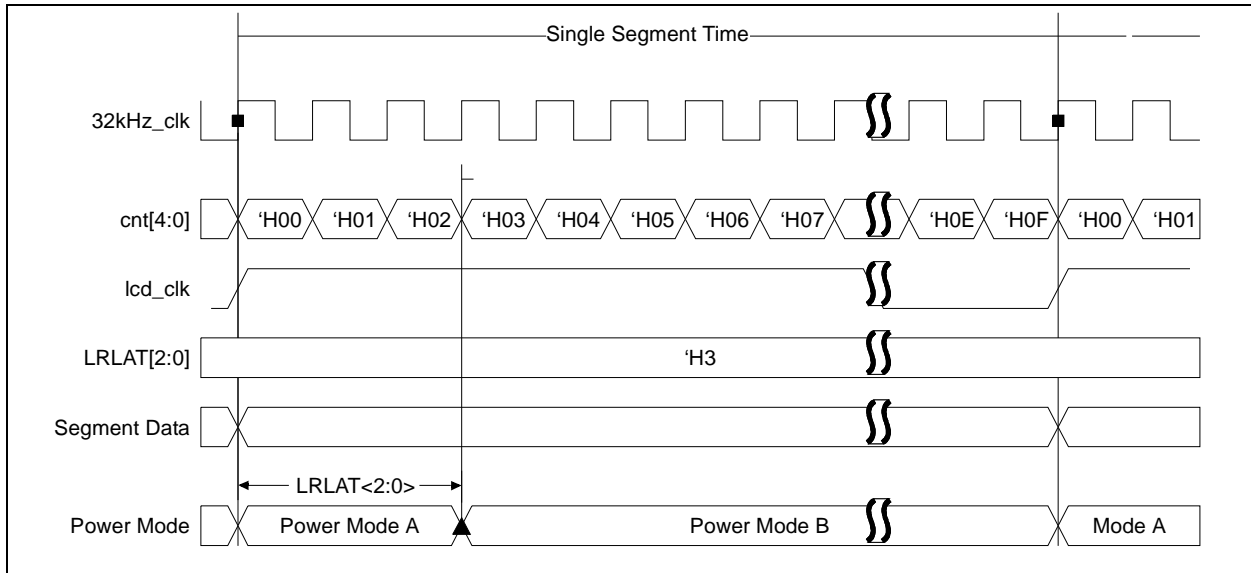
## 21.4.3 AUTOMATIC POWER MODE SWITCHING

As an LCD segment is electrically only a capacitor, current is drawn only during the interval where the voltage is switching. To minimize total device current, the LCD internal reference ladder can be operated in a different power mode for the transition portion of the duration. This is controlled by the LCDRL Register (Register 21-7).

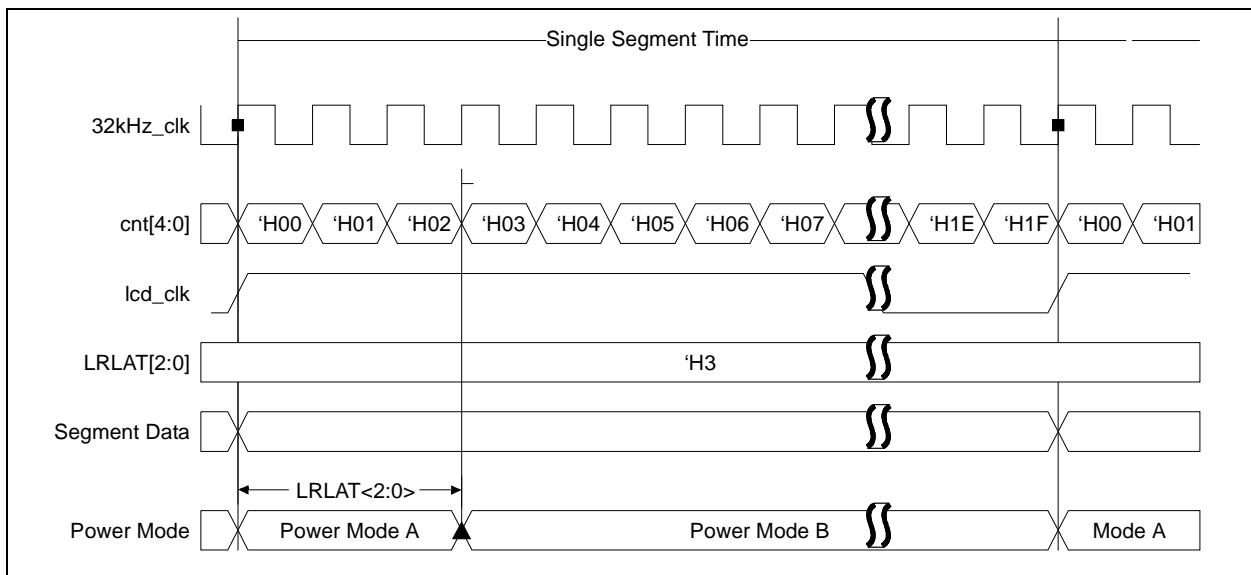
The LCDLAD register allows switching between two power modes, designated 'A' and 'B'. 'A' Power mode is active for a programmable time, beginning at the time when the LCD segments transition. 'B' Power mode is the remaining time before the segments or commons change again. The LRLAT<2:0> bits select how long, if any, that the 'A' Power mode is active. Refer to Figure 21-4.

To implement this, the 5-bit prescaler used to divide the 32 kHz clock down to the LCD controller's 1 kHz base rate is used to select the power mode.

**FIGURE 21-4: LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE A**



**FIGURE 21-5: LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE B**



# PIC16F193X/LF193X

## REGISTER 21-7: LCDRL: LCD REFERENCE LADDER CONTROL REGISTERS

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0
bit 7							bit 0

### Legend:

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

- bit 7-6     **LRLAP<1:0>**: LCD Reference Ladder A Time Power Control bits  
 During Time interval A (Refer to Figure 21-4):  
 00 = Internal LCD Reference Ladder is powered down and unconnected  
 01 = Internal LCD Reference Ladder is powered in low-power mode  
 10 = Internal LCD Reference Ladder is powered in medium-power mode  
 11 = Internal LCD Reference Ladder is powered in high-power mode
- bit 5-4     **LRLBP<1:0>**: LCD Reference Ladder B Time Power Control bits  
 During Time interval B (Refer to Figure 21-4):  
 00 = Internal LCD Reference Ladder is powered down and unconnected  
 01 = Internal LCD Reference Ladder is powered in low-power mode  
 10 = Internal LCD Reference Ladder is powered in medium-power mode  
 11 = Internal LCD Reference Ladder is powered in high-power mode
- bit 3       **Unimplemented**: Read as '0'
- bit 2-0     **LRLAT<2:0>**: LCD Reference Ladder A Time interval control bits  
 Sets the number of 32 kHz clocks that the A Time interval power mode is active

For type A waveforms (WFT = 0):

- 000 = Internal LCD Reference Ladder is always in 'B' power mode
- 001 = Internal LCD Reference Ladder is in 'A' power mode for 1 clock and 'B' power mode for 15 clocks
- 010 = Internal LCD Reference Ladder is in 'A' power mode for 2 clocks and 'B' power mode for 14 clocks
- 011 = Internal LCD Reference Ladder is in 'A' power mode for 3 clocks and 'B' power mode for 13 clocks
- 100 = Internal LCD Reference Ladder is in 'A' power mode for 4 clocks and 'B' power mode for 12 clocks
- 101 = Internal LCD Reference Ladder is in 'A' power mode for 5 clocks and 'B' power mode for 11 clocks
- 110 = Internal LCD Reference Ladder is in 'A' power mode for 6 clocks and 'B' power mode for 10 clocks
- 111 = Internal LCD Reference Ladder is in 'A' power mode for 7 clocks and 'B' power mode for 9 clocks

For type B waveforms (WFT = 1):

- 000 = Internal LCD Reference Ladder is always in 'B' power mode.
- 001 = Internal LCD Reference Ladder is in 'A' power mode for 1 clock and 'B' power mode for 31 clocks
- 010 = Internal LCD Reference Ladder is in 'A' power mode for 2 clocks and 'B' power mode for 30 clocks
- 011 = Internal LCD Reference Ladder is in 'A' power mode for 3 clocks and 'B' power mode for 29 clocks
- 100 = Internal LCD Reference Ladder is in 'A' power mode for 4 clocks and 'B' power mode for 28 clocks
- 101 = Internal LCD Reference Ladder is in 'A' power mode for 5 clocks and 'B' power mode for 27 clocks
- 110 = Internal LCD Reference Ladder is in 'A' power mode for 6 clocks and 'B' power mode for 26 clocks
- 111 = Internal LCD Reference Ladder is in 'A' power mode for 7 clocks and 'B' power mode for 25 clocks

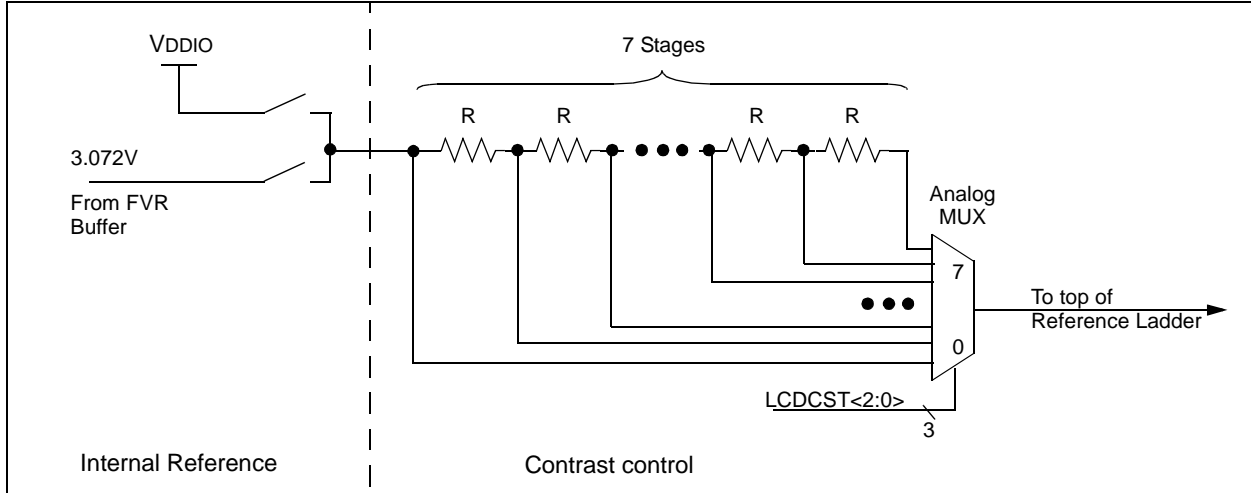
## 21.4.4 CONTRAST CONTROL

The LCD contrast control circuit consists of a seven-tap resistor ladder, controlled by the LCDCST bits. Refer to Figure 21-6.

The contrast control circuit is used to decrease the output voltage of the signal source by a total of approximately 10%, when  $LCDCST = 111$ .

Whenever the LCD module is inactive ( $LCDA = 0$ ), the contrast control ladder will be turned off (open).

**FIGURE 21-6: INTERNAL REFERENCE AND CONTRAST CONTROL BLOCK DIAGRAM**



## 21.4.5 INTERNAL REFERENCE

Under firmware control, an internal reference for the LCD bias voltages can be enabled. When enabled, the source of this voltage can be either  $VDDIO$  or a voltage 3 times the main fixed voltage reference (3.072V). When no internal reference is selected, the LCD contrast control circuit is disabled and LCD bias must be provided externally.

Whenever the LCD module is inactive ( $LCDA = 0$ ), the internal reference will be turned off.

When the internal reference is enabled and the Fixed Voltage Reference is selected, the LCDIRI bit can be used to minimize power consumption by tying into the LCD reference ladder automatic power mode switching. When  $LCDIRI = 1$ , the power mode that the internal LCD reference ladder enables the buffer when in power mode 'A' and disables it when in power mode 'B'.

**Note:** The LCD module automatically turns on the fixed voltage reference when needed.

## 21.4.6 VLCD<3:1> PINS

The  $VLCD<3:1>$  pins provide the ability for an external LCD bias network to be used instead of the internal ladder. Use of the  $VLCD<3:1>$  pins does not prevent use of the internal ladder. Each VLCD pin has an independent control in the LCDREF register (Register 21-3), allowing access to any or all of the LCD Bias signals. This architecture allows for maximum flexibility in different applications.

For example, the  $VLCD<3:1>$  pins may be used to add capacitors to the internal reference ladder, increasing the drive capacity.

For applications where the internal contrast control is insufficient, the firmware can choose to only enable the VLCD3 pin, allowing an external contrast control circuit to use the internal reference divider.

# PIC16F193X/LF193X

## 21.5 LCD Multiplex Types

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

- Static (only COM0 is used)
- 1/2 multiplex (COM<1:0> are used)
- 1/3 multiplex (COM<2:0> are used)
- 1/4 multiplex (COM<3:0> are used)

The LMUX<1:0> bit setting of the LCDCON register decides which of the LCD common pins are used (see Table 21-4 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.

**TABLE 21-4: COMMON PIN USAGE**

Multiplex	LMUX <1:0>	COM3	COM2	COM1
Static	00	Unused	Unused	Unused
1/2	01	Unused	Unused	Active
1/3	10	Unused	Active	Active
1/4	11	Active	Active	Active

## 21.6 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 normal I/O, not LCD pins.

## 21.7 Pixel Control

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

Register 21-6 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.

## 21.8 LCD Frame Frequency

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

**TABLE 21-5: FRAME FREQUENCY FORMULAS**

Multiplex	Frame Frequency =
Static	$\text{Clock source}/(4 \times 1 \times (\text{LP}<3:0> + 1))$
1/2	$\text{Clock source}/(2 \times 2 \times (\text{LP}<3:0> + 1))$
1/3	$\text{Clock source}/(1 \times 3 \times (\text{LP}<3:0> + 1))$
1/4	$\text{Clock source}/(1 \times 4 \times (\text{LP}<3:0> + 1))$

**Note:** Clock source is Fosc/256, T1OSC or LFINTOSC.

**TABLE 21-6: APPROXIMATE FRAME FREQUENCY (IN Hz) USING Fosc @ 8 MHz, TIMER1 @ 32.768 kHz OR LFINTOSC**

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



# PIC16F193X/LF193X

**TABLE 21-7: LCD SEGMENT MAPPING WORKSHEET**

LCD Function	COM0		COM1		COM2		COM3	
	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment	LCDDATAx Address	LCD Segment
SEG0	LCDDATA0, 0		LCDDATA3, 0		LCDDATA6, 0		LCDDATA9, 0	
SEG1	LCDDATA0, 1		LCDDATA3, 1		LCDDATA6, 1		LCDDATA9, 1	
SEG2	LCDDATA0, 2		LCDDATA3, 2		LCDDATA6, 2		LCDDATA9, 2	
SEG3	LCDDATA0, 3		LCDDATA3, 3		LCDDATA6, 3		LCDDATA9, 3	
SEG4	LCDDATA0, 4		LCDDATA3, 4		LCDDATA6, 4		LCDDATA9, 4	
SEG5	LCDDATA0, 5		LCDDATA3, 5		LCDDATA6, 5		LCDDATA9, 5	
SEG6	LCDDATA0, 6		LCDDATA3, 6		LCDDATA6, 6		LCDDATA9, 6	
SEG7	LCDDATA0, 7		LCDDATA3, 7		LCDDATA6, 7		LCDDATA9, 7	
SEG8	LCDDATA1, 0		LCDDATA4, 0		LCDDATA7, 0		LCDDATA10, 0	
SEG9	LCDDATA1, 1		LCDDATA4, 1		LCDDATA7, 1		LCDDATA10, 1	
SEG10	LCDDATA1, 2		LCDDATA4, 2		LCDDATA7, 2		LCDDATA10, 2	
SEG11	LCDDATA1, 3		LCDDATA4, 3		LCDDATA7, 3		LCDDATA10, 3	
SEG12	LCDDATA1, 4		LCDDATA4, 4		LCDDATA7, 4		LCDDATA10, 4	
SEG13	LCDDATA1, 5		LCDDATA4, 5		LCDDATA7, 5		LCDDATA10, 5	
SEG14	LCDDATA1, 6		LCDDATA4, 6		LCDDATA7, 6		LCDDATA10, 6	
SEG15	LCDDATA1, 7		LCDDATA4, 7		LCDDATA7, 7		LCDDATA10, 7	
SEG16	LCDDATA2, 0		LCDDATA5, 0		LCDDATA8, 0		LCDDATA11, 0	
SEG17	LCDDATA2, 1		LCDDATA5, 1		LCDDATA8, 1		LCDDATA11, 1	
SEG18	LCDDATA2, 2		LCDDATA5, 2		LCDDATA8, 2		LCDDATA11, 2	
SEG19	LCDDATA2, 3		LCDDATA5, 3		LCDDATA8, 3		LCDDATA11, 3	
SEG20	LCDDATA2, 4		LCDDATA5, 4		LCDDATA8, 4		LCDDATA11, 4	
SEG21	LCDDATA2, 5		LCDDATA5, 5		LCDDATA8, 5		LCDDATA11, 5	
SEG22	LCDDATA2, 6		LCDDATA5, 6		LCDDATA8, 6		LCDDATA11, 6	
SEG23	LCDDATA2, 7		LCDDATA5, 7		LCDDATA8, 7		LCDDATA11, 7	

# PIC16F193X/LF193X

## 21.9 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.

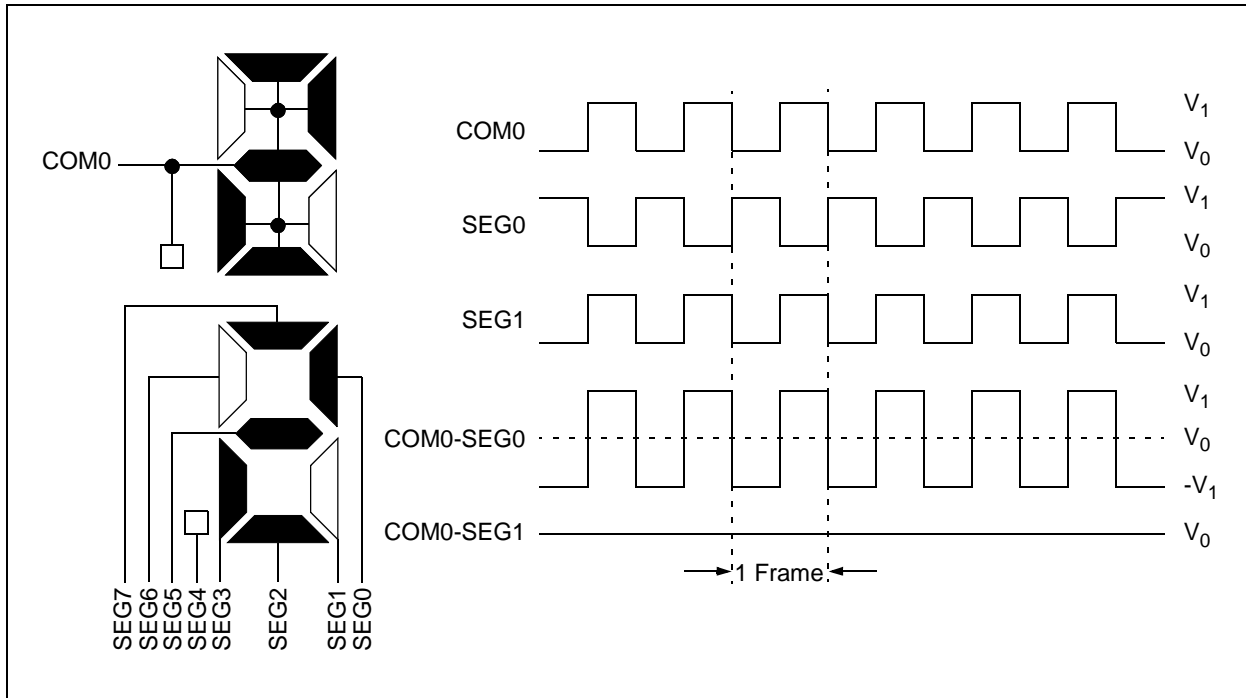
The LCDs can be driven by two types of waveform: Type-A and Type-B. In Type-A waveform, the phase changes within each common type, whereas in Type-B waveform, the phase changes on each frame boundary. Thus, Type-A waveform maintains 0 VDC over a single frame, whereas Type-B waveform takes two frames.

**Note 1:** If Sleep has to be executed with LCD Sleep disabled (LCDCON<SLPEN> is '1'), then care must be taken to execute Sleep only when VDC on all the pixels is '0'.

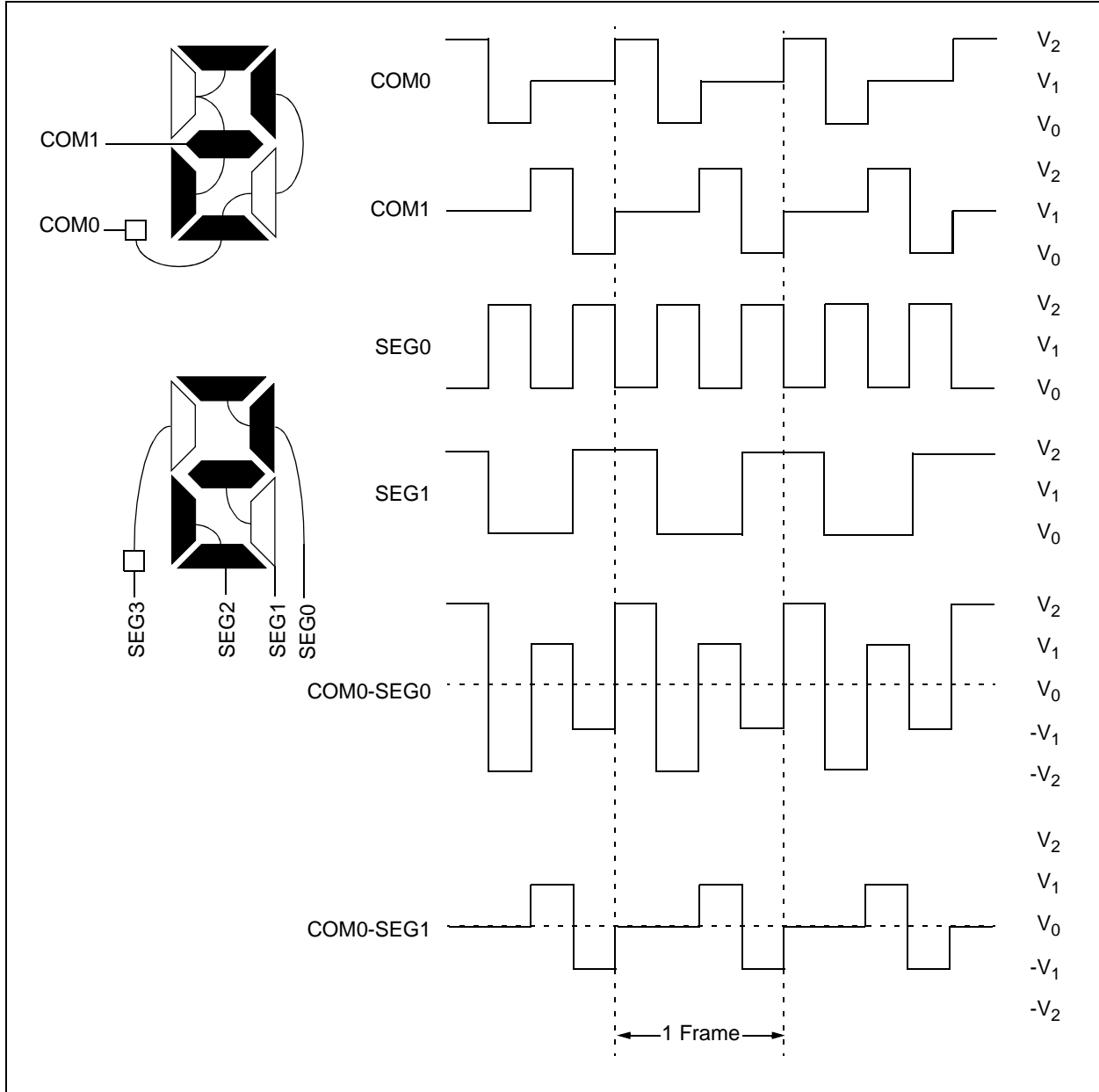
**2:** When the LCD clock source is Fosc/256, if Sleep is executed, irrespective of the LCDCON<SLPEN> setting, the LCD immediately goes into Sleep. Thus, take care to see that VDC on all pixels is '0' when Sleep is executed.

Figure 21-7 through Figure 21-17 provide waveforms for static, half-multiplex, 1/3-multiplex and 1/4-multiplex drives for Type-A and Type-B waveforms.

**FIGURE 21-7: TYPE-A/TYPE-B WAVEFORMS IN STATIC DRIVE**

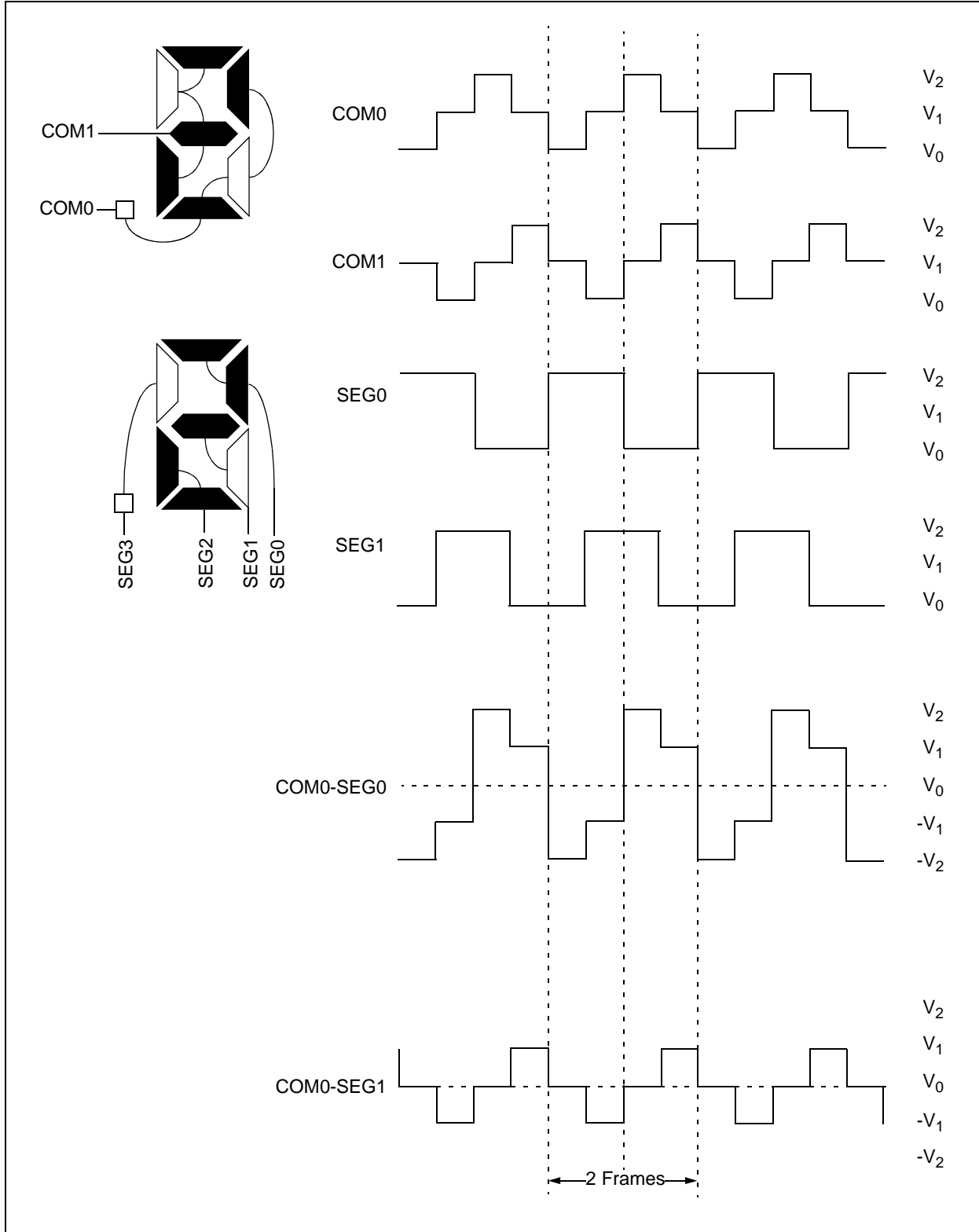


**FIGURE 21-8: TYPE-A WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE**

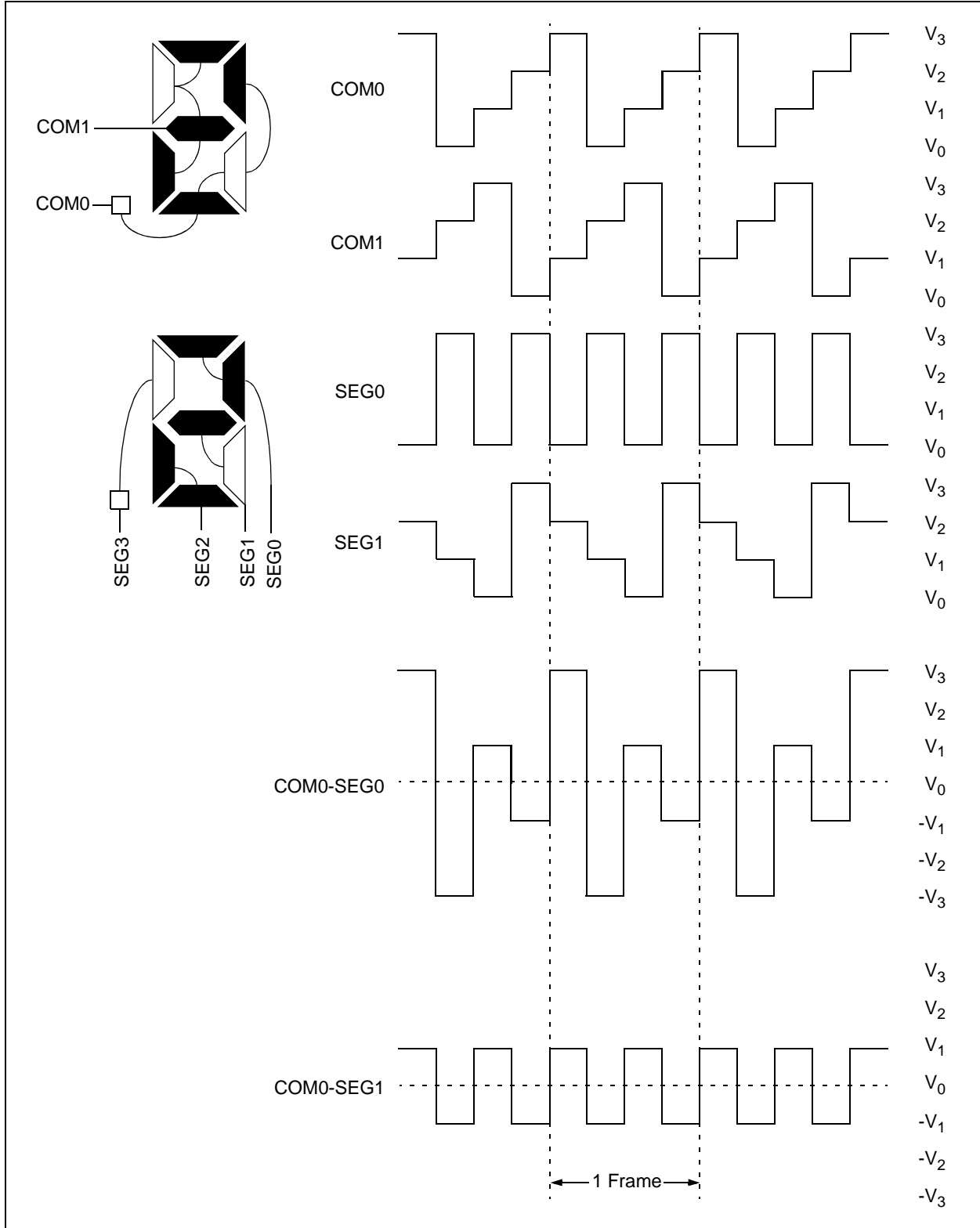


# PIC16F193X/LF193X

FIGURE 21-9: TYPE-B WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE

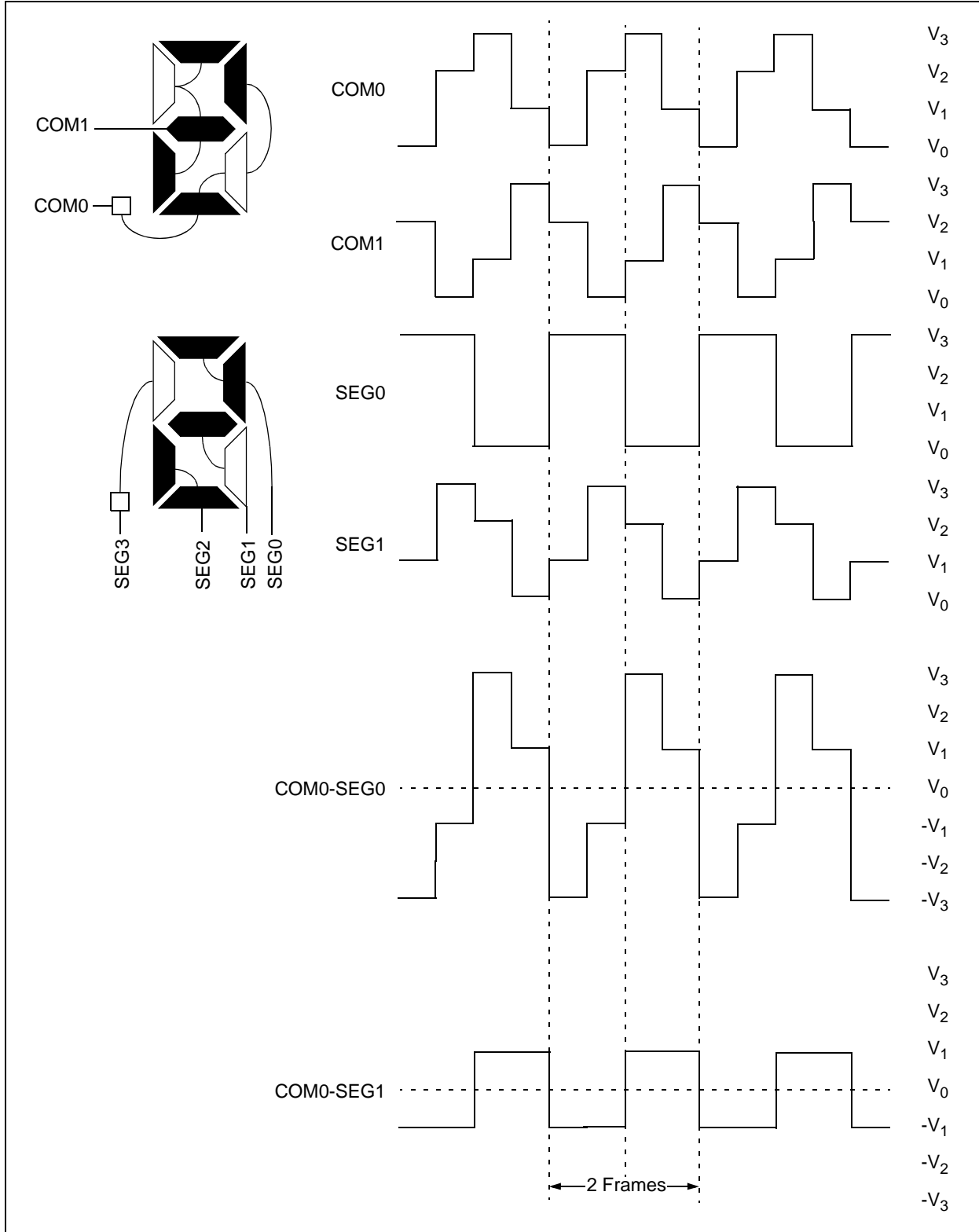


**FIGURE 21-10: TYPE-A WAVEFORMS IN 1/2 MUX, 1/3 BIAS DRIVE**

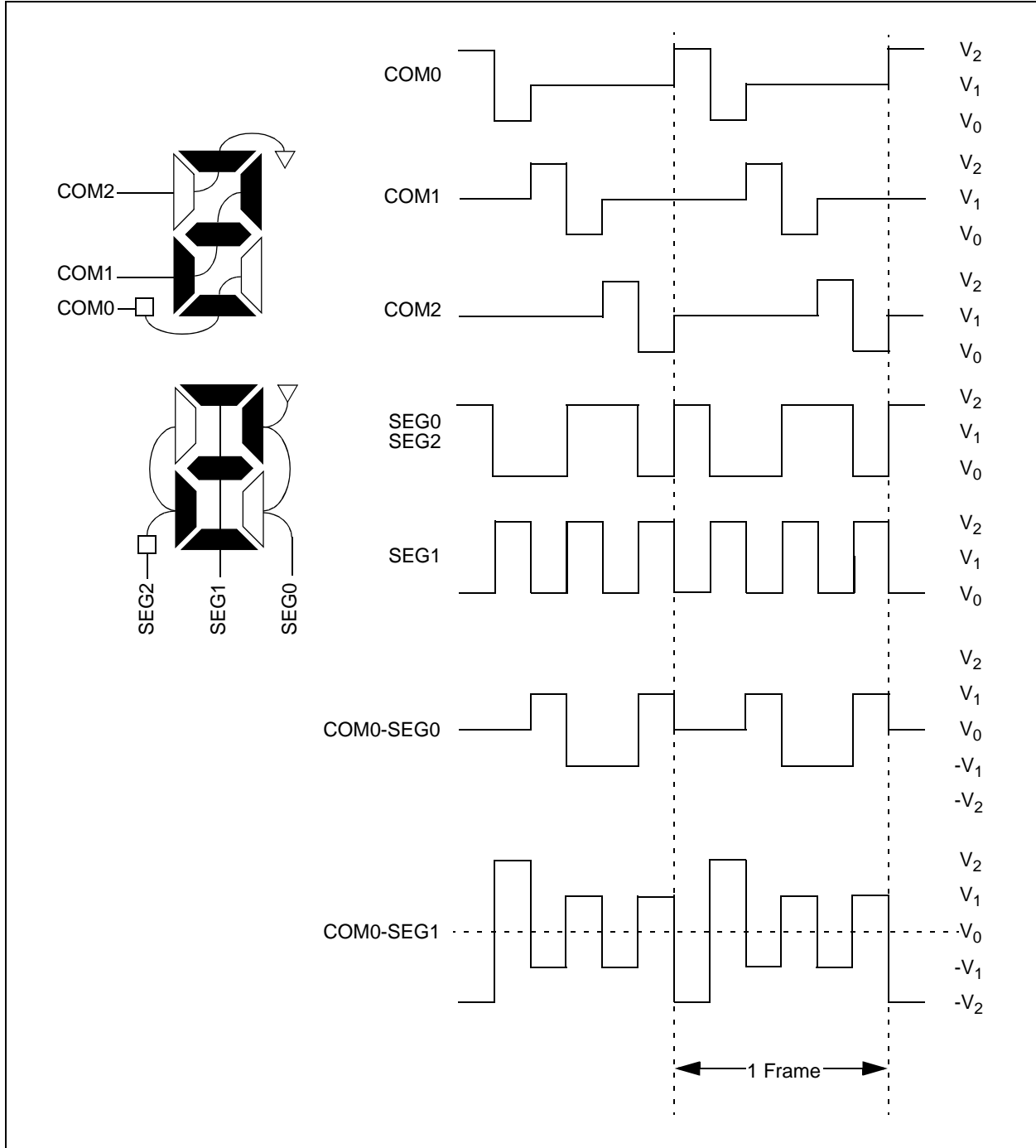


# PIC16F193X/LF193X

FIGURE 21-11: TYPE-B WAVEFORMS IN 1/2 MUX, 1/3 BIAS DRIVE

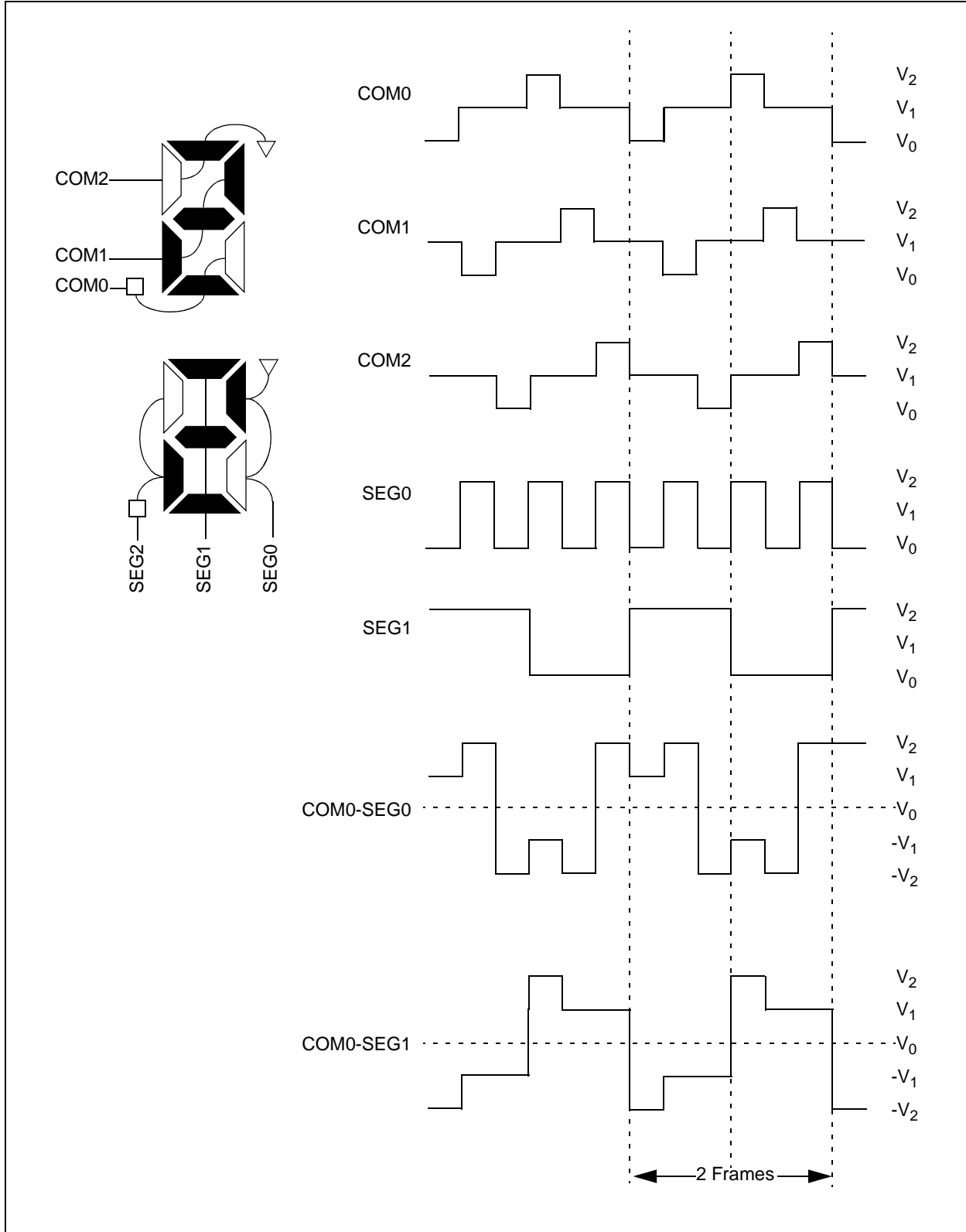


**FIGURE 21-12: TYPE-A WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE**



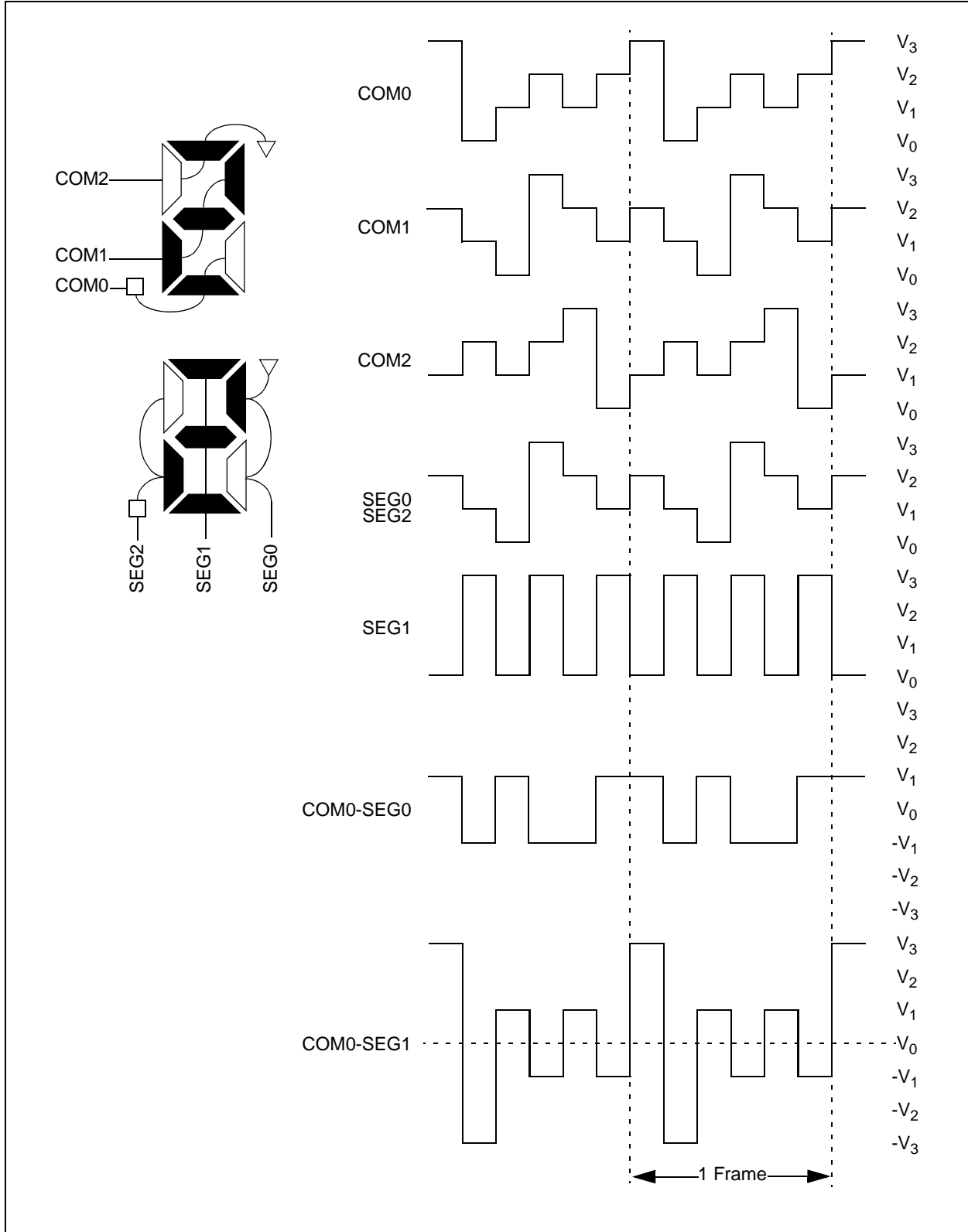
# PIC16F193X/LF193X

FIGURE 21-13: TYPE-B WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE



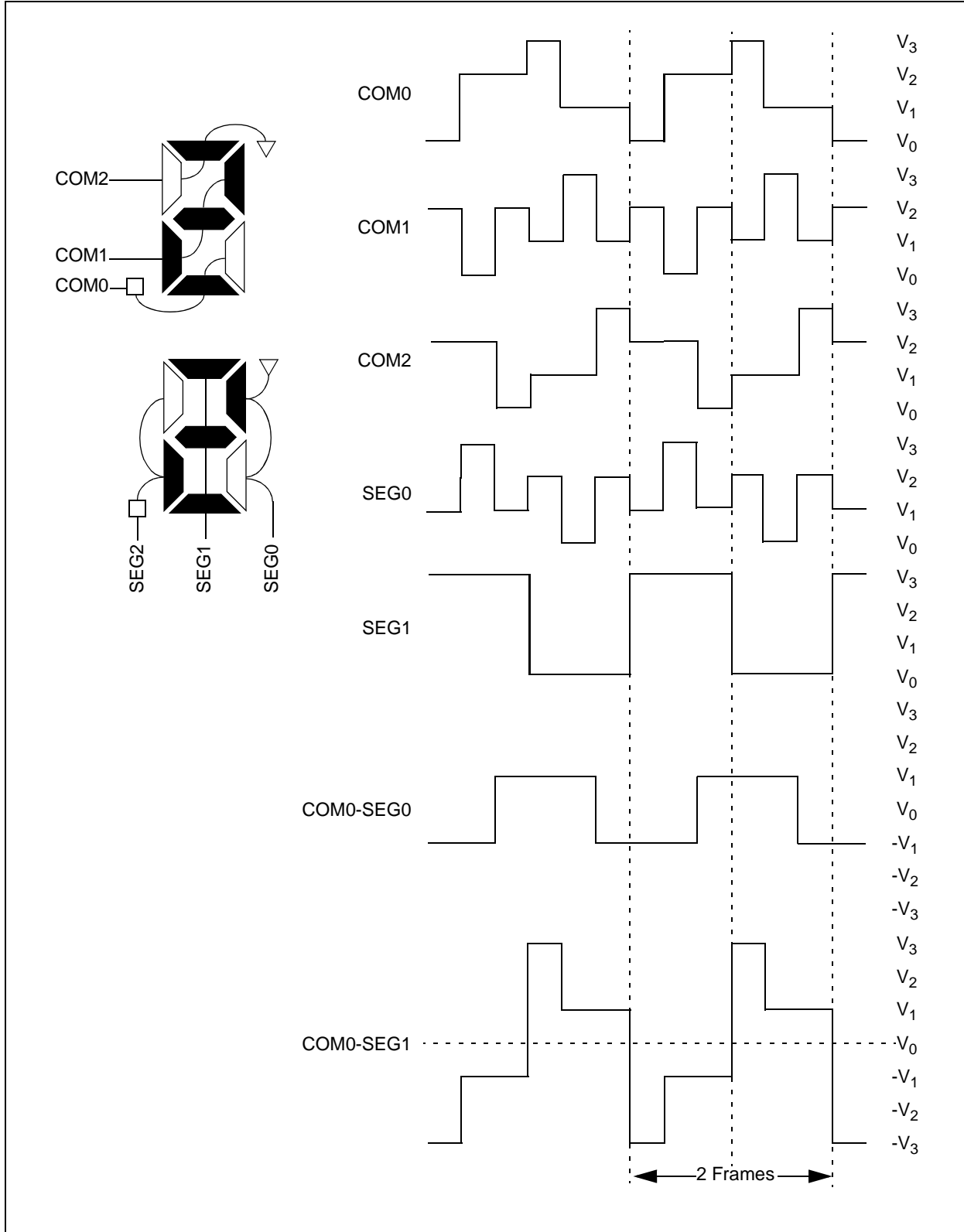


**FIGURE 21-14: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE**

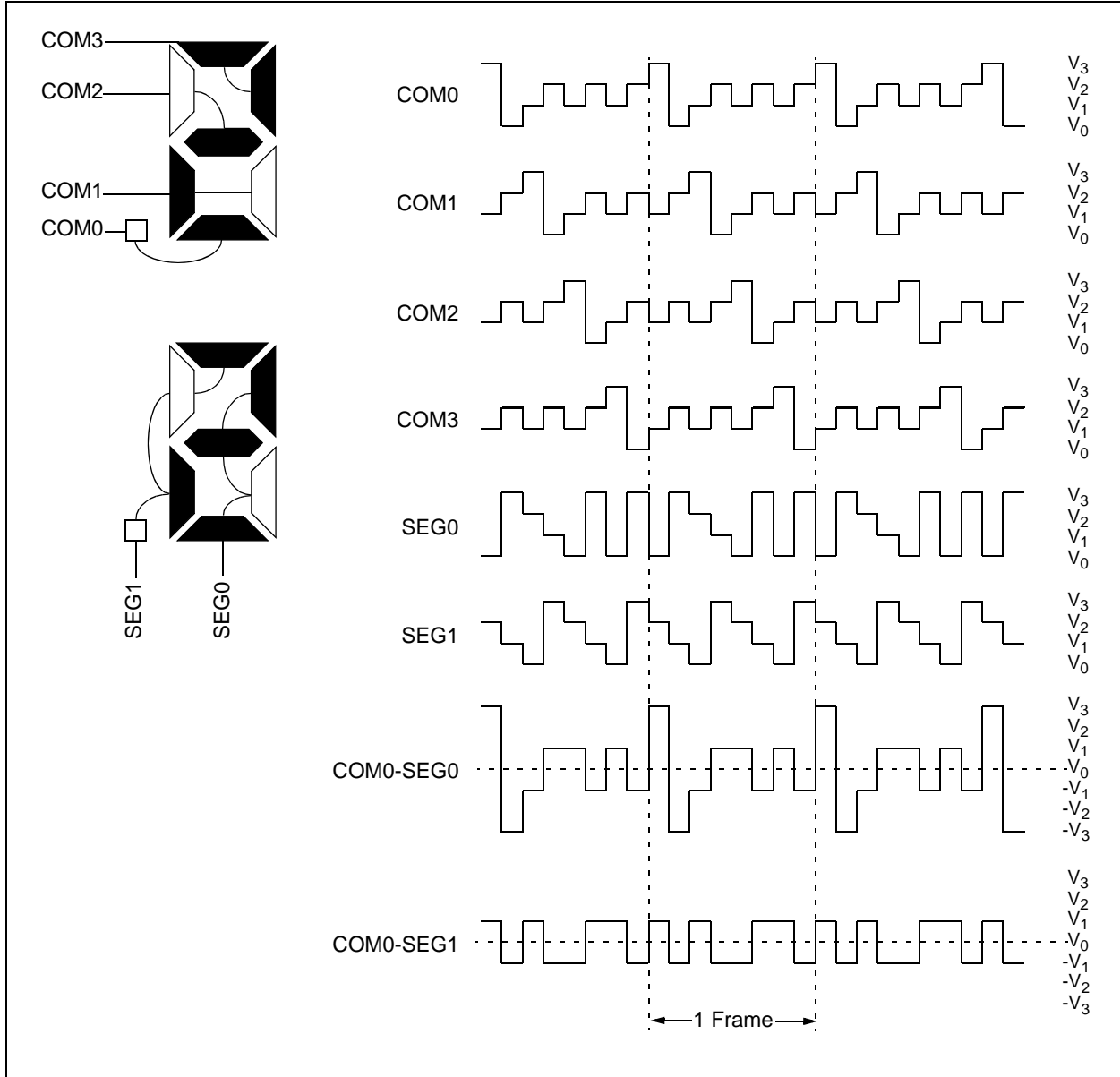


# PIC16F193X/LF193X

FIGURE 21-15: TYPE-B WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE

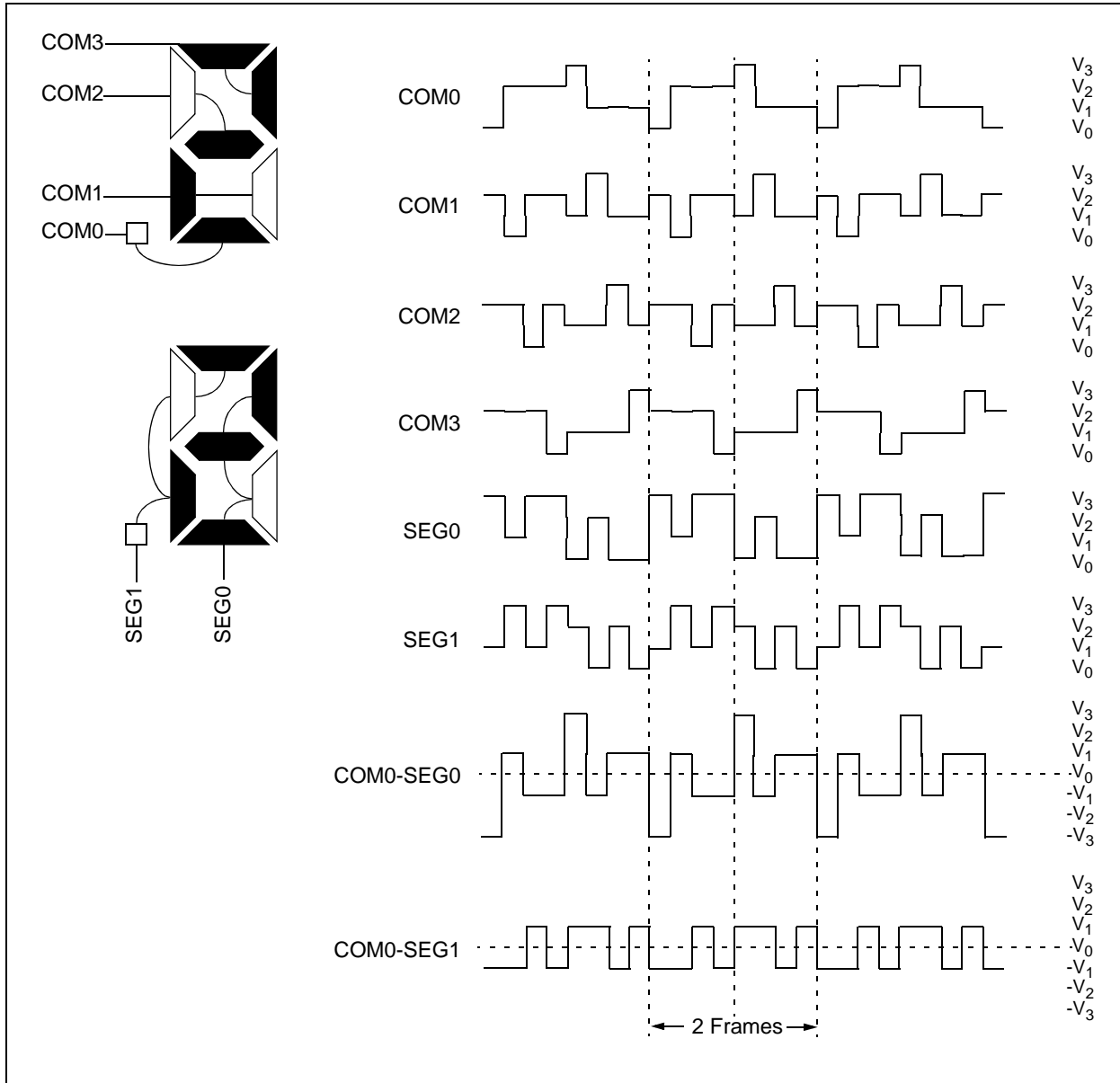


**FIGURE 21-16: TYPE-A WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE**



# PIC16F193X/LF193X

FIGURE 21-17: TYPE-B WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE



## 21.10 LCD Interrupts

The LCD module provides an interrupt in two cases. An interrupt when the LCD controller goes from active to inactive controller. An interrupt also provides unframe boundaries for Type B waveform. The LCD timing generation provides an interrupt that defines the LCD frame timing.

### 21.10.1 LCD INTERRUPT ON MODULE SHUTDOWN

An LCD interrupt is generated when the module completes shutting down (LCDA goes from '1' to '0').

### 21.10.2 LCD FRAME INTERRUPTS

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 21-18. 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.

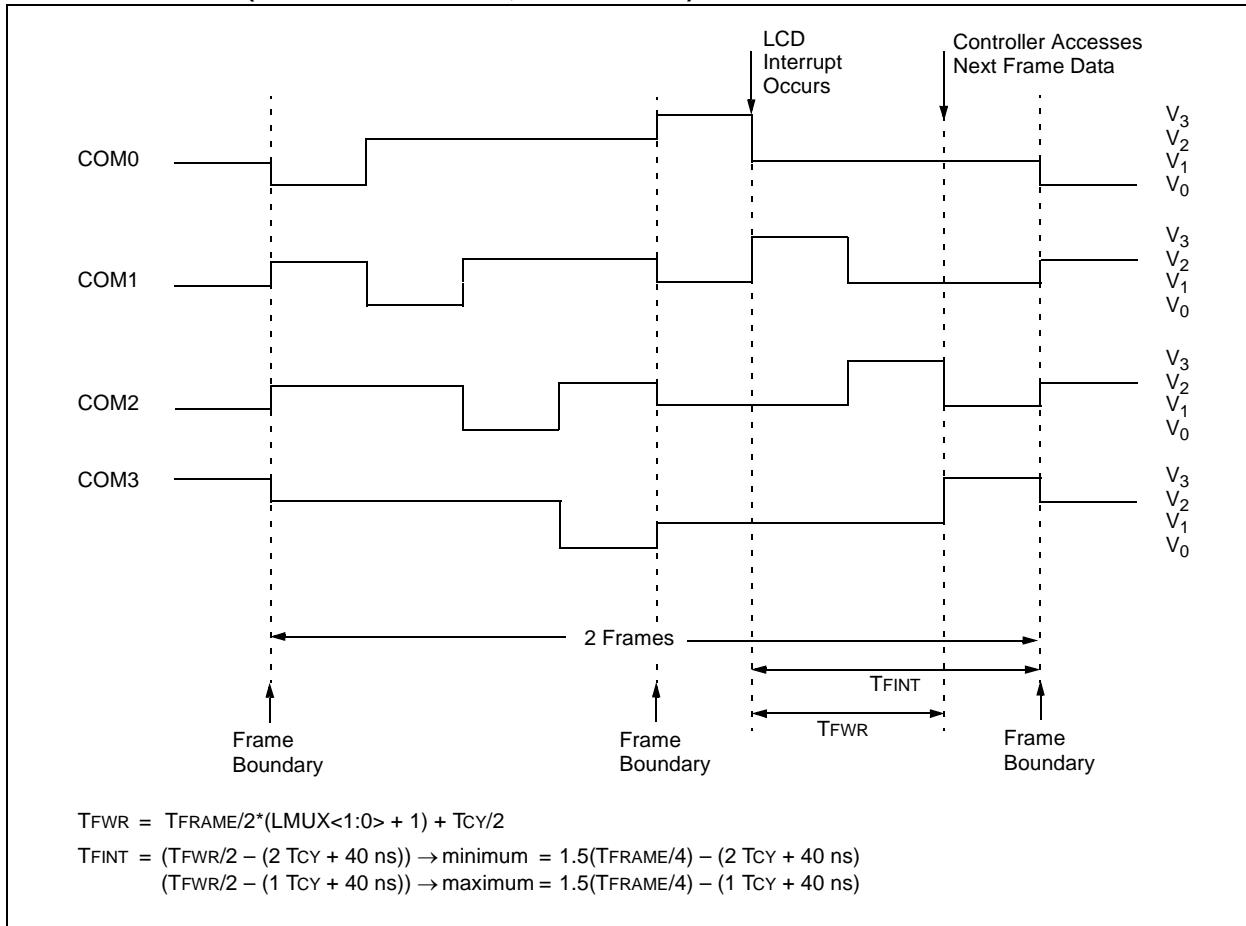
When the LCD driver is running with Type-B waveforms and the LMUX<1:0> bits are not equal to '00' (static drive), there are some additional issues that must be addressed. Since the DC voltage on the pixel takes two frames to maintain zero volts, the pixel data must not change between subsequent frames. 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 bit of the LCDCON register is set and the write does not occur.

<p><b>Note:</b> The LCD frame interrupt is not generated when the Type-A waveform is selected and when the Type-B with no multiplex (static) is selected.</p>
---

# PIC16F193X/LF193X

**FIGURE 21-18: WAVEFORMS AND INTERRUPT TIMING IN QUARTER-DUTY CYCLE DRIVE (EXAMPLE – TYPE-B, NON-STATIC)**



## 21.11 Operation During Sleep

The LCD module can operate during Sleep. The selection is controlled by bit SLPEN of the LCDCON register. 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 21-19 shows this operation.

The LCD module can be configured to operate during Sleep. The selection is controlled by bit SLPEN of the LCDCON register. Clearing SLPEN and correctly configuring the LCD module clock will allow the LCD module to operate during Sleep. Setting SLPEN and correctly executing the LCD module shutdown will disable the LCD module during Sleep and save power.

If a SLEEP instruction is executed and SLPEN = 1, the LCD module will immediately cease all functions, drive the outputs to Vss and go into a very low-current mode. The SLEEP instruction should only be executed after the LCD module has been disabled and the current cycle completed, thus ensuring that there are no DC voltages on the glass. To disable the LCD module, clear the LCDEN bit. The LCD module will complete the disabling process after the current frame, clear the LCDA bit and optionally cause an interrupt.

The steps required to properly enter Sleep with the LCD disabled are:

- Clear LCDEN
- Wait for LCDA = 0 either by polling or by interrupt
- Execute SLEEP

If SLPEN = 0 and SLEEP is executed while the LCD module clock source is FOSC/4, then the LCD module will halt with the pin driving the last LCD voltage pattern. Prolonged exposure to a fixed LCD voltage pattern will cause damage to the LCD glass. To prevent LCD glass damage, either perform the proper LCD module shutdown prior to Sleep, or change the LCD module clock to allow the LCD module to continue operation during Sleep.

If a SLEEP instruction is executed and SLPEN = 0 and the LCD module clock is either T1OSC or LFINTOSC, the module will continue to display the current contents of the LCDDATA registers. While in Sleep, the LCD data cannot be changed. If the LCDIE bit is set, the device will wake from Sleep on the next LCD frame boundary. The LCD module current consumption will not decrease in this mode; however, the overall device power consumption will be lower due to the shutdown of the CPU and other peripherals.

Table 21-8 shows the status of the LCD module during a Sleep while using each of the three available clock sources.

**Note:** When the LCDEN bit is cleared, the LCD module will be disabled at the completion of frame. At this time, the port pins will revert to digital functionality. To minimize power consumption due to floating digital inputs, the LCD pins should be driven low using the PORT and TRIS registers.

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 shutdown of the core and other peripheral functions.

Table 21-8 shows the status of the LCD module during Sleep while using each of the three available clock sources:

**TABLE 21-8: LCD MODULE STATUS DURING SLEEP**

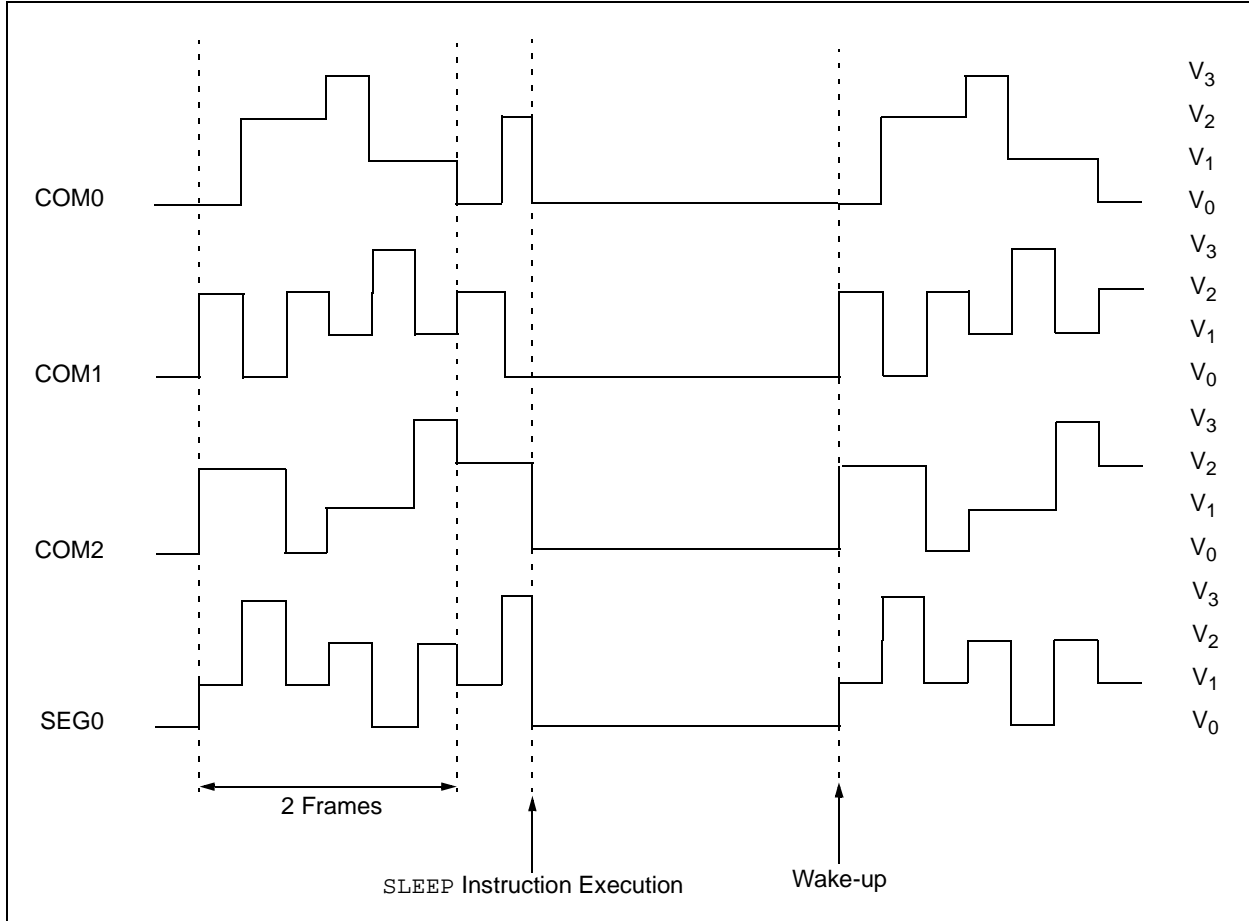
Clock Source	SLPEN	Operational During Sleep
T1OSC	0	Yes
	1	No
LFINTOSC	0	Yes
	1	No
FOSC/4	0	No
	1	No

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

If LCD interrupts are being generated (Type-B waveform with a multiplex mode not static) and LCDIE = 1, the device will awaken from Sleep on the next frame boundary.

# PIC16F193X/LF193X

FIGURE 21-19: SLEEP ENTRY/EXIT WHEN SLPEN = 1





## 21.12 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> of the LCDPS register.
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 (LCDDATA23 on PIC16F1938).
5. Clear LCD Interrupt Flag, LCDIF bit of the PIR2 register and if desired, enable the interrupt by setting bit LCDIE of the PIE2 register.
6. Configure bias voltages by setting the LCDRL, LCDREF and the associated ANSELx registers as needed.
7. Enable the LCD module by setting bit LCDEN of the LCDCON register.

## 21.13 Disabling the LCD Module

To disable the LCD module, write all '0's to the LCDCON register.

## 21.14 LCD Current Consumption

When using the LCD module the current consumption consists of the following three factors:

- Oscillator Selection
- LCD Bias Source
- Capacitance of the LCD segments

The current consumption of just the LCD module can be considered negligible compared to these other factors.

### 21.14.1 OSCILLATOR SELECTION

The current consumed by the clock source selected must be considered when using the LCD module. See **Section 28.0 "Electrical Specifications"** for oscillator current consumption information.

### 21.14.2 LCD BIAS SOURCE

The LCD bias source, internal or external, can contribute significantly to the current consumption. Use the highest possible resistor values while maintaining contrast to minimize current.

### 21.14.3 CAPACITANCE OF THE LCD SEGMENTS

The LCD segments which can be modeled as capacitors which must be both charged and discharged every frame. The size of the LCD segment and its technology determines the segment's capacitance.

# PIC16F193X/LF193X

**TABLE 21-9: REGISTERS ASSOCIATED WITH LCD OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFE	TMR0IF	INTF	IOCF	73
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	243
LCDCST	—	—	—	—	—	LCDCST2	LCDCST1	LCDCST0	246
LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	247
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	247
LCDDATA2	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	247
LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	247
LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	247
LCDDATA5	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	247
LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	247
LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	247
LCDDATA8	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	247
LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	247
LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	247
LCDDATA11	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	247
LCDPS	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0	244
LCDREF	LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	245
LCDRL	LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0	252
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	247
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	247
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	247
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	169

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the LCD module.

## 22.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

Figure 22-1 is a block diagram of the SPI interface module.

### 22.1 Master SSP (MSSP) Module Overview

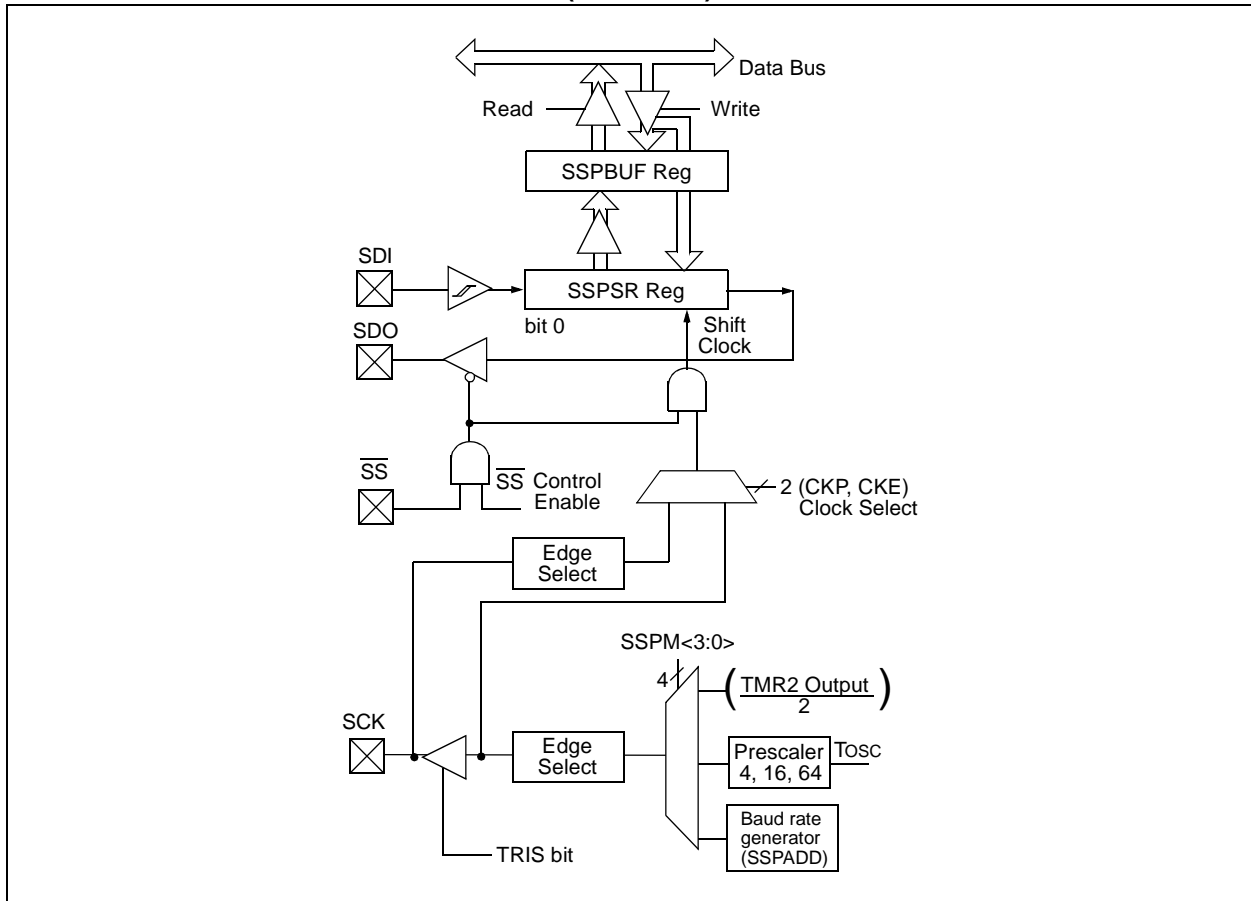
The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C™)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy chain connection of Slave devices

**FIGURE 22-1: MSSP BLOCK DIAGRAM (SPI MODE)**



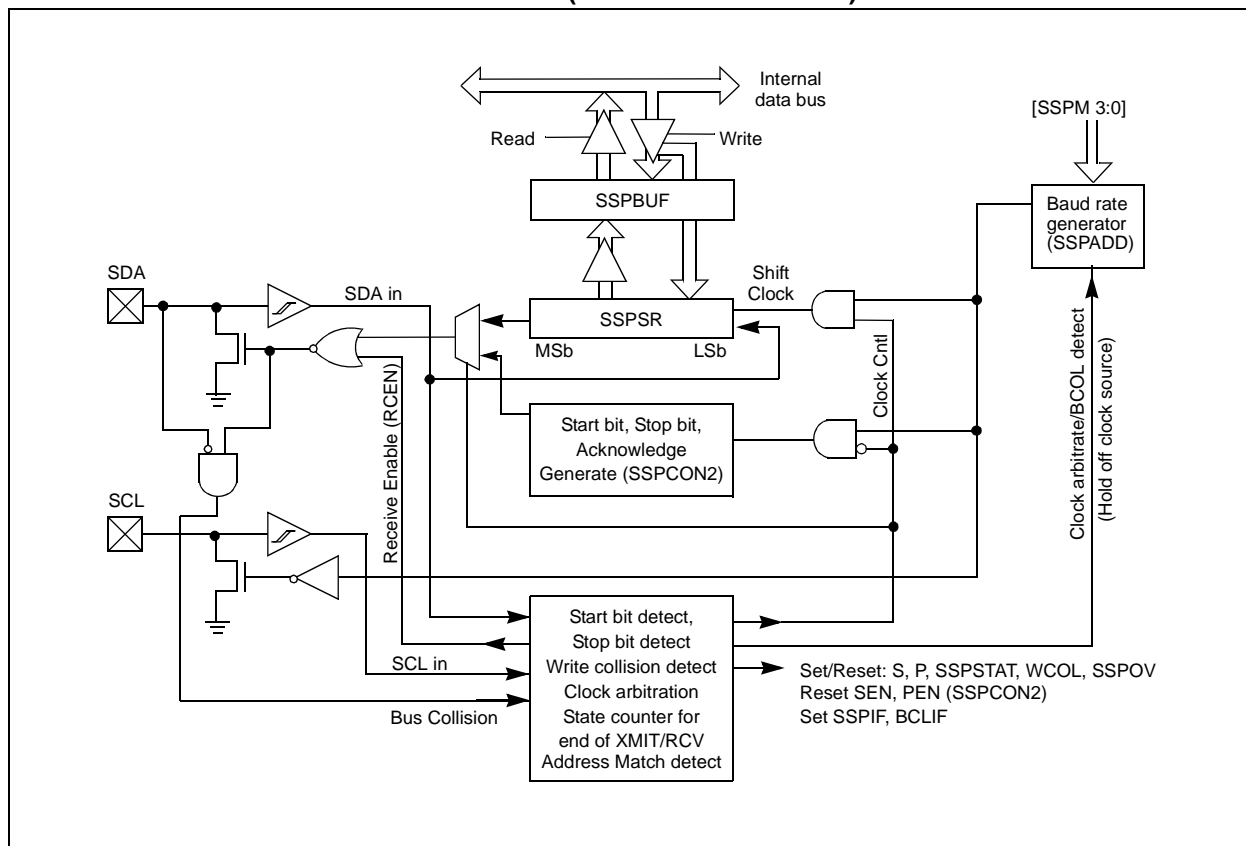
# PIC16F193X/LF193X

The I<sup>2</sup>C interface supports the following modes and features:

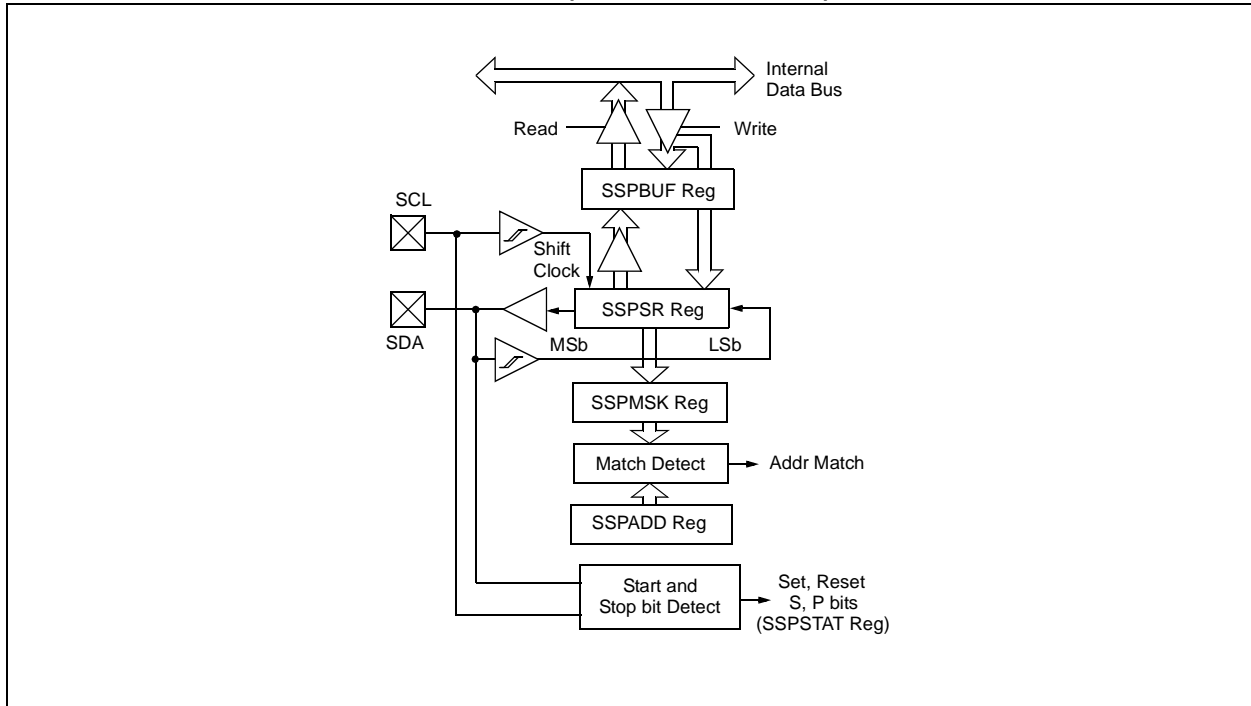
- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited Multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDA hold times

Figure 22-2 is a block diagram of the I<sup>2</sup>C interface module in Master mode. Figure 22-3 is a diagram of the I<sup>2</sup>C interface module in Slave mode.

**FIGURE 22-2: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



**FIGURE 22-3: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ SLAVE MODE)**



## 22.2 MSSP Control Registers

The MSSP module has seven associated registers:

- MSSP STATUS register (SSPSTAT)
- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Control Register 3 (SSPCON3)
- MSSP Address Masking register (SSPMSK)
- MSSP Data Buffer register (SSPBUF)
- MSSP Address register (SSPADD)

# PIC16F193X/LF193X

## REGISTER 22-1: SSPSTAT: SSP STATUS REGISTER

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

### Legend:

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

bit 7	<p><b>SMP:</b> SPI Data Input Sample bit</p> <p><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</p> <p><u>SPI Slave mode:</u> SMP must be cleared when SPI is used in Slave mode</p> <p><u>In I<sup>2</sup>C Master or Slave mode:</u> 1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz) 0 = Slew rate control enabled for high speed mode (400 kHz)</p>
bit 6	<p><b>CKE:</b> SPI Clock Edge Select bit (SPI mode only)</p> <p><u>CKP = 0:</u> 1 = Data transmitted on rising edge of SCK 0 = Data transmitted on falling edge of SCK</p> <p><u>CKP = 1:</u> 1 = Data transmitted on falling edge of SCK 0 = Data transmitted on rising edge of SCK</p>
bit 5	<p><b>D<math>\bar{A}</math>:</b> Data/Address bit (I<sup>2</sup>C mode only)</p> <p>1 = Indicates that the last byte received or transmitted was data 0 = Indicates that the last byte received or transmitted was address</p>
bit 4	<p><b>P:</b> Stop bit</p> <p>(I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)</p> <p>1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset) 0 = Stop bit was not detected last</p>
bit 3	<p><b>S:</b> Start bit</p> <p>(I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)</p> <p>1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset) 0 = Start bit was not detected last</p>
bit 2	<p><b>R<math>\bar{W}</math>:</b> Read/Write bit information (I<sup>2</sup>C mode only)</p> <p>This bit holds the R<math>\bar{W}</math> bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not ACK bit.</p> <p><u>In I<sup>2</sup>C Slave mode:</u> 1 = Read 0 = Write</p> <p><u>In I<sup>2</sup>C Master mode:</u> 1 = Transmit is in progress 0 = Transmit is not in progress</p> <p>OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.</p>
bit 1	<p><b>UA:</b> Update Address bit (10-bit I<sup>2</sup>C mode only)</p> <p>1 = Indicates that the user needs to update the address in the SSPADD register 0 = Address does not need to be updated</p>
bit 0	<p><b>BF:</b> Buffer Full Status bit</p> <p><u>Receive (SPI and I<sup>2</sup>C modes):</u> 1 = Receive complete, SSPBUF is full 0 = Receive not complete, SSPBUF is empty</p> <p><u>Transmit (I<sup>2</sup>C mode only):</u> 1 = Data transmit in progress (does not include the ACK and Stop bits), SSPBUF is full 0 = Data transmit complete (does not include the ACK and Stop bits), SSPBUF is empty</p>

## REGISTER 22-2: SSPCON1: SSP CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

### Legend:

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

bit 7	<p><b>WCOL:</b> Write Collision Detect bit</p> <p><u>Master mode:</u></p> <p>1 = A write to the SSPBUF register was attempted while the I<sup>2</sup>C conditions were not valid for a transmission to be started</p> <p>0 = No collision</p> <p><u>Slave mode:</u></p> <p>1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)</p> <p>0 = No collision</p>
bit 6	<p><b>SSPOV:</b> Receive Overflow Indicator bit<sup>(1)</sup></p> <p><u>In SPI mode:</u></p> <p>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. 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 (must be cleared in software).</p> <p>0 = No overflow</p> <p><u>In I<sup>2</sup>C mode:</u></p> <p>1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).</p> <p>0 = No overflow</p>
bit 5	<p><b>SSPEN:</b> Synchronous Serial Port Enable bit</p> <p>In both modes, when enabled, these pins must be properly configured as input or output</p> <p><u>In SPI mode:</u></p> <p>1 = Enables serial port and configures SCK, SDO, SDI and <math>\overline{SS}</math> as the source of the serial port pins<sup>(2)</sup></p> <p>0 = Disables serial port and configures these pins as I/O port pins</p> <p><u>In I<sup>2</sup>C mode:</u></p> <p>1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins<sup>(3)</sup></p> <p>0 = Disables serial port and configures these pins as I/O port pins</p>
bit 4	<p><b>CKP:</b> Clock Polarity Select bit</p> <p><u>In SPI mode:</u></p> <p>1 = Idle state for clock is a high level</p> <p>0 = Idle state for clock is a low level</p> <p><u>In I<sup>2</sup>C Slave mode:</u></p> <p>SCK release control</p> <p>1 = Enable clock</p> <p>0 = Holds clock low (clock stretch). (Used to ensure data setup time.)</p> <p><u>In I<sup>2</sup>C Master mode:</u></p> <p>Unused in this mode</p>
bit 3-0	<p><b>SSPM&lt;3:0&gt;:</b> Synchronous Serial Port Mode Select bits</p> <p>0000 = SPI Master mode, clock = Fosc/4</p> <p>0001 = SPI Master mode, clock = Fosc/16</p> <p>0010 = SPI Master mode, clock = Fosc/64</p> <p>0011 = SPI Master mode, clock = TMR2 output/2</p> <p>0100 = SPI Slave mode, clock = SCK pin, <math>\overline{SS}</math> pin control enabled</p> <p>0101 = SPI Slave mode, clock = SCK pin, <math>\overline{SS}</math> pin control disabled, <math>\overline{SS}</math> can be used as I/O pin</p> <p>0110 = I<sup>2</sup>C Slave mode, 7-bit address</p> <p>0111 = I<sup>2</sup>C Slave mode, 10-bit address</p> <p>1000 = I<sup>2</sup>C Master mode, clock = Fosc / (4 * (SSPADD+1))<sup>(4)</sup></p> <p>1001 = Reserved</p> <p>1010 = SPI Master mode, clock = Fosc/(4 * (SSPADD+1))</p> <p>1011 = I<sup>2</sup>C firmware controlled Master mode (Slave idle)</p> <p>1100 = Reserved</p> <p>1101 = Reserved</p> <p>1110 = I<sup>2</sup>C Slave mode, 7-bit address with Start and Stop bit interrupts enabled</p> <p>1111 = I<sup>2</sup>C Slave mode, 10-bit address with Start and Stop bit interrupts enabled</p>

- Note**
- 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
  - 2: When enabled, these pins must be properly configured as input or output.
  - 3: When enabled, the SDA and SCL pins must be configured as inputs.
  - 4: SSPADD values of 0, 1 or 2 are not supported for I<sup>2</sup>C Mode.

# PIC16F193X/LF193X

## REGISTER 22-3: SSPCON2: SSP CONTROL REGISTER 2

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

### Legend:

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

- bit 7      **GCEN:** General Call Enable bit (in I<sup>2</sup>C Slave mode only)  
 1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR  
 0 = General call address disabled
- bit 6      **ACKSTAT:** Acknowledge Status bit (in I<sup>2</sup>C mode only)  
 1 = Acknowledge was not received  
 0 = Acknowledge was received
- bit 5      **ACKDT:** Acknowledge Data bit (in I<sup>2</sup>C mode only)  
In Receive mode:  
 Value transmitted when the user initiates an Acknowledge sequence at the end of a receive  
 1 = Not Acknowledge  
 0 = Acknowledge
- bit 4      **ACKEN:** Acknowledge Sequence Enable bit (in I<sup>2</sup>C Master mode only)  
In Master Receive mode:  
 1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit.  
 Automatically cleared by hardware.  
 0 = Acknowledge sequence idle
- bit 3      **RCEN:** Receive Enable bit (in I<sup>2</sup>C Master mode only)  
 1 = Enables Receive mode for I<sup>2</sup>C  
 0 = Receive idle
- bit 2      **PEN:** Stop Condition Enable bit (in I<sup>2</sup>C Master mode only)  
SCK Release Control:  
 1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Stop condition Idle
- bit 1      **RSEN:** Repeated Start Condition Enabled bit (in I<sup>2</sup>C Master mode only)  
 1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Repeated Start condition Idle
- bit 0      **SEN:** Start Condition Enabled bit (in I<sup>2</sup>C Master mode only)  
In Master mode:  
 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.  
 0 = Start condition Idle  
In Slave mode:  
 1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)  
 0 = Clock stretching is disabled

**Note 1:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I<sup>2</sup>C module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).



# PIC16F193X/LF193X

## REGISTER 22-4: SSPCON3: SSP CONTROL REGISTER 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

### Legend:

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

- bit 7      **ACKTIM:** Acknowledge Time Status bit (I<sup>2</sup>C mode only)  
 1 = Indicates the I<sup>2</sup>C bus is in an Acknowledge sequence, set on 8<sup>TH</sup> falling edge of SCL clock  
 0 = Not an Acknowledge sequence, cleared on 9<sup>TH</sup> rising edge of SCL clock
- bit 6      **PCIE:** Stop Condition Interrupt Enable bit (I<sup>2</sup>C mode only)  
 1 = Enable interrupt on detection of Stop condition  
 0 = Stop detection interrupts are disabled<sup>(2)</sup>
- bit 5      **SCIE:** Start Condition Interrupt Enable bit (I<sup>2</sup>C mode only)  
 1 = Enable interrupt on detection of Start or Restart conditions  
 0 = Start detection interrupts are disabled<sup>(2)</sup>
- bit 4      **BOEN:** Buffer Overwrite Enable bit  
In SPI Slave mode:<sup>(1)</sup>  
 1 = SSPBUF updates every time that a new data byte is shifted in ignoring the BF bit  
 0 = If new byte is received with BF bit of the SSPSTAT register already set, SSPOV bit of the SSPCON1 register is set, and the buffer is not updated  
In I<sup>2</sup>C Master mode:  
 This bit is ignored.  
In I<sup>2</sup>C Slave mode:  
 1 = SSPBUF is updated and  $\overline{\text{ACK}}$  is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.  
 0 = SSPBUF is only updated when SSPOV is clear
- bit 3      **SDAHT:** SDA Hold Time Selection bit (I<sup>2</sup>C mode only)  
 1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL  
 0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL
- bit 2      **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I<sup>2</sup>C Slave mode only)  
 If on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCLIF bit of the PIR2 register is set, and bus goes idle  
 1 = Enable slave bus collision interrupts  
 0 = Slave bus collision interrupts are disabled
- bit 1      **AHEN:** Address Hold Enable bit (I<sup>2</sup>C Slave mode only)  
 1 = Following the 8th falling edge of SCL for a matching received address byte; CKP bit of the SSPCON1 register will be cleared and the SCL will be held low.  
 0 = Address holding is disabled
- bit 0      **DHEN:** Data Hold Enable bit (I<sup>2</sup>C Slave mode only)  
 1 = Following the 8th falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPCON1 register and SCL is held low.  
 0 = Data holding is disabled

- Note 1:** For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPBUF.
- 2:** This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.

# PIC16F193X/LF193X

## REGISTER 22-5: SSPMSK: SSP MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0 <sup>(2)</sup>
bit 7							bit 0

### Legend:

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

bit 7-1                      **MSK<7:1>**: Mask bits  
1 = The received address bit n is compared to SSPADD<n> to detect I<sup>2</sup>C address match  
0 = The received address bit n is not used to detect I<sup>2</sup>C address match

bit 0                      **MSK<0>**: Mask bit for I<sup>2</sup>C Slave mode, 10-bit Address  
I<sup>2</sup>C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):  
1 = The received address bit 0 is compared to SSPADD<0> to detect I<sup>2</sup>C address match  
0 = The received address bit 0 is not used to detect I<sup>2</sup>C address match  
I<sup>2</sup>C Slave mode, 7-bit address, the bit is ignored

## REGISTER 22-6: SSPADD: MSSP ADDRESS AND BAUD RATE REGISTER (I<sup>2</sup>C MODE)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
bit 7							bit 0

### Legend:

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

### Master mode:

bit 7-0                      **ADD<7:0>**: Baud Rate Clock Divider bits  
SCL pin clock period = ((ADD<7:0> + 1) \* 4)/Fosc

### 10-Bit Slave mode — Most Significant Address byte:

bit 7-3                      **Not used**: Unused for Most Significant Address byte. Bit state of this register is a "don't care". Bit pattern sent by master is fixed by I<sup>2</sup>C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1                      **ADD<2:1>**: Two Most Significant bits of 10-bit address

bit 0                      **Not used**: Unused in this mode. Bit state is a "don't care".

### 10-Bit Slave mode — Least Significant Address byte:

bit 7-0                      **ADD<7:0>**: Eight Least Significant bits of 10-bit address

### 7-Bit Slave mode:

bit 7-1                      **ADD<7:1>**: 7-bit address

bit 0                      **Not used**: Unused in this mode. Bit state is a "don't care".

## 22.3 SPI Mode

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

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

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

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

Figure 22-1 shows the block diagram of the MSSP module when operating in SPI mode.

### 22.3.1 REGISTERS

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSPSTAT)
- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 3 (SSPCON3)
- MSSP Data Buffer register (SSPBUF)
- MSSP Address register (SSPADD)
- MSSP Shift register (SSPSR)  
(Not directly accessible)

SSPCON1 and SSPSTAT are the control and STATUS registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

In one SPI master mode, SSPADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 22.7 “Baud Rate Generator”**.

SSPSR is the shift register used for shifting data in and out. SSPBUF provides indirect access to the SSPSR register. SSPBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPSR and SSPBUF together create a buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

### 22.3.2 OPERATIONS

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

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

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full Detect bit, BF of the SSPSTAT register, 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 of the SSPCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPBUF register to complete 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. The Buffer Full bit, BF of the SSPSTAT register, 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 MSSP interrupt is used to determine when the transmission/reception has completed. 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.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSPSTAT register indicates the various Status conditions.

# PIC16F193X/LF193X

## 22.3.3 ENABLING SPI I/O

To enable the serial port, SSP Enable bit, SSPEN of the SSPCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCONx registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and 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 as follows:

- SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- $\overline{SS}$  must have corresponding TRIS bit set

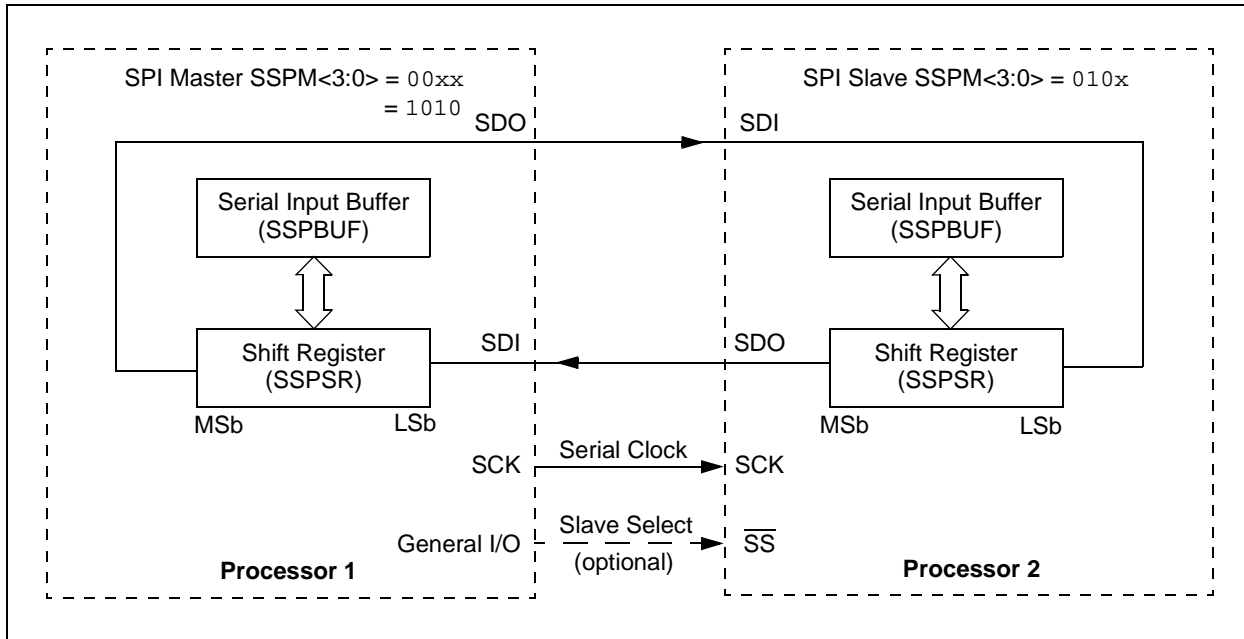
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

## 22.3.4 TYPICAL CONNECTION

Figure 22-4 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 22-4: SPI MASTER/SLAVE CONNECTION**



## 22.3.5 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 22-4) 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).

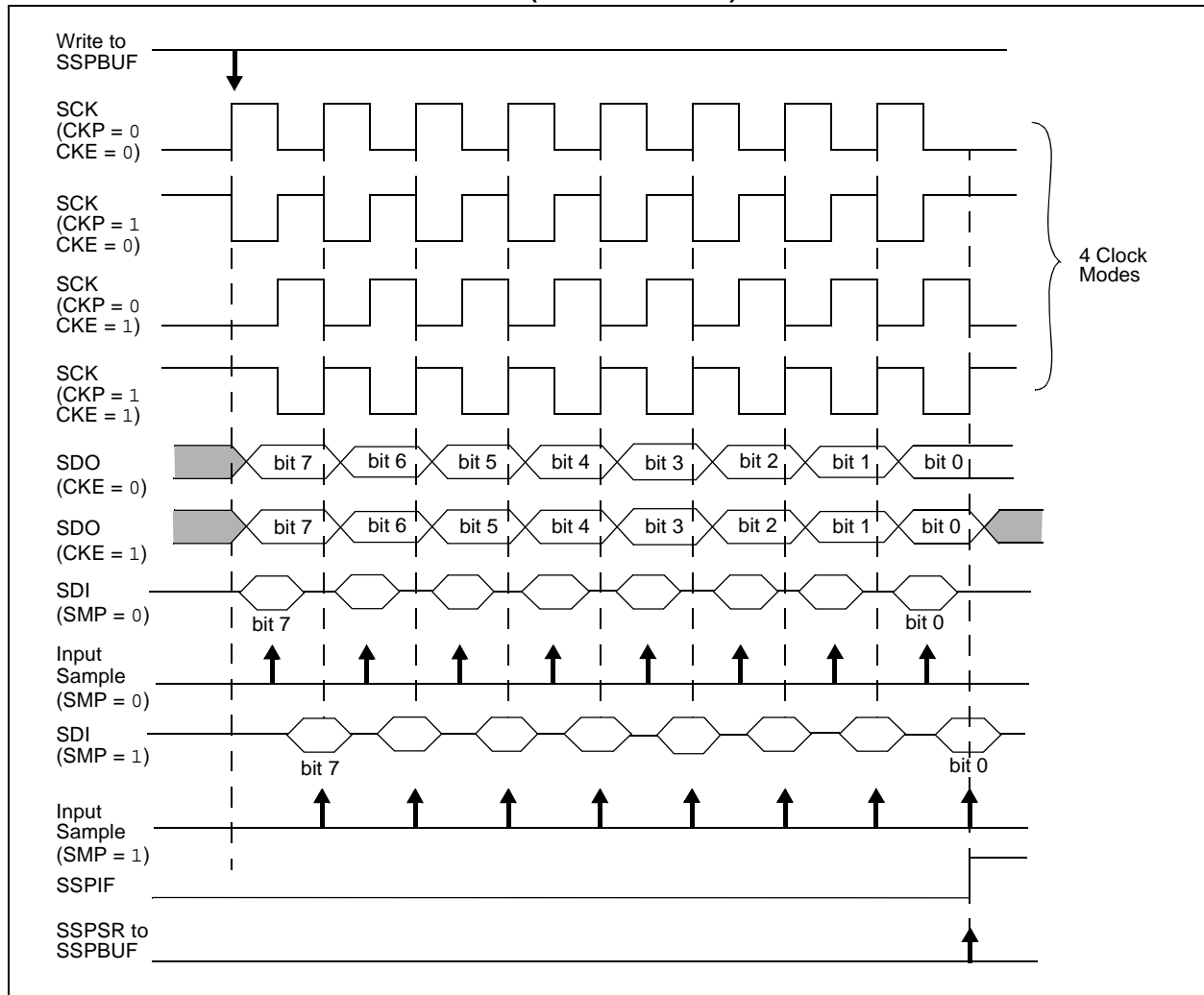
The clock polarity is selected by appropriately programming the CKP bit of the SSPCON1 register and the CKE bit of the SSPSTAT register. This then, would give waveforms for SPI communication as shown in Figure 22-5, Figure 22-6 and Figure 22-7, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{osc}/4$  (or  $T_{CY}$ )
- $F_{osc}/16$  (or  $4 * T_{CY}$ )
- $F_{osc}/64$  (or  $16 * T_{CY}$ )
- Timer2 output/2
- $F_{osc}/(4 * (SSPADD + 1))$

Figure 22-5 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 22-5: SPI MODE WAVEFORM (MASTER MODE)**



# PIC16F193X/LF193X

## 22.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPCON1 register.

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. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

### 22.3.6.1 Buffer Overwrite Enable

In SPI daisy-chained configurations only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPCON3 register will enable writes to the SSPBUF register, even if the previous byte has not been read. Allowing the software to ignore data that may not apply to it.

## 22.3.7 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 ( $SSPCON1\langle 3:0 \rangle = 0100$ ).

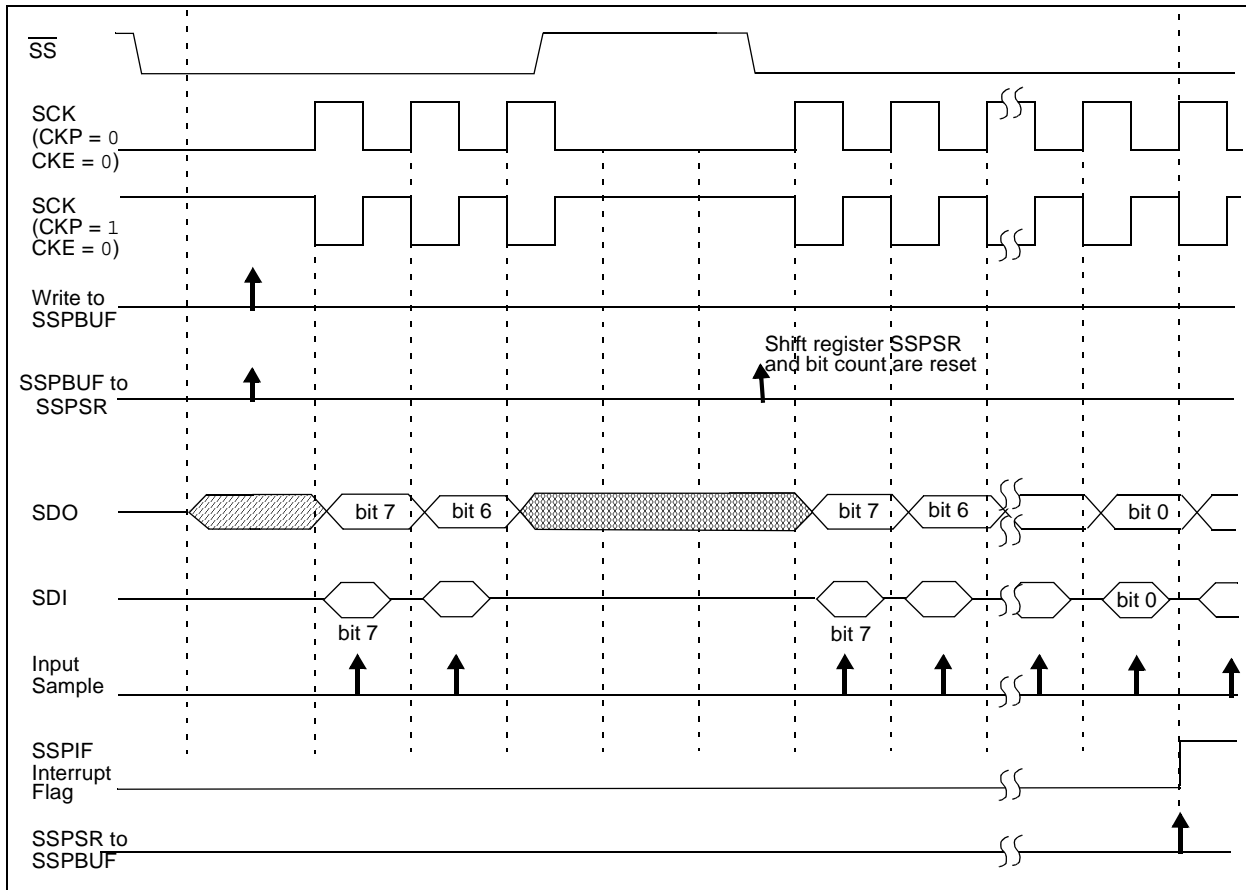
When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven.

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.

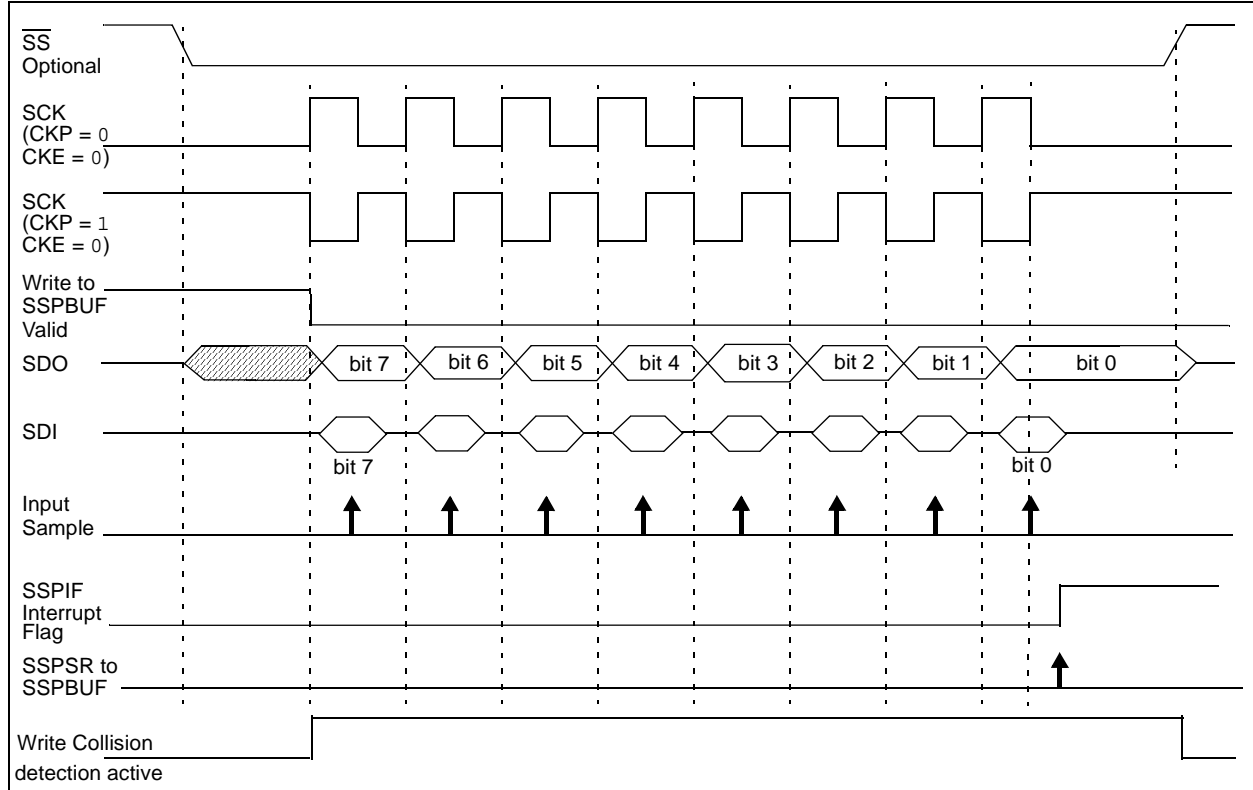
- Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPCON1\langle 3:0 \rangle = 0100$ ), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.
- 2:** When the SPI is used in Slave mode with CKE set; the user must enable  $\overline{SS}$  pin control.
- 3:** While operated in SPI Slave mode the SMP bit of the SSPSTAT register must remain clear.

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.

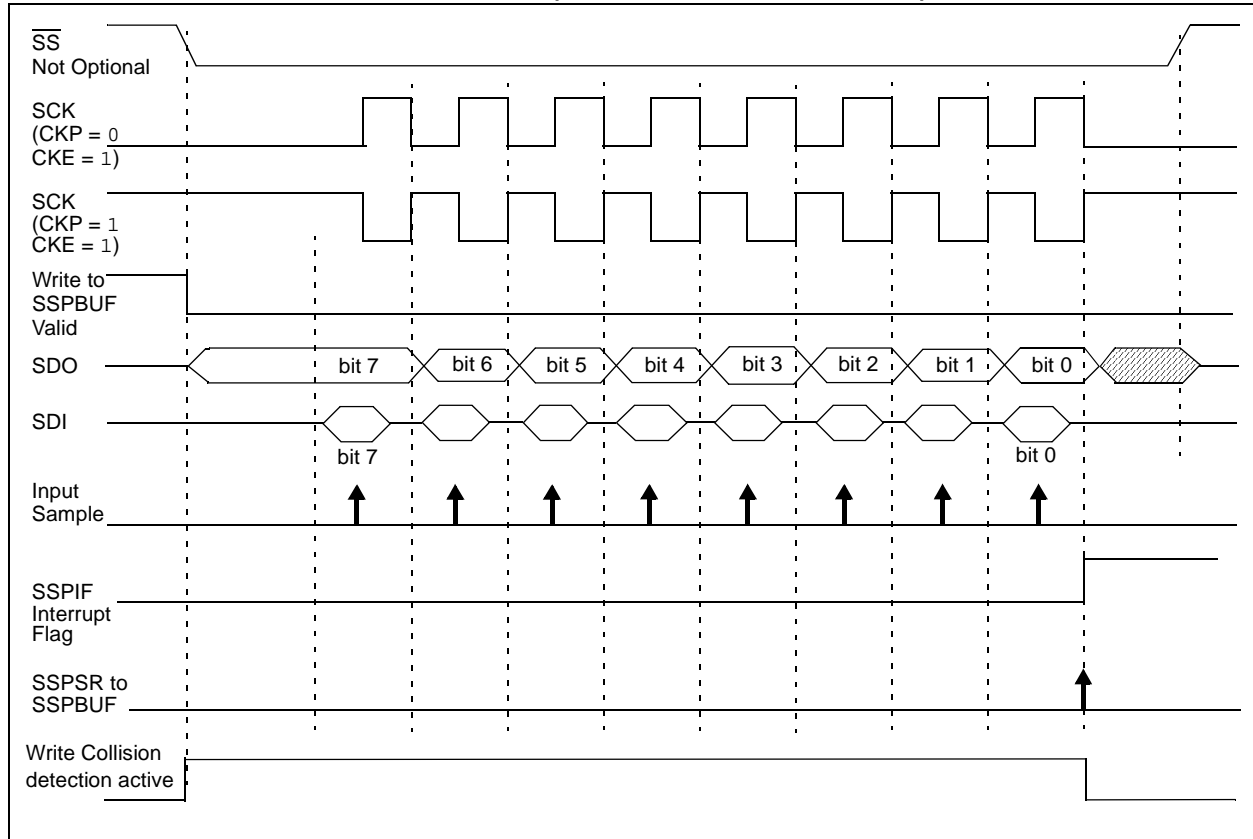
**FIGURE 22-6: SLAVE SELECT SYNCHRONIZATION WAVEFORM**



**FIGURE 22-7: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**



**FIGURE 22-8: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



# PIC16F193X/LF193X

## 22.3.8 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

When MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller:

- from Sleep, in Slave mode
- from Idle, in Slave or Master mode

If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

**TABLE 22-1: REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	84
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIR1	TMR1GIf	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								281*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	277
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	279
SSPSTAT	SMP	CKE	D/Ā	P	S	R/Ā	UA	BF	276
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	86
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	94

**Legend:** Shaded cells are not used by the MSSP in SPI mode.

\* Page provides register information.



## 22.4 I<sup>2</sup>C MODE

All MSSP I<sup>2</sup>C communication is byte oriented and shifted out MSb first. Six SFR registers and 2 interrupt flags interface the module with the PIC<sup>®</sup> microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I<sup>2</sup>C devices.

### 22.4.1 SDA AND SCL PINS

Selection of any I<sup>2</sup>C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

**Note:** Data is tied to output zero when an I<sup>2</sup>C mode is enabled.

### 22.4.2 BYTE FORMAT

All communication in I<sup>2</sup>C is done in 9-bit segments. A byte is sent from a Master to a Slave or vice-versa, followed by an Acknowledge bit sent back. After the 8th falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

### 22.4.3 DEFINITION OF I<sup>2</sup>C TERMINOLOGY

There is language and terminology in the description of I<sup>2</sup>C communication that have definitions specific to I<sup>2</sup>C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Phillips I<sup>2</sup>C specification.

**TABLE 22-2: I<sup>2</sup>C BUS TERMS**

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

# PIC16F193X/LF193X

## 22.4.4 START CONDITION

The I<sup>2</sup>C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 22-8 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I<sup>2</sup>C Specification that states no bus collision can occur on a Start.

**Note:** The Philips I<sup>2</sup>C Specification states that a bus collision cannot occur on a Start, and should occur during the address sequence.

## 22.4.5 STOP CONDITION

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

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

## 22.4.6 RESTART CONDITION

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

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

After a full match with R/W clear in 10-bit mode, a prior match flag is set and maintained. Until a Stop condition, a high address with R/W clear, or high address match fails.

## 22.4.7 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPCON3 register can enable the generation of an interrupt in Slave mode. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

FIGURE 22-9: I<sup>2</sup>C START AND STOP CONDITIONS

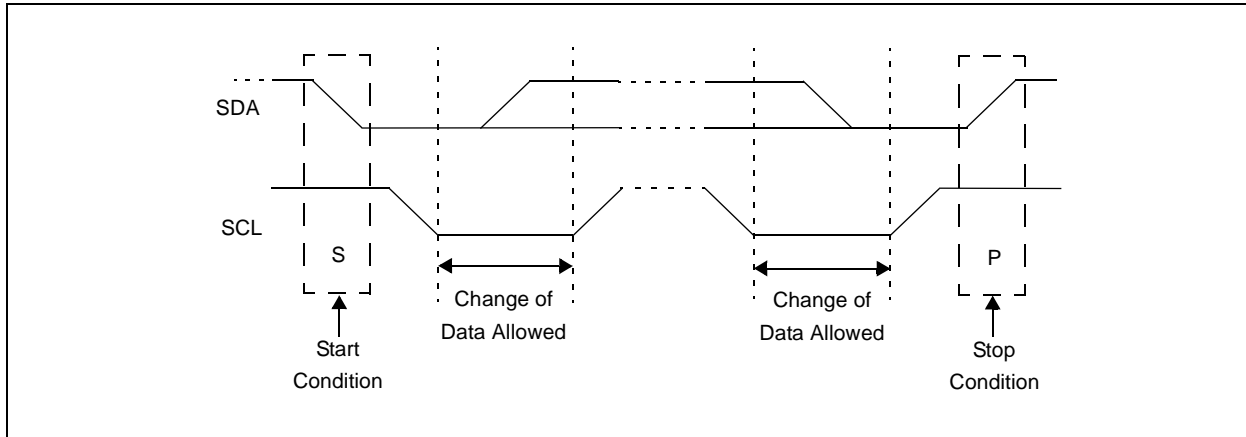
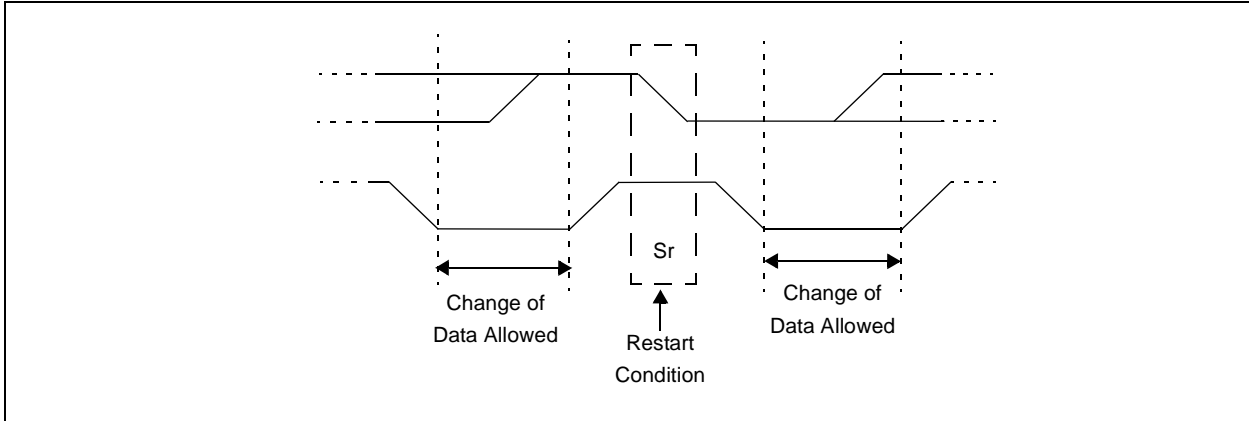


FIGURE 22-10: I<sup>2</sup>C RESTART CONDITION



# PIC16F193X/LF193X

---

## 22.4.8 ACKNOWLEDGE SEQUENCE

The 9th SCL pulse for any transferred byte in I<sup>2</sup>C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ( $\overline{ACK}$ ) is an active-low signal, pulling the SDA line low indicated to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an  $\overline{ACK}$  is placed in the ACKSTAT bit of the SSPCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the  $\overline{ACK}$  value sent back to the transmitter. The ACKDT bit of the SSPCON2 register is set/cleared to determine the response.

Slave hardware will generate an  $\overline{ACK}$  response if the AHEN and DHEN bits of the SSPCON3 register are clear.

There are certain conditions where an  $\overline{ACK}$  will not be sent by the slave. If the BF bit of the SSPSTAT register or the SSPOV bit of the SSPCON1 register are set when a byte is received.

## 22.5 I<sup>2</sup>C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected in the SSPM bits of SSPCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operated the same as the other modes with SSPIF additionally getting set upon detection of a Start, Restart, or Stop condition.

### 22.5.1 SLAVE MODE ADDRESSES

The SSPADD register (Register 22-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches the value is loaded into the SSPBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 22-5) affects the address matching process. See **Section 22.5.9 “SSP Mask Register”** for more information.

#### 22.5.1.1 I<sup>2</sup>C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

#### 22.5.1.2 I<sup>2</sup>C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb of the 10-bit address and stored in bits 2 and 1 of the SSPADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPADD with the low address. The low address byte is clocked in and all 8 bits are compared to the low address value in SSPADD. Even if there is not an address match; SSPIF and UA are set, and SCL is held low until SSPADD is updated to receive a high byte again. When SSPADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/W bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

## 22.5.2 SLAVE RECEPTION

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

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPSTAT register is set, or bit SSPOV of the SSPCON1 register is set. The BOEN bit of the SSPCON3 register modifies this operation. For more information see Register 22-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPIF, must be cleared by software.

When the SEN bit of the SSPCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPCON1 register, except sometimes in 10-bit mode. See **Section 22.3.5 “Master Mode”** for more detail.

### 22.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I<sup>2</sup>C Slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 22-10 and Figure 22-11 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I<sup>2</sup>C communication.

1. Start bit detected.
2. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with  $\overline{R/W}$  bit clear is received.
4. The slave pulls SDA low sending an  $\overline{ACK}$  to the master, and sets SSPIF bit.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an  $\overline{ACK}$  to the master, and sets SSPIF bit.
10. Software clears SSPIF.
11. Software reads the received byte from SSPBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the Master.
13. Master sends Stop condition, setting P bit of SSPSTAT, and the bus goes idle.

### 22.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the 8th falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to ACK the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I<sup>2</sup>C communication. Figure 22-12 displays a module using both address and data holding. Figure 22-13 includes the operation with the SEN bit of the SSPCON2 register set.

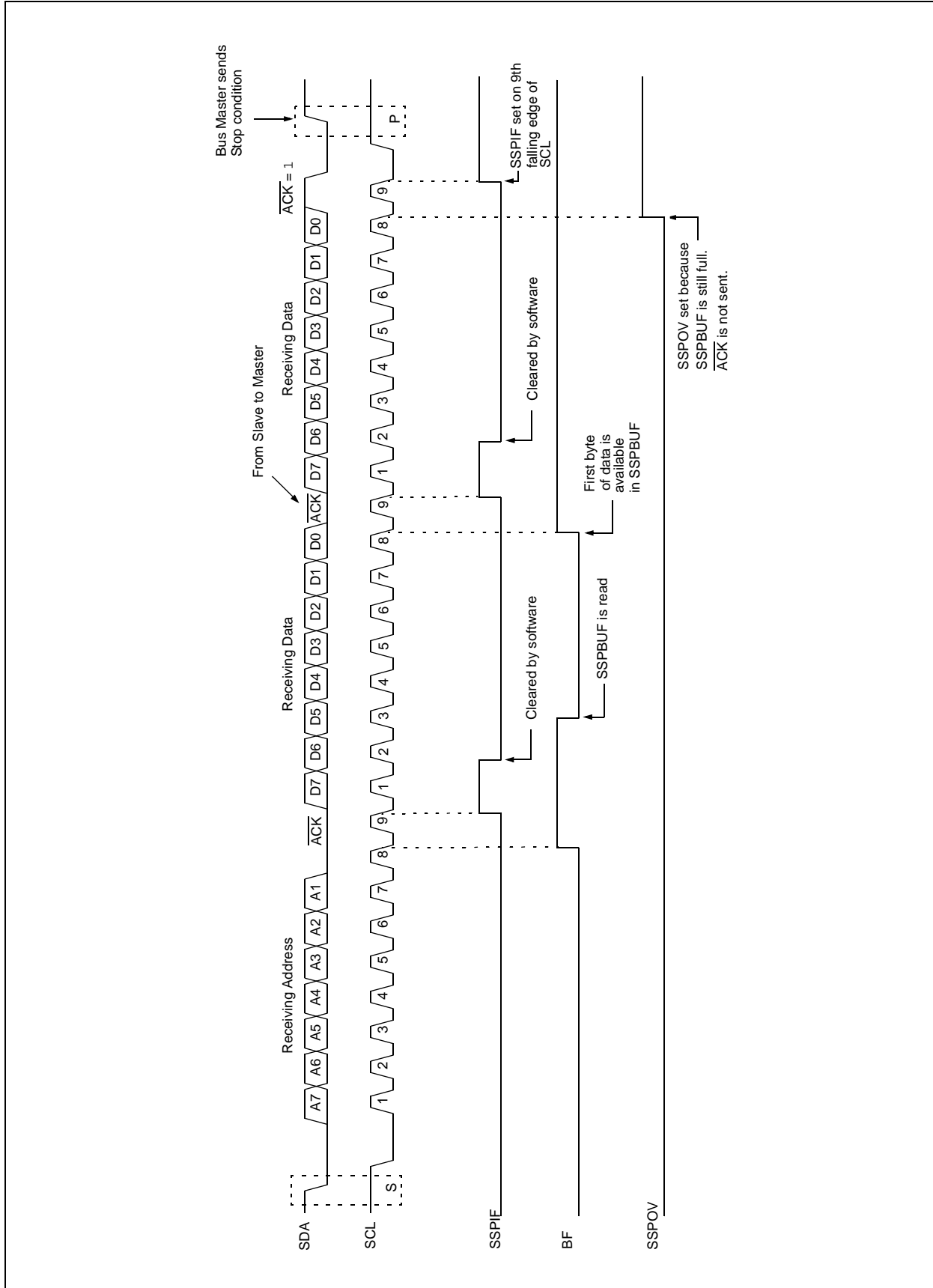
1. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
2. Matching address with  $\overline{R/W}$  bit clear is clocked in. SSPIF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSPIF.
4. Slave can look at the ACKTIM bit of the SSPCON3 register to determine if the SSPIF was after or before the  $\overline{ACK}$ .
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets  $\overline{ACK}$  value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPIF is set after an  $\overline{ACK}$ , not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the  $\overline{ACK}$ .
10. Slave clears SSPIF.

**Note:** SSPIF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to Master is SSPIF not set

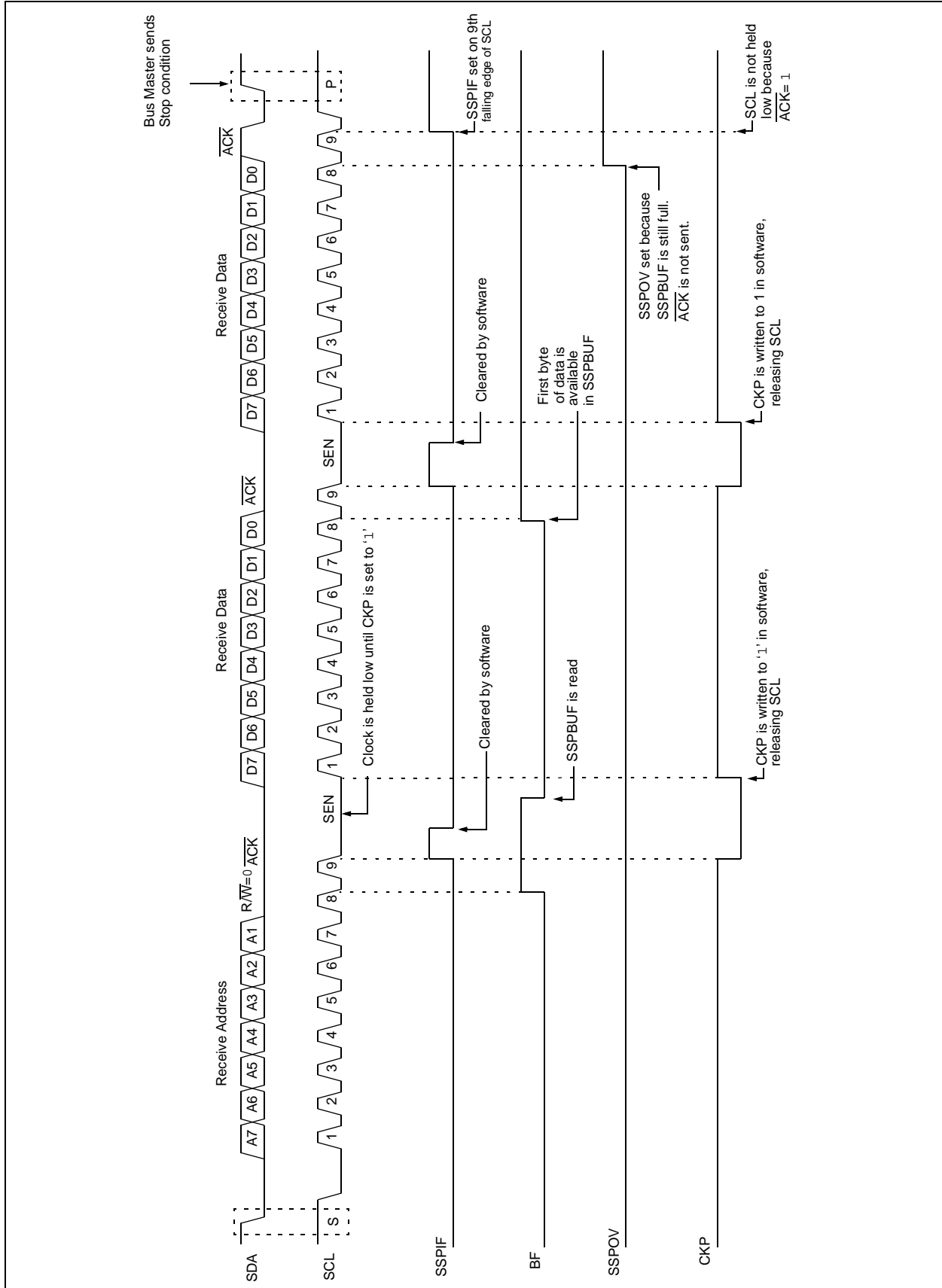
11. SSPIF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an  $\overline{ACK} = 1$ , or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

# PIC16F193X/LF193X

FIGURE 22-11: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)

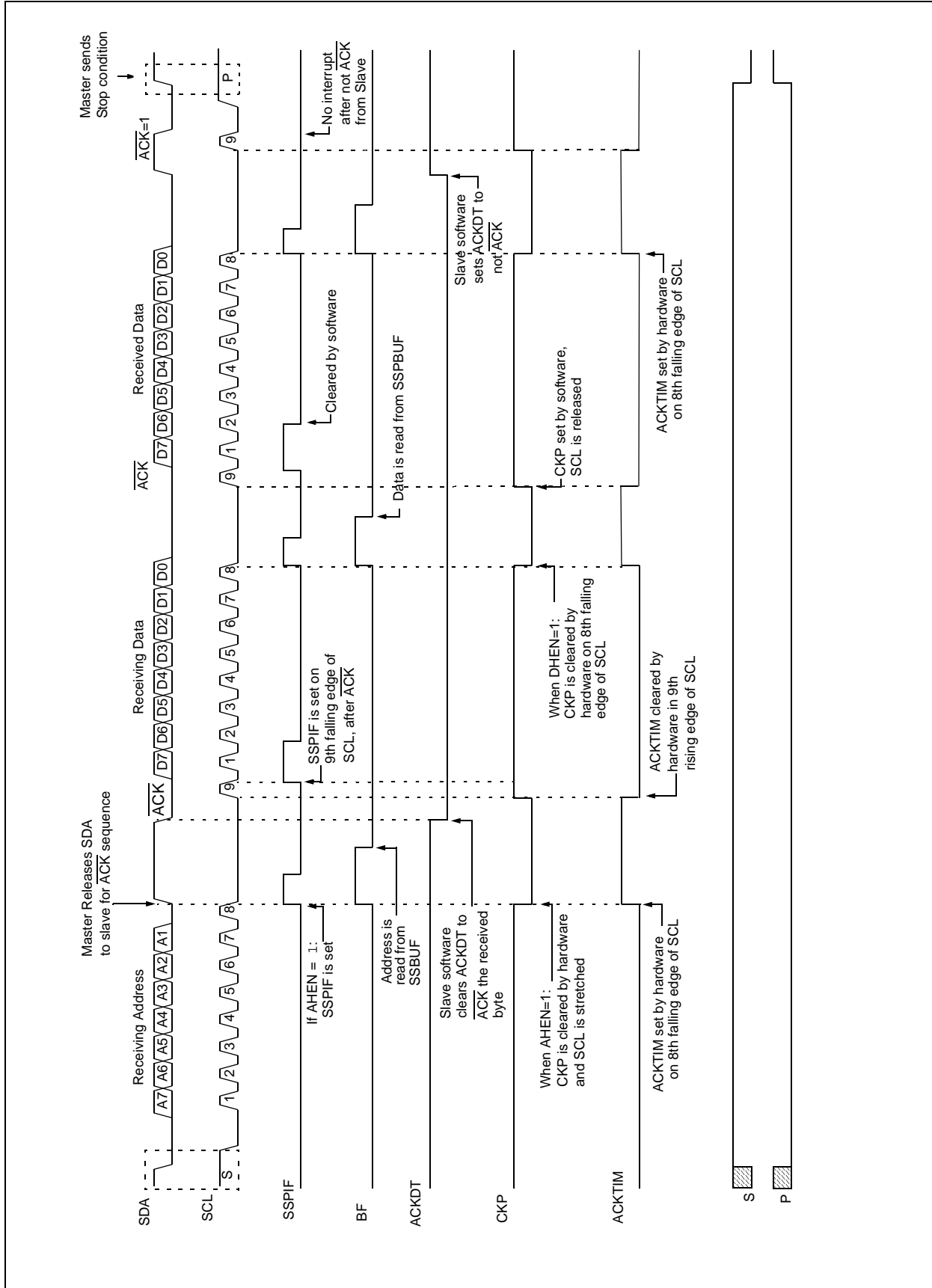


**FIGURE 22-12: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**



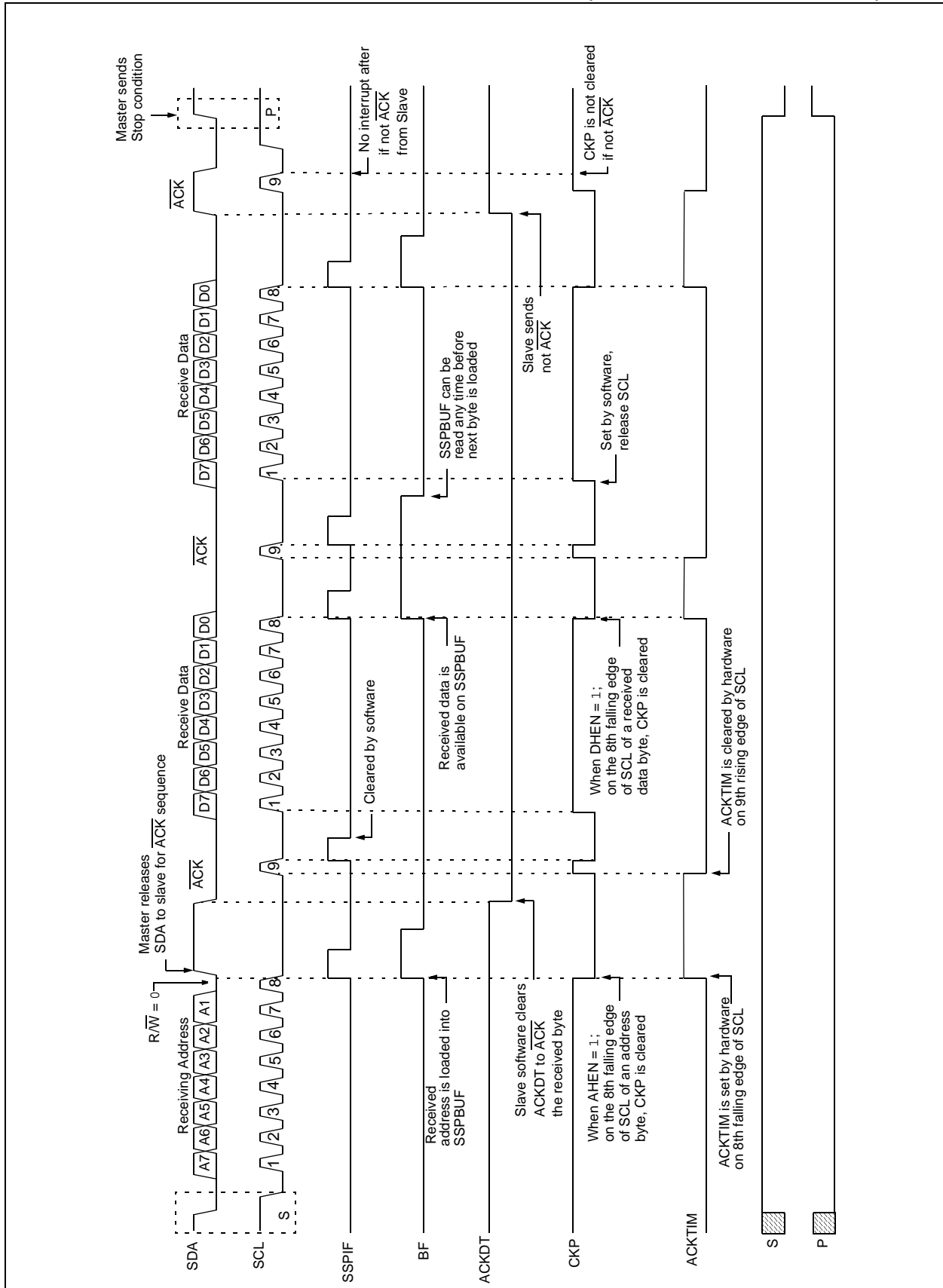
# PIC16F193X/LF193X

FIGURE 22-13: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)





**FIGURE 22-14: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)**



# PIC16F193X/LF193X

## 22.5.3 SLAVE TRANSMISSION

When the  $\overline{R/W}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register, and an  $\overline{ACK}$  pulse is sent by the slave on the ninth bit.

Following the  $\overline{ACK}$ , slave hardware clears the CKP bit and the SCL pin is held low (see **Section 22.5.6 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit of the SSPCON1 register. 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.

The  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This  $\overline{ACK}$  value is copied to the ACKSTAT bit of the SSPCON2 register. If ACKSTAT is set (not  $\overline{ACK}$ ), then the data transfer is complete. In this case, when the not  $\overline{ACK}$  is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ( $\overline{ACK}$ ), the next transmit data must be loaded into the SSPBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared by software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

### 22.5.3.1 7-bit Transmission

A Master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 22-14 can be used as a reference to this list.

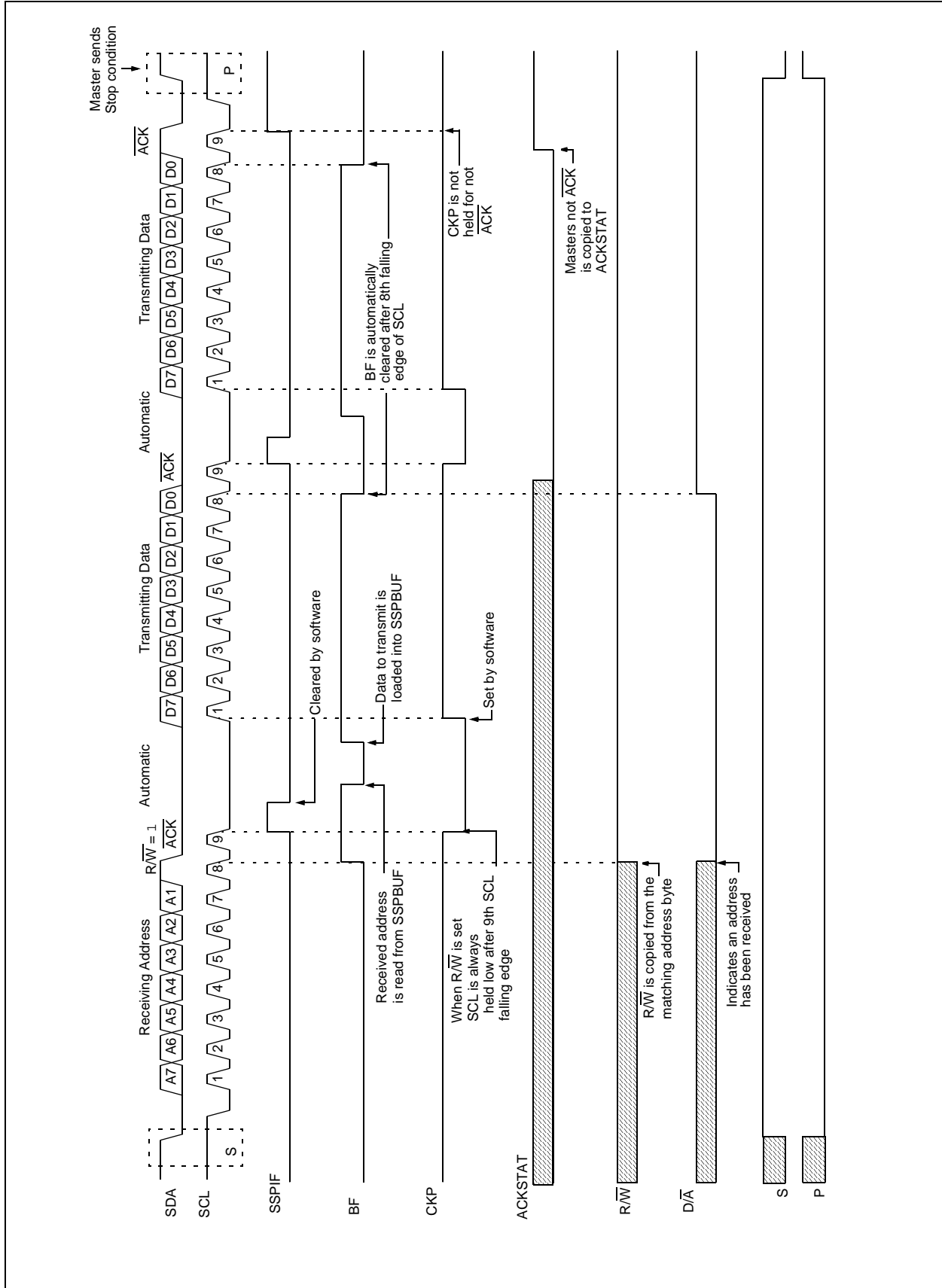
1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with  $\overline{R/W}$  bit set is received by the Slave setting SSPIF bit.
4. Slave hardware generates an  $\overline{ACK}$  and sets SSPIF.
5. SSPIF bit is cleared by user.
6. Software reads the received address from SSPBUF, clearing BF.
7.  $\overline{R/W}$  is set so CKP was automatically cleared after the  $\overline{ACK}$ .
8. The slave software loads the transmit data into SSPBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPIF is set after the  $\overline{ACK}$  response from the master is loaded into the ACKSTAT register.
11. SSPIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

**Note 1:** If the master  $\overline{ACK}$ s the clock will be stretched.

**2:** ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not  $\overline{ACK}$ ; the clock is not held, but SSPIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

FIGURE 22-15: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)



# PIC16F193X/LF193X

---

## 22.5.3.2 7-bit TRANSMISSION WITH Address Hold Enabled

Setting the AHEN bit of the SSPCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPIF interrupt is set.

Figure 22-15 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

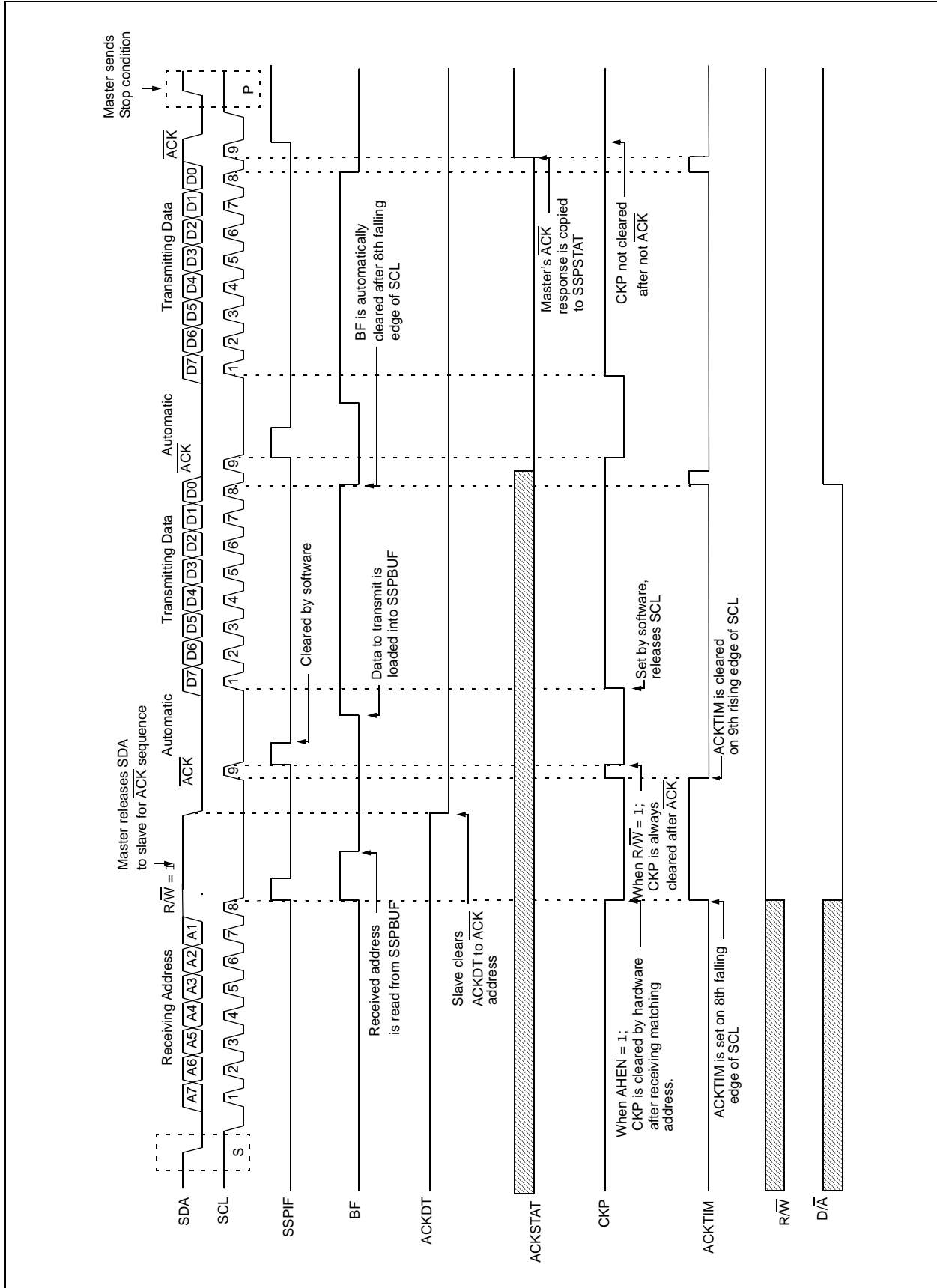
1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with  $\overline{R/\overline{W}}$  bit set. After the 8th falling edge of the SCL line the CKP bit is cleared and SSPIF interrupt is generated.
4. Slave software clears SSPIF.
5. Slave software reads ACKTIM bit of SSPCON3 register, and  $\overline{R/\overline{W}}$  and  $\overline{D/\overline{A}}$  of the SSPSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to  $\overline{ACK}$  or not  $\overline{ACK}$  and sets ACKDT bit of the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the  $\overline{ACK}$  value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPIF after the  $\overline{ACK}$  if the  $\overline{R/\overline{W}}$  bit is set.
11. Slave software clears SSPIF.
12. Slave loads value to transmit to the master into SSPBUF setting the BF bit.

**Note:**  $\overline{SSPBUF}$  cannot be loaded until after the  $\overline{ACK}$ .

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an  $\overline{ACK}$  value on the 9th SCL pulse.
15. Slave hardware copies the  $\overline{ACK}$  value into the ACKSTAT bit of the SSPCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not  $\overline{ACK}$  the slave releases the bus allowing the master to send a Stop and end the communication.

**Note:** Master must send a not  $\overline{ACK}$  on the last byte to ensure that the slave releases the SCL line to receive a Stop.

**FIGURE 22-16: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 1)**



# PIC16F193X/LF193X

---

## 22.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSP module configured as an I<sup>2</sup>C Slave in 10-bit Addressing mode.

Figure 22-16 and is used as a visual reference for this description.

This is a step by step process of what must be done by slave software to accomplish I<sup>2</sup>C communication.

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with  $\overline{R/W}$  bit clear; UA bit of the SSPSTAT register is set.
4. Slave sends  $\overline{ACK}$  and SSPIF is set.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPBUF clearing the BF flag.
7. Slave loads low address into SSPADD, releasing SCL.
8. Master sends matching low address byte to the Slave; UA bit is set.

**Note:** Updates to the SSPADD register are not allowed until after the ACK sequence.

9. Slave sends  $\overline{ACK}$  and SSPIF is set.

**Note:** If the low address does not match, SSPIF and UA are still set so that the slave software can set SSPADD back to the high address. BF is not set because there is no match. CKP is unaffected.

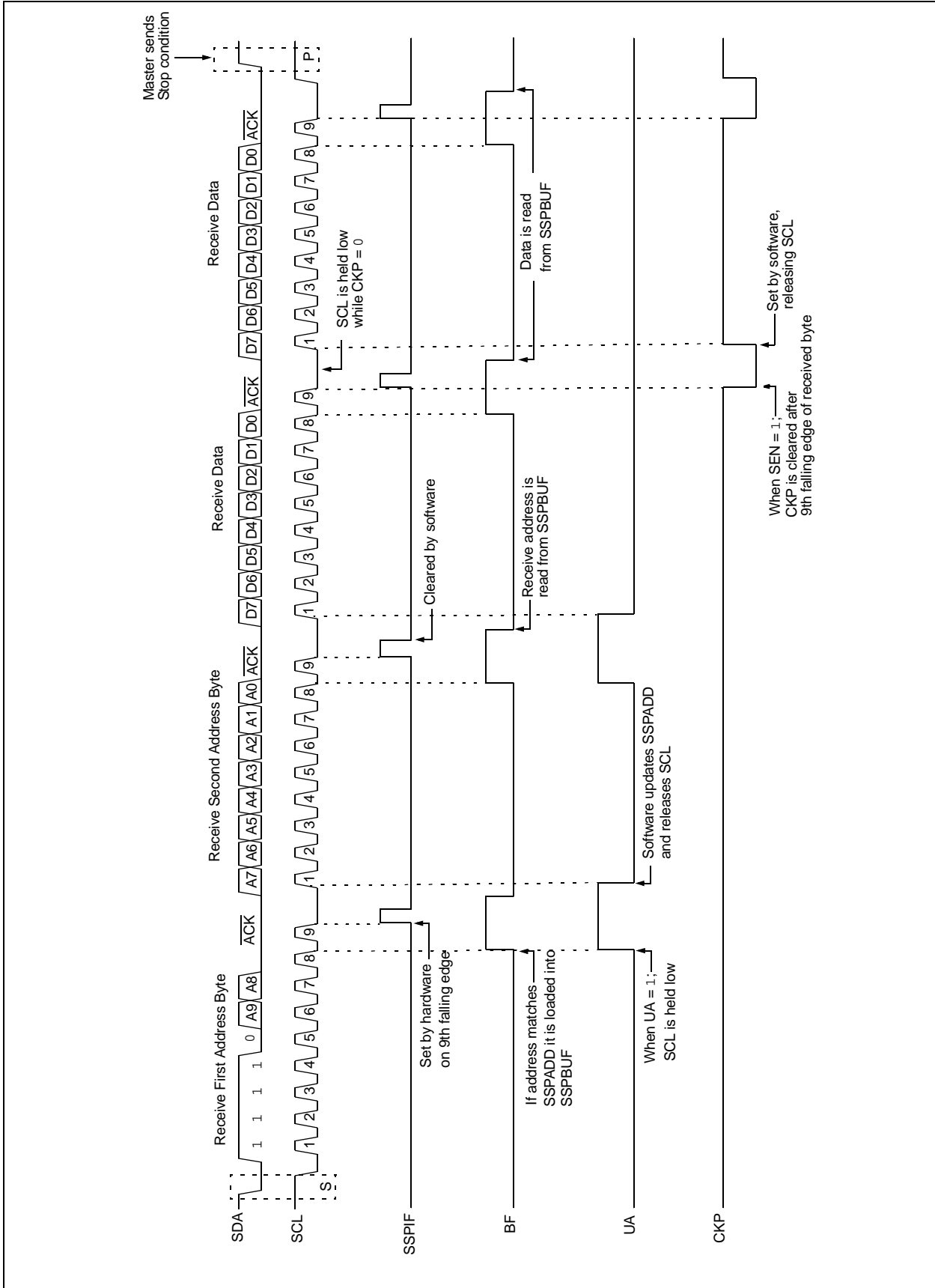
10. Slave clears SSPIF.
11. Slave reads the received matching address from SSPBUF clearing BF.
12. Slave loads high address into SSPADD.
13. Master clocks a data byte to the slave and clocks out the slaves  $\overline{ACK}$  on the 9th SCL pulse; SSPIF is set.
14. If SEN bit of SSPCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPIF.
16. Slave reads the received byte from SSPBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

## 22.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 22-17 can be used as a reference of a slave in 10-bit addressing with AHEN set.

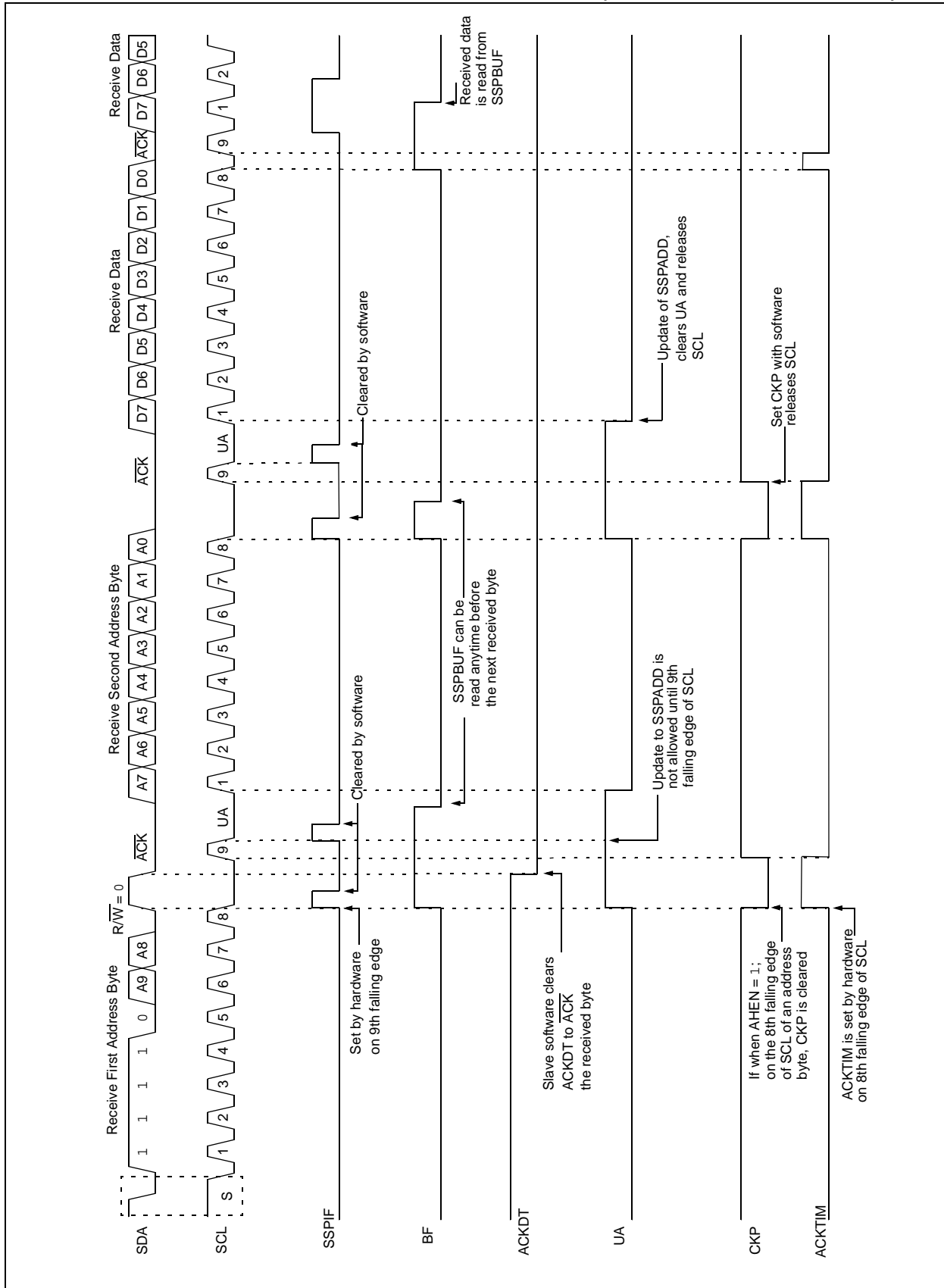
Figure 22-18 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

**FIGURE 22-17: I<sup>2</sup>C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**



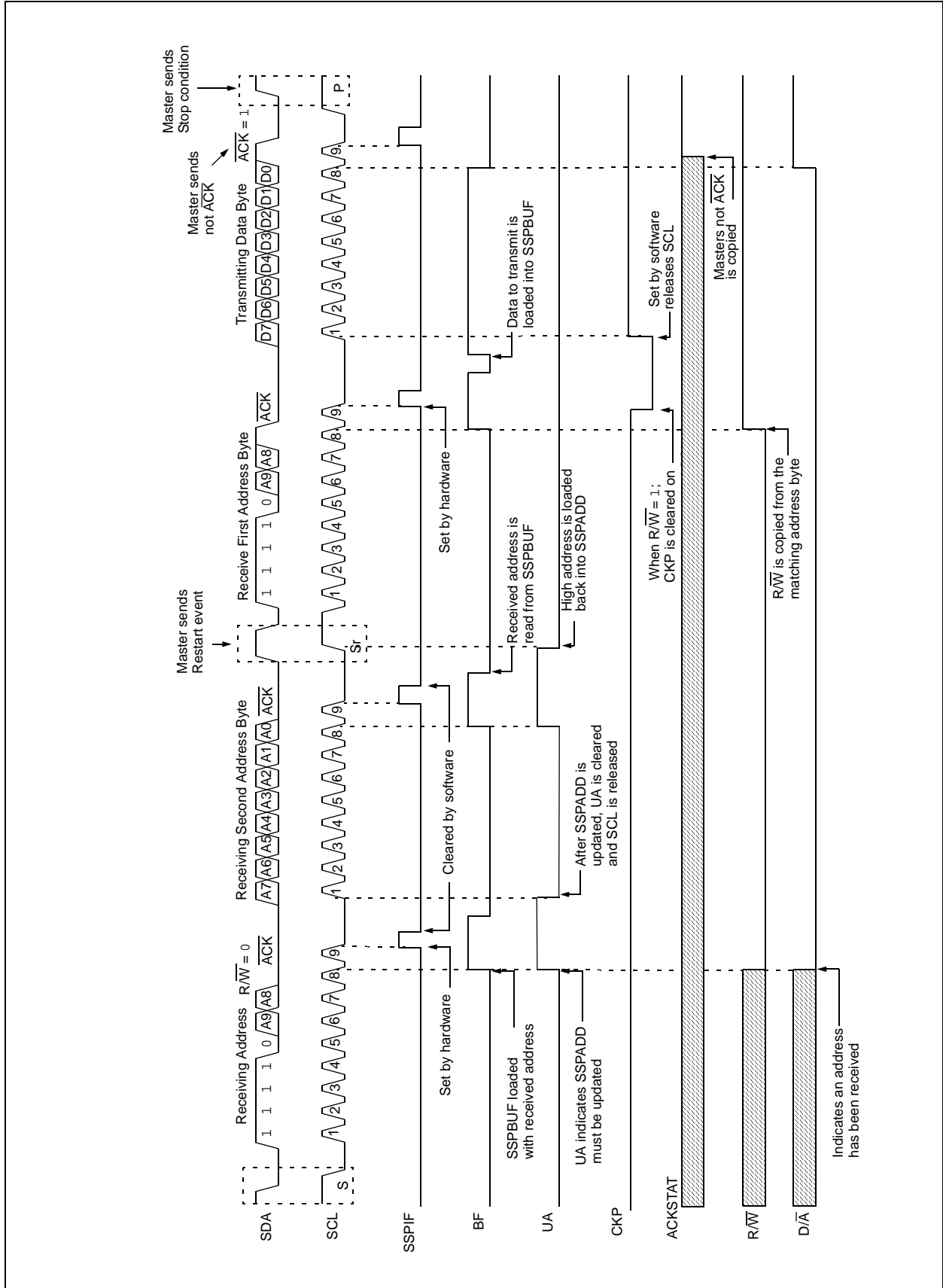
# PIC16F193X/LF193X

FIGURE 22-18: I<sup>2</sup>C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)





**FIGURE 22-19: I<sup>2</sup>C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)**



# PIC16F193X/LF193X

## 22.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

### 22.5.6.1 Normal Clock Stretching

Following an  $\overline{\text{ACK}}$  if the  $\overline{\text{R}\overline{\text{W}}}$  bit of SSPSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPBUF with data to transfer to the master. If the SEN bit of SSPCON2 is set, the slave hardware will always stretch the clock after the  $\overline{\text{ACK}}$  sequence. Once the slave is ready; CKP is set by software and communication resumes.

**Note 1:** The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPBUF was read before the 9th falling edge of SCL.

**2:** Previous versions of the module did not stretch the clock for a transmission if SSPBUF was loaded before the 9th falling edge of SCL. It is now always cleared for read requests.

### 22.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPADD.

**Note:** Previous versions of the module did not stretch the clock if the second address byte did not match.

### 22.5.6.3 Byte NACKing

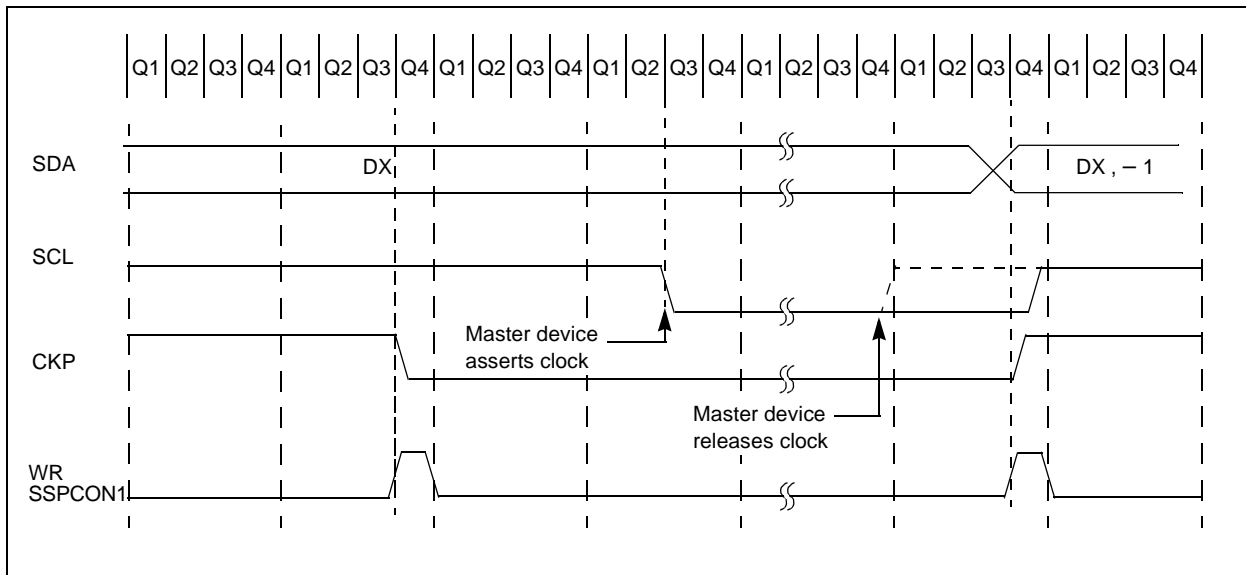
When AHEN bit of SSPCON3 is set; CKP is cleared by hardware after the 8th falling edge of SCL for a received matching address byte. When DHEN bit of SSPCON3 is set; CKP is cleared after the 8th falling edge of SCL for received data.

Stretching after the 8th falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

## 22.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing 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<sup>2</sup>C 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<sup>2</sup>C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 22-19).

**FIGURE 22-20: CLOCK SYNCHRONIZATION TIMING**



## 22.5.8 GENERAL CALL ADDRESS SUPPORT

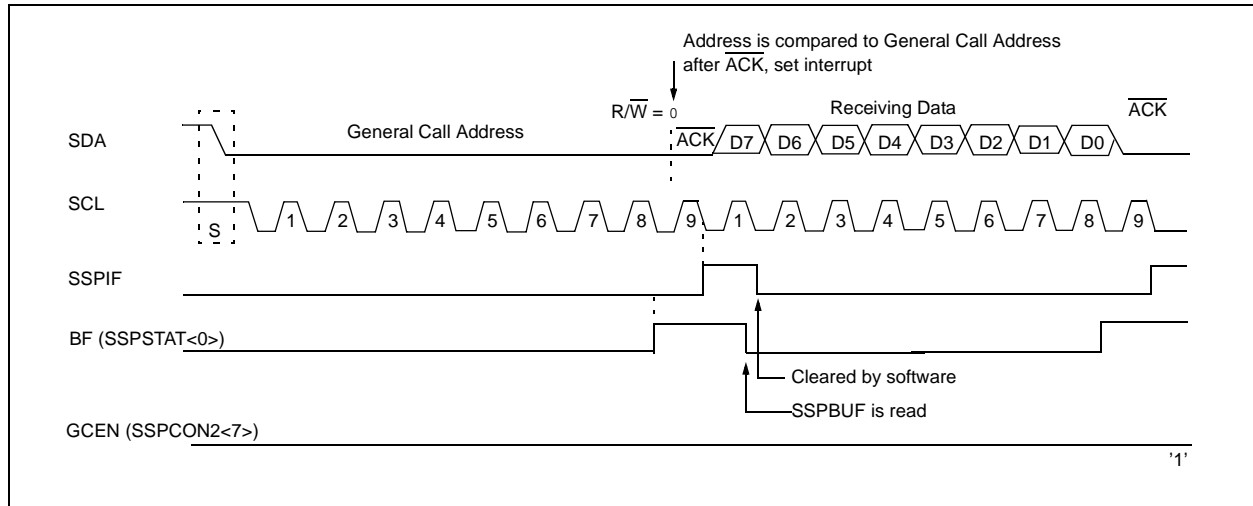
The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I<sup>2</sup>C protocol, defined as address 0x00. When the GCEN bit of the SSPCON2 register is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave software can read SSPBUF and respond. Figure 22-20 shows a General Call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the 8th falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

**FIGURE 22-21: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE**



## 22.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 22-5) is available in I<sup>2</sup>C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

# PIC16F193X/LF193X

## 22.6 I<sup>2</sup>C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPCON1 register and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are set as inputs and are manipulated by the MSSP hardware.

Master mode of operation is supported by 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 MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I<sup>2</sup>C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

**Note 1:** The MSSP module, when configured in I<sup>2</sup>C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur

- 2:** When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

### 22.6.1 I<sup>2</sup>C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

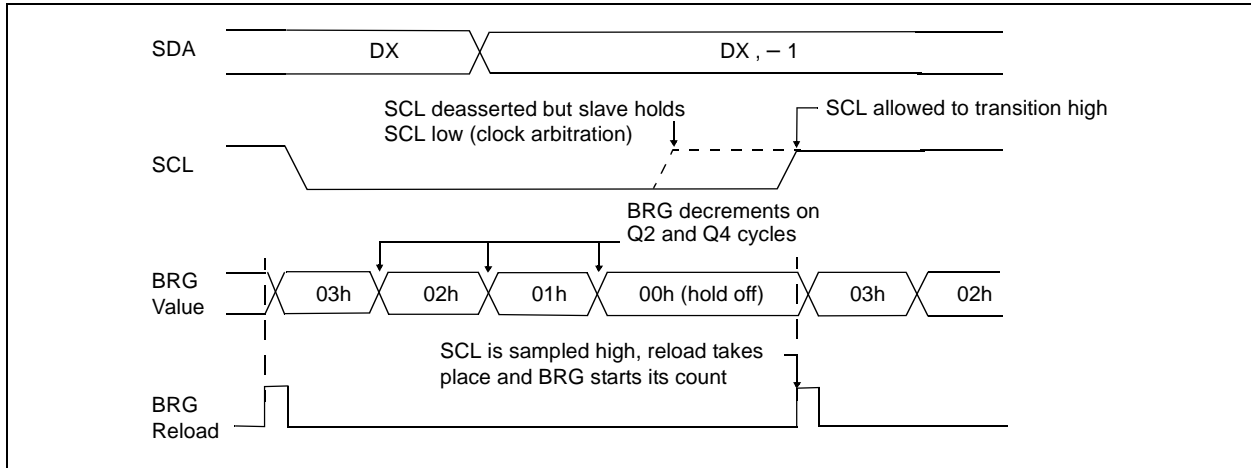
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 22.7 "Baud Rate Generator"** for more detail.

## 22.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 22-22).

**FIGURE 22-22: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



## 22.6.3 WCOL STATUS FLAG

If the user writes the SSPBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur). Any time the WCOL bit is set it indicates that an action on SSPBUF was attempted while the module was not Idle.

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

# PIC16F193X/LF193X

## 22.6.4 I<sup>2</sup>C MASTER MODE START CONDITION TIMING

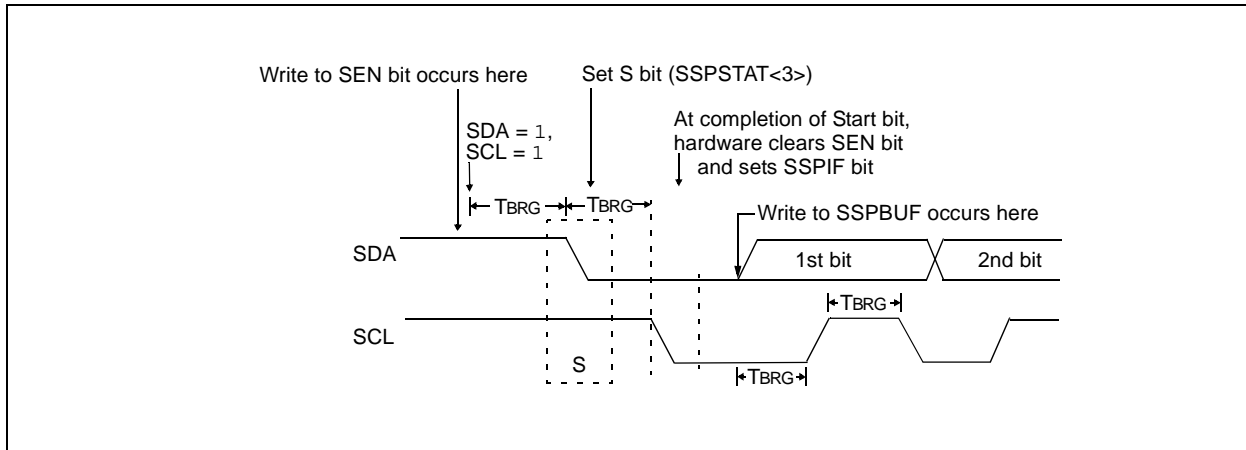
To initiate a Start condition, the user sets the Start Enable bit, SEN bit of the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPCON2 reg-

ister will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

**Note 1:** If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I<sup>2</sup>C module is reset into its Idle state.

**2:** The Philips I<sup>2</sup>C Specification states that a bus collision cannot occur on a Start.

**FIGURE 22-23: FIRST START BIT TIMING**



## 22.6.5 I<sup>2</sup>C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit of the SSPCON2 register is programmed high and the Master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the

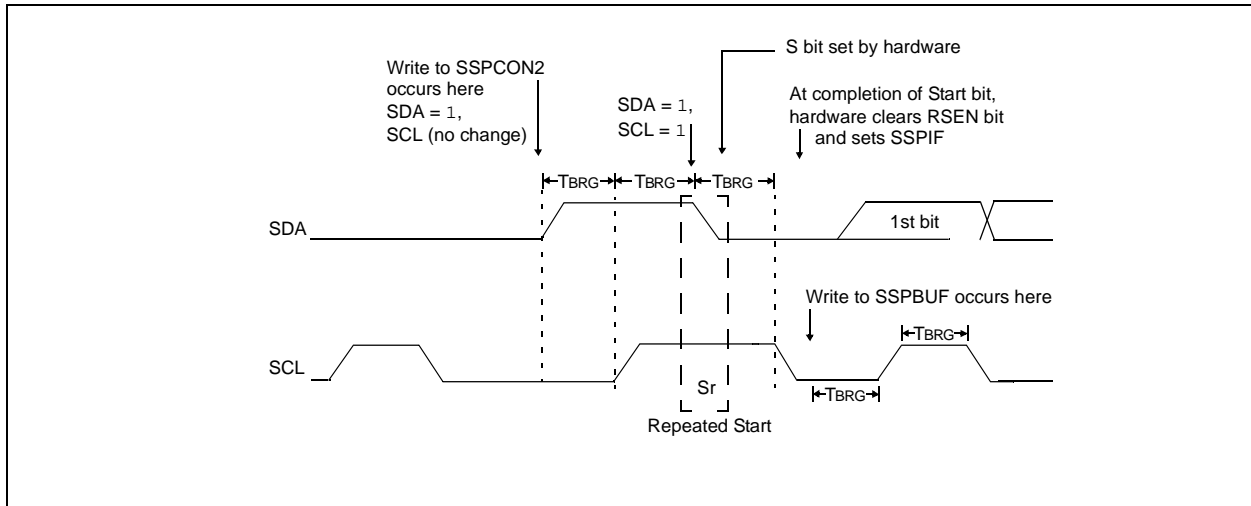
SSPCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPSTAT register will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

**FIGURE 22-24: REPEAT START CONDITION WAVEFORM**



# PIC16F193X/LF193X

## 22.6.6 I<sup>2</sup>C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time if an address match occurred, or if data was received properly. The status of  $\overline{\text{ACK}}$  is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 22-24).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the  $\overline{\text{ACK}}$  bit is loaded into the ACKSTAT Status bit of the SSPCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

### 22.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPSTAT register is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

### 22.6.6.2 WCOL Status Flag

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

WCOL must be cleared by software before the next transmission.

### 22.6.6.3 ACKSTAT Status Flag

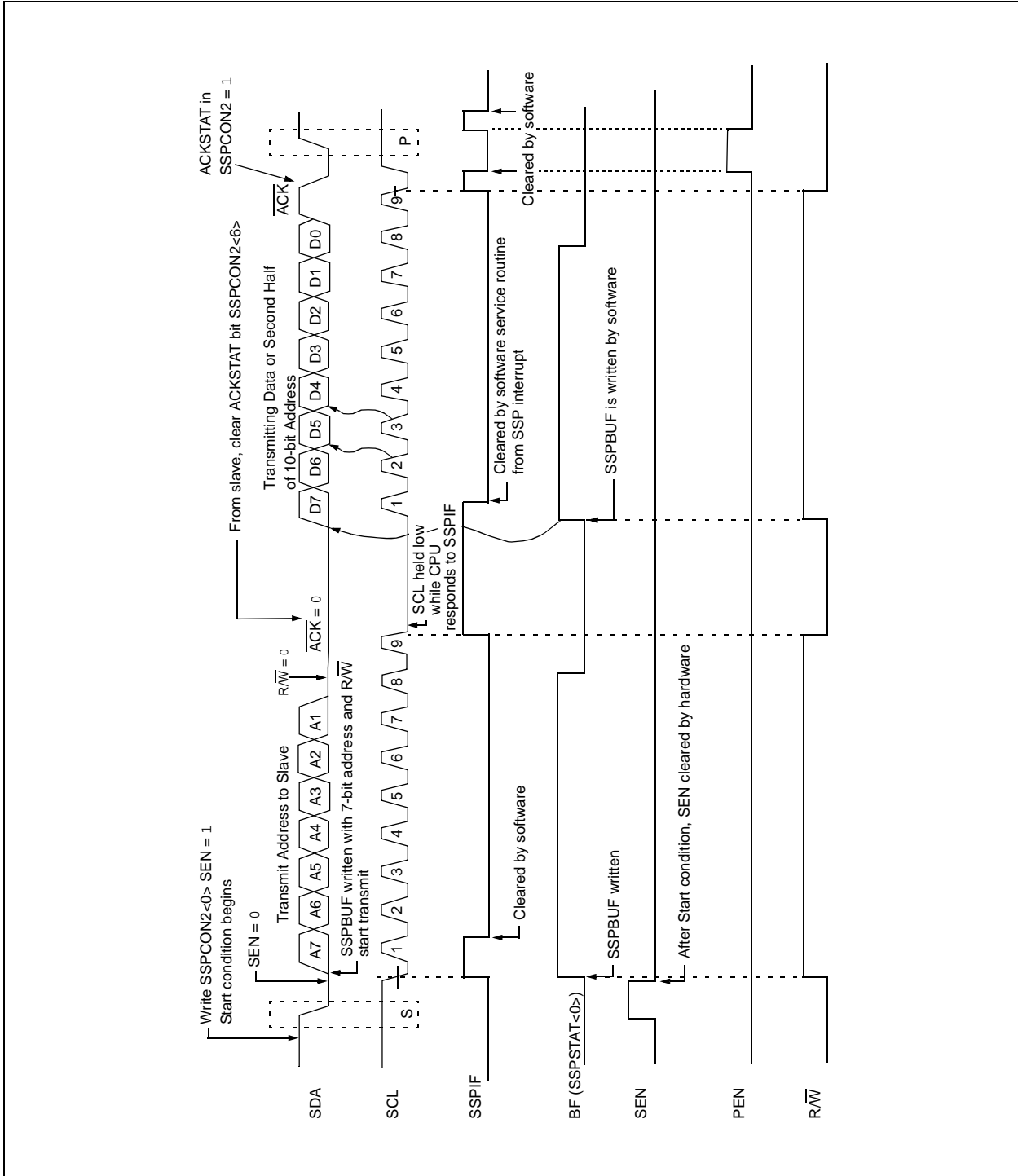
In Transmit mode, the ACKSTAT bit of the SSPCON2 register is cleared when the slave has sent an Acknowledge ( $\text{ACK} = 0$ ) and is set when the slave does not Acknowledge ( $\text{ACK} = 1$ ). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

### 22.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all 8 bits are transmitted. Transmission begins as soon as SSPBUF is written to.
7. The MSSP module shifts in the  $\overline{\text{ACK}}$  bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
9. The user loads the SSPBUF with eight bits of data.
10. Data is shifted out the SDA pin until all 8 bits are transmitted.
11. The MSSP module shifts in the  $\overline{\text{ACK}}$  bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.



**FIGURE 22-25: I<sup>2</sup>C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**



# PIC16F193X/LF193X

---

## 22.6.7 I<sup>2</sup>C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN bit of the SSPCON2 register.

**Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, ACKEN bit of the SSPCON2 register.

### 22.6.7.1 BF Status Flag

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

### 22.6.7.2 SSPOV Status Flag

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

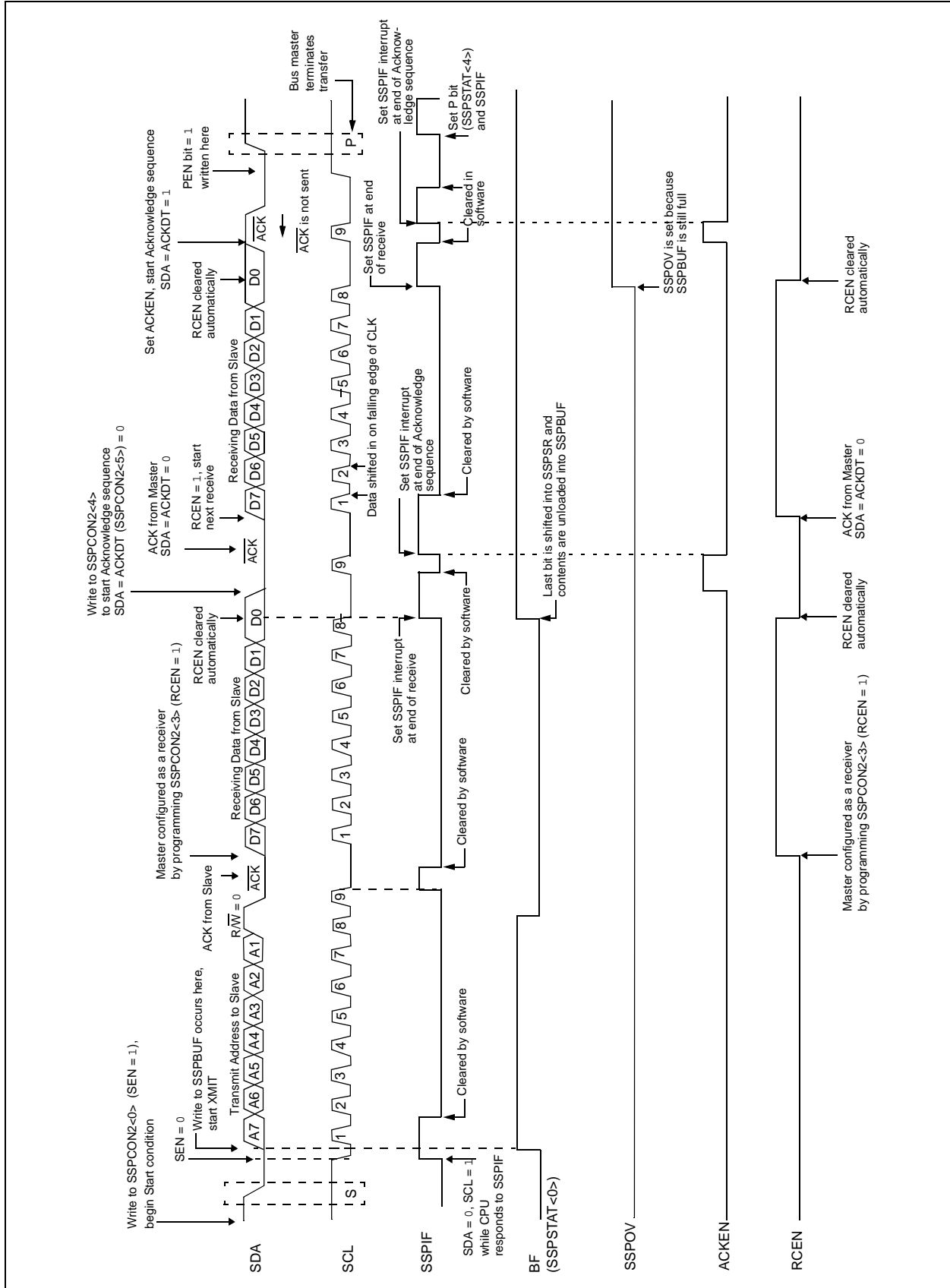
### 22.6.7.3 WCOL Status Flag

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

### 22.6.7.4 Typical Receive Sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. User writes SSPBUF with the slave address to transmit and the R/W bit set.
5. Address is shifted out the SDA pin until all 8 bits are transmitted. Transmission begins as soon as SSPBUF is written to.
6. The MSSP module shifts in the  $\overline{\text{ACK}}$  bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
7. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
8. User sets the RCEN bit of the SSPCON2 register and the Master clocks in a byte from the slave.
9. After the 8th falling edge of SCL, SSPIF and BF are set.
10. Master clears SSPIF and reads the received byte from SSPBUF, clears BF.
11. Master sets  $\overline{\text{ACK}}$  value sent to slave in ACKDT bit of the SSPCON2 register and initiates the  $\overline{\text{ACK}}$  by setting the ACKEN bit.
12. Masters  $\overline{\text{ACK}}$  is clocked out to the Slave and SSPIF is set.
13. User clears SSPIF.
14. Steps 8-13 are repeated for each received byte from the slave.
15. Master sends a not  $\overline{\text{ACK}}$  or Stop to end communication.

**FIGURE 22-26: I<sup>2</sup>C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**



# PIC16F193X/LF193X

## 22.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 22-26).

### 22.6.8.1 WCOL Status Flag

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

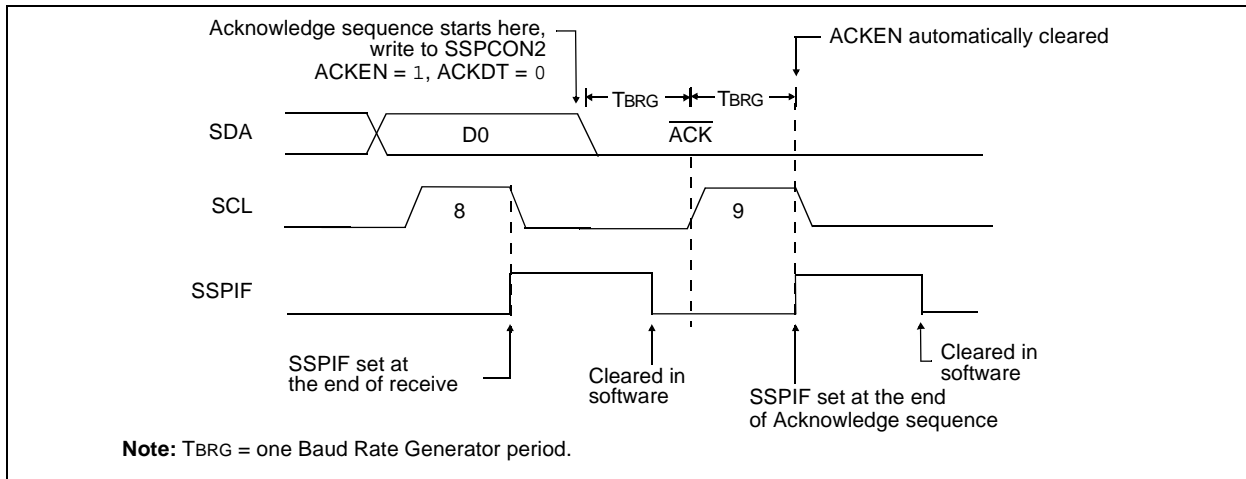
## 22.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 22-27).

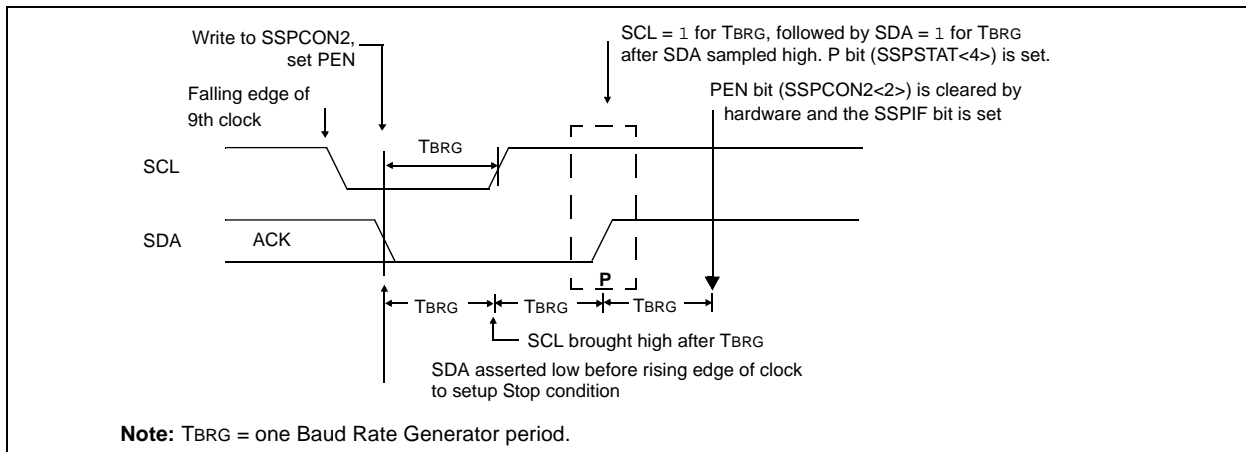
### 22.6.9.1 WCOL Status Flag

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

**FIGURE 22-27: ACKNOWLEDGE SEQUENCE WAVEFORM**



**FIGURE 22-28: STOP CONDITION RECEIVE OR TRANSMIT MODE**



## 22.6.10 SLEEP OPERATION

While in Sleep mode, the I<sup>2</sup>C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

## 22.6.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

## 22.6.12 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 MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit of the SSPSTAT register is set, or the bus is Idle, with 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 for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

## 22.6.13 MULTI-MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I<sup>2</sup>C port to its Idle state (Figure 22-28).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

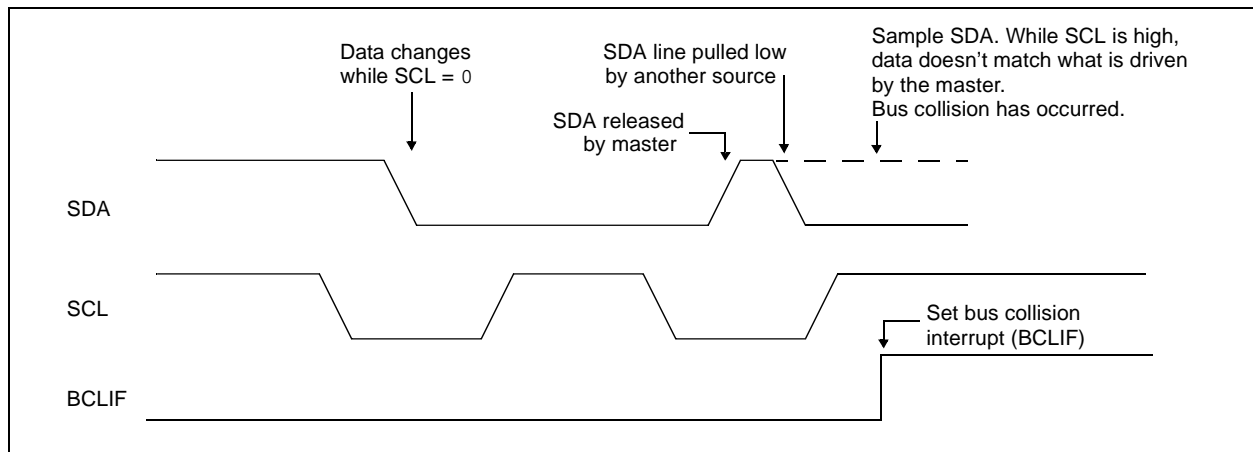
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I<sup>2</sup>C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

**FIGURE 22-29: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



# PIC16F193X/LF193X

## 22.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 22-29).
- b) SCL is sampled low before SDA is asserted low (Figure 22-30).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

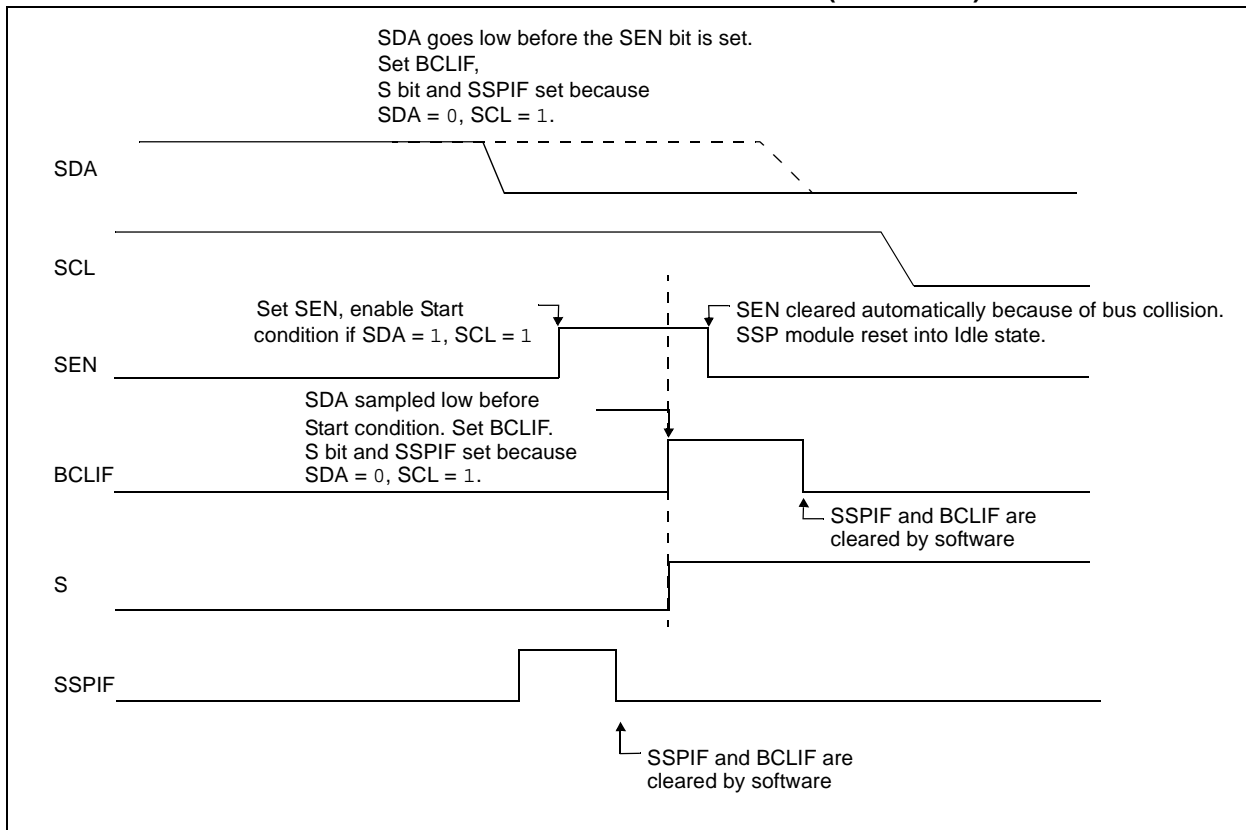
- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 22-29).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

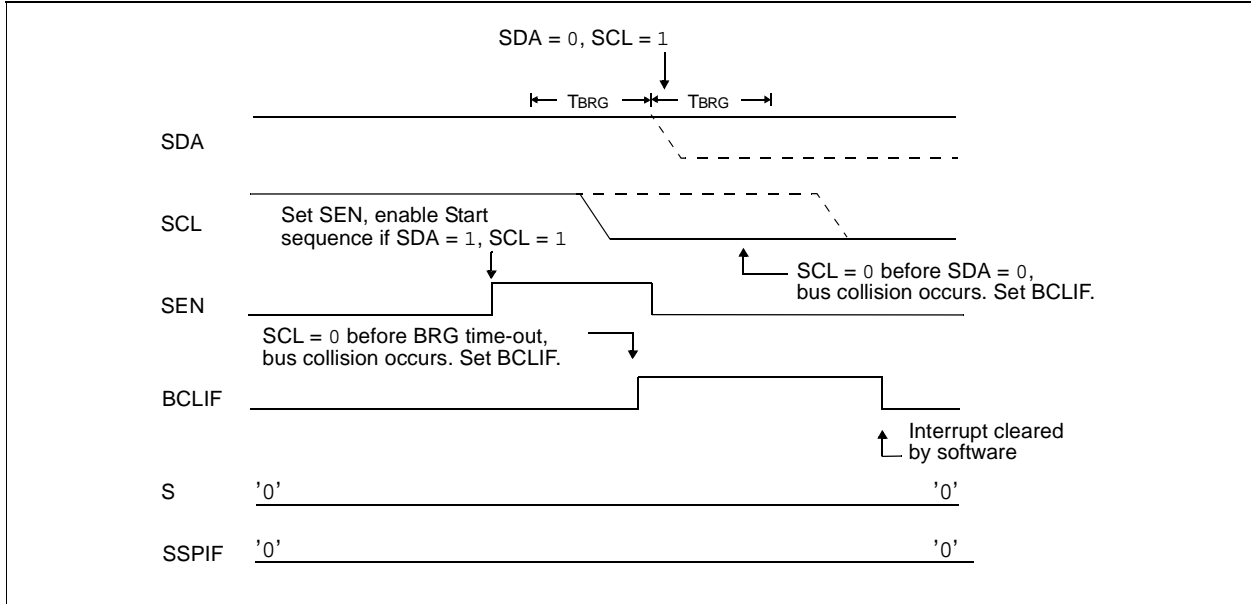
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 22-31). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

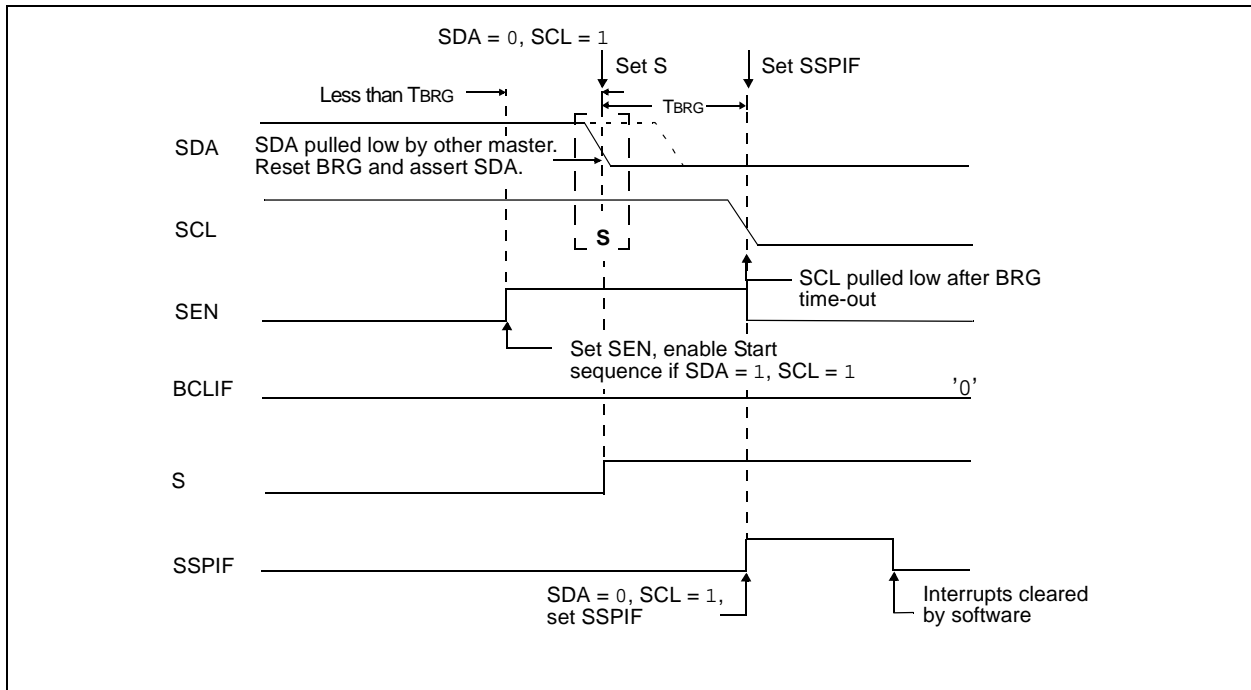
**FIGURE 22-30: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**FIGURE 22-31: BUS COLLISION DURING START CONDITION (SCL = 0)**



**FIGURE 22-32: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



# PIC16F193X/LF193X

## 22.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

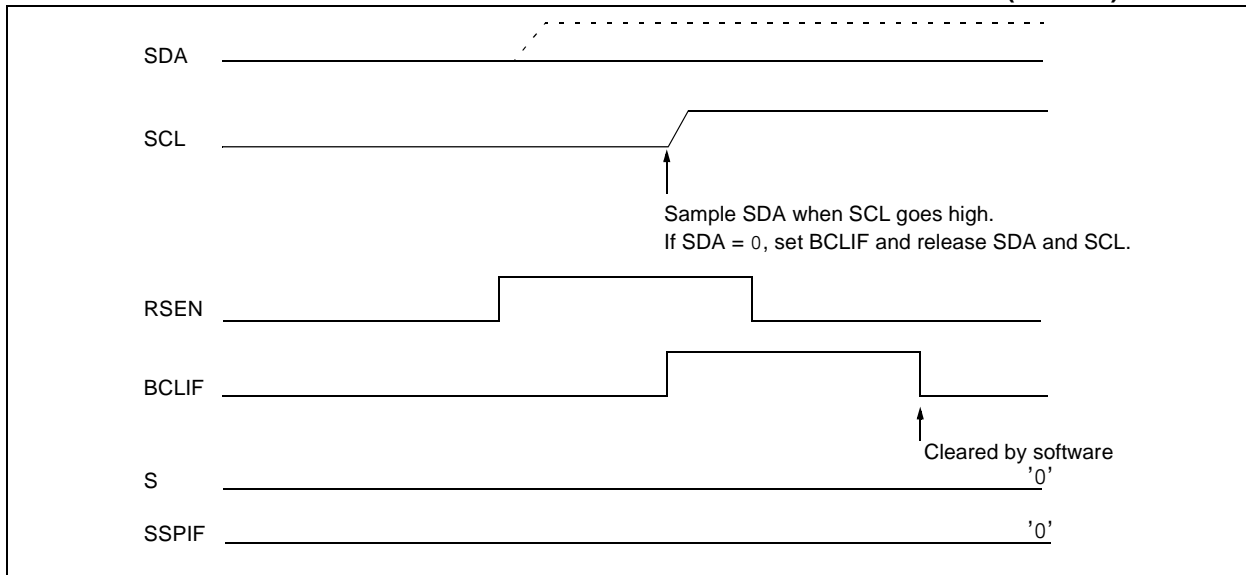
When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 22-32). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

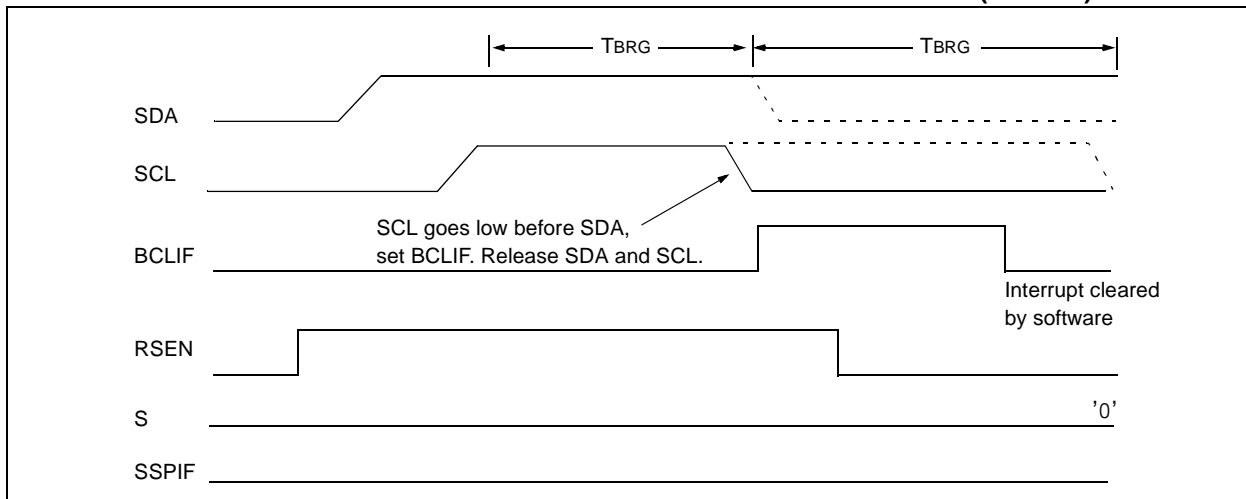
If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 22-33.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is released and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

**FIGURE 22-33: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**



**FIGURE 22-34: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**





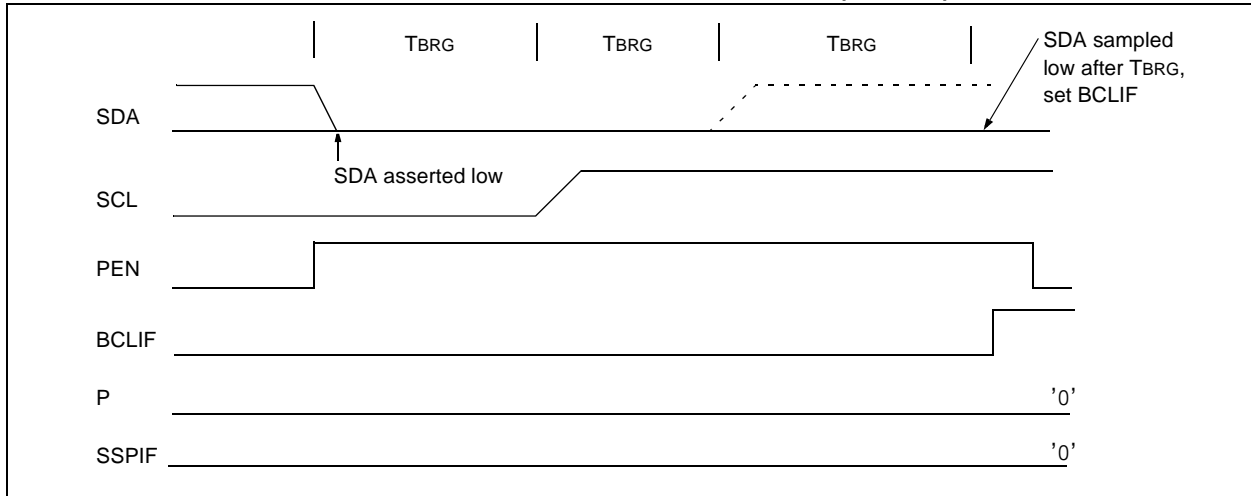
## 22.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

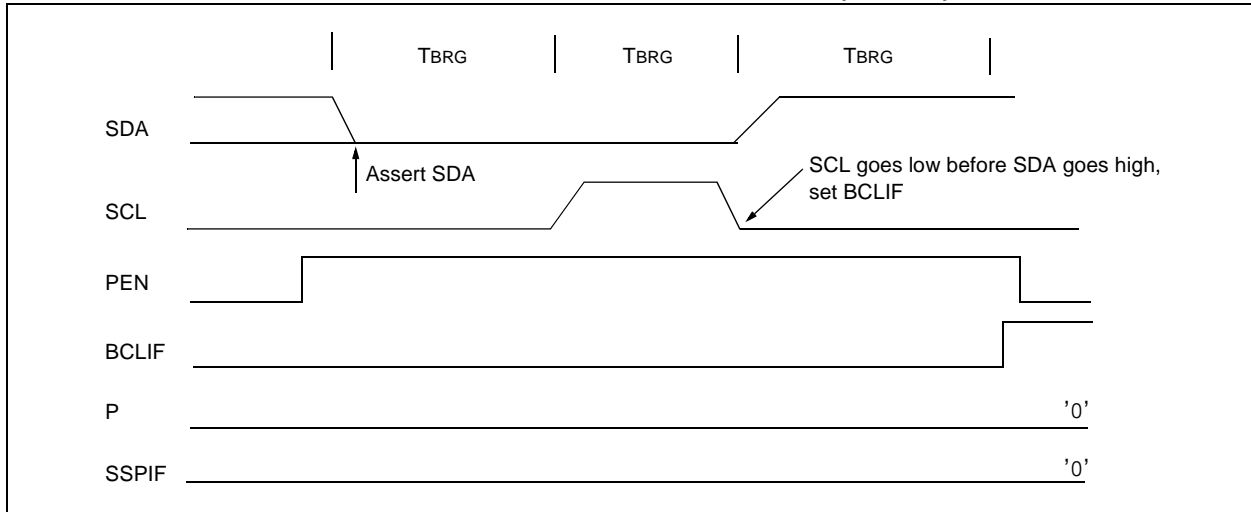
- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 22-34). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 22-35).

**FIGURE 22-35: BUS COLLISION DURING A STOP CONDITION (CASE 1)**



**FIGURE 22-36: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



# PIC16F193X/LF193X

## 22.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both I<sup>2</sup>C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPADD register (Register 22-6). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

An internal signal “Reload” in Figure 22-36 triggers the value from SSPADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module

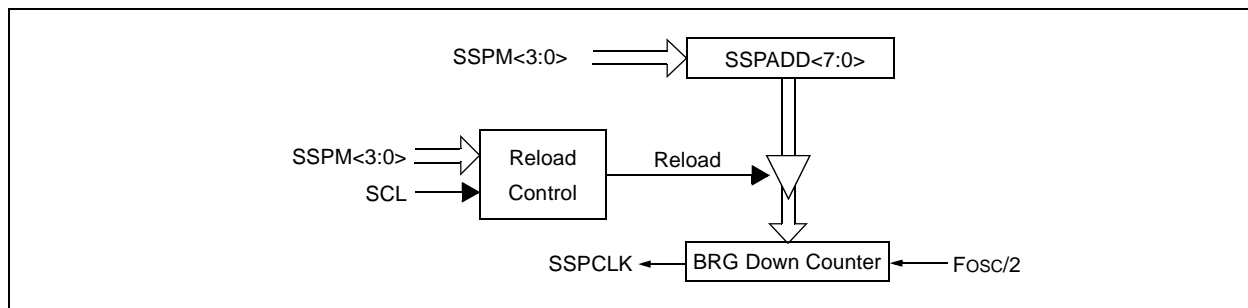
clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 22-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

### EQUATION 22-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPADD + 1)(4)}$$

**FIGURE 22-37: BAUD RATE GENERATOR BLOCK DIAGRAM**



**Note:** Values of 0x00, 0x01 and 0x02 are not valid for SSPADD when used as a Baud Rate Generator for I<sup>2</sup>C. This is an implementation limitation.

**TABLE 22-3: MSSP CLOCK RATE W/BRG**

Fosc	Fcy	BRG Value	Fclock (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz <sup>(1)</sup>
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz <sup>(1)</sup>
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz
4 MHz	1 MHz	00h	250 kHz <sup>(2)</sup>

**Note 1:** The I<sup>2</sup>C interface does not conform to the 400 kHz I<sup>2</sup>C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

**2:** SPI mode only.

## 23.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

The Data EEPROM and Flash program memory are readable and writable 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 (SFRs). 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 EEDATL location being accessed. These devices have 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When accessing the program memory block of the PIC16F1936/PIC16F1937 devices, the EEDATL and EEDATH registers form a 2-byte word that holds the 14-bit data for read/write, and the EEADRL and EEADRH registers form a 2-byte word that holds the 15-bit address of the program memory location being read.

The EEPROM data memory allows byte read and write. An EEPROM 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.

Depending on the setting of the Flash Program Memory Self Write Enable bits WRT<1:0> of the Configuration Word Register 2, the device may or may not be able to write certain blocks of the program memory. However, reads from the program memory are always allowed.

When the device is code-protected, the device programmer can no longer access data or program memory. When code-protected, the CPU may continue to read and write the data EEPROM memory and Flash program memory.

## 23.1 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 32K words of program memory.

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 EEPROM address value, only the LSB of the address is written to the EEADRL register.

### 23.1.1 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, any subsequent operations will operate on the EEPROM memory. When set, any subsequent operations will operate on the program memory. On Reset, EEPROM is selected by default.

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 occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

Interrupt flag bit EEIF of the PIR2 register is set when write is complete. It must be cleared in the software.

Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence. To enable writes, a specific pattern must be written to EECON2.

# PIC16F193X/LF193X

## REGISTER 23-1: EEDATL: EEPROM DATA REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0
bit 7							bit 0

### Legend:

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

bit 7-0                      **EEDATL<7:0>**: 8 Least Significant data bits of data EEPROM or Read from program memory

## REGISTER 23-2: EEDATH: EEPROM DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 0

### Legend:

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

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

bit 5-0                      **EEDATH<5:0>**: 6 Most Significant Data bits from program memory

## REGISTER 23-3: EEADRL: EEPROM ADDRESS REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

### Legend:

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

bit 7-0                      **EEADRL<7:0>**: 8 Least Significant Address bits for EEPROM or program memory

## REGISTER 23-4: EEADRH: EEPROM ADDRESS HIGH BYTE REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	EEADRH6	EEADRH5	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0
bit 7							bit 0

### Legend:

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

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

bit 6-0                      **EEADRH<6:0>**: Specifies the 7 Most Significant Address bits or high bits for program memory reads

# PIC16F193X/LF193X

## REGISTER 23-5: EECON1: EEPROM CONTROL 1 REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7						bit 0	

### Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = bit is cleared by hardware

- bit 7      **EEPGD:** Flash Program/Data EEPROM Memory Select bit  
 1 = Accesses program space Flash memory  
 0 = Accesses data EEPROM memory
- bit 6      **CFGS:** Flash Program/Data EEPROM or Configuration Select bit  
 1 = Accesses Configuration, User ID and Device ID Registers  
 0 = Accesses Flash Program or data EEPROM Memory
- bit 5      **LWLO:** Load Write Latches Only bit  
If **EEPGD** = 1 or **CFGS** = 1: (accessing program Flash)  
 1 = The next WR command does not initiate a write to the PFM; only the program memory latches are updated.  
 0 = The next WR command writes a value from EEDATH:EEDATL into program memory latches and initiates a write to the PFM of all the data stored in the program memory latches.  
If **EEPGD** = 0 and **CFGS** = 1: (Accessing data EEPROM)  
 LWLO is ignored. The next WR command initiates a write to the data EEPROM.
- bit 4      **FREE:** Program Flash Erase Enable bit  
If **EEPGD** = 1 or **CFGS** = 1: (accessing program Flash)  
 1 = Perform a program Flash erase operation on the next WR command (cleared by hardware after completion of erase).  
 0 = Perform a program Flash write operation on the next WR command.  
If **EEPGD** = 0 and **CFGS** = 0: (Accessing data EEPROM)  
 FREE is ignored. The next WR command will initiate both a erase cycle and a write cycle.
- bit 3      **WRERR:** EEPROM Error Flag bit  
 1 = Condition could indicate an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write '1') of the WR bit.  
 0 = The program or erase operation completed normally.
- bit 2      **WREN:** Program/Erase Enable bit  
 1 = Allows program/erase cycles  
 0 = Inhibits programming/erasing of program Flash and data EEPROM
- bit 1      **WR:** Write Control bit  
 1 = Initiates a program Flash or data EEPROM program/erase operation.  
 The operation is self-timed and the bit is cleared by hardware once operation is complete.  
 The WR bit can only be set (not cleared) in software.  
 0 = Program/erase operation to the Flash or data EEPROM is complete and inactive.
- bit 0      **RD:** Read Control bit  
 1 = Initiates an program Flash or data EEPROM read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.  
 0 = Does not initiate a program Flash or data EEPROM data read.

# PIC16F193X/LF193X

---

## REGISTER 23-6: EECON2: EEPROM CONTROL 2 REGISTER

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
EEUNLK7	EEUNLK6	EEUNLK5	EEUNLK4	EEUNLK3	EEUNLK2	EEUNLK1	EEUNLK0
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

S = Bit can only be set

x = Bit is unknown

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

'1' = Bit is set

'0' = Bit is cleared

bit 7-0

**EEUNLK<7:0>**: Data EEPROM Unlock Pattern bits

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the EECON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes. Refer to **Section 23.1.3 "Writing to the Data EEPROM Memory"** for more information.

## 23.1.2 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADRL register, clear the EEPGD and CFGS control bits of the EECON1 register, and then set control bit RD. The data is available at 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 23-1: DATA EEPROM READ

```
BANKSEL EEADRL      ;
MOVLW  DATA_EE_ADDR ;
MOVWF  EEADRL      ;Data Memory
                ;Address to read
BCF    EECON1, CFGS ;Deselect Config space
BCF    EECON1, EEPGD;Point to DATA memory
BSF    EECON1, RD   ;EE Read
MOVWF  EEDATL, W   ;W = EEDATL
BCF    STATUS, RP1 ;Bank 0
```

**Note:** Data EEPROM can be read regardless of the setting of the CPD bit.

## 23.1.3 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 above sequence 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.

### EXAMPLE 23-2: DATA EEPROM WRITE

```
BANKSEL EEADRL      ;
MOVLW  DATA_EE_ADDR ;
MOVWF  EEADRL      ;Data Memory Address to write
MOVLW  DATA_EE_DATA ;
MOVWF  EEDATL      ;Data Memory Value to write
BCF    EECON1, CFGS ;Deselect Configuration space
BCF    EECON1, EEPGD;Point to DATA memory
BSF    EECON1, WREN ;Enable writes

BCF    INTCON, GIE  ;Disable INTs.
BTFSC  INTCON, GIE  ;SEE AN576
GOTO   $-2
MOVLW  55h          ;
MOVWF  EECON2      ;Write 55h
MOVLW  AAh          ;
MOVWF  EECON2      ;Write AAh
BSF    EECON1, WR   ;Set WR bit to begin write

BCF    EECON1, WREN ;Disable writes
BTFSC  EECON1, WR   ;Wait for write to complete
GOTO   $-2          ;Done
```

Required Sequence

# PIC16F193X/LF193X

## 23.1.4 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

1. Write the Least and Most Significant address bits to the EEADRL and EEADRH registers.
2. Clear the CFGS bit of the EECON1 register.
3. Set the EEPGD control bit of the EECON1 register.
4. Then, set control bit RD of the EECON1 register.

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.

**Note 1:** The two instructions following a program memory read are required to be NOPS. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

**2:** Data EEPROM can be read regardless of the setting of the CPD bit.

### EXAMPLE 23-3: FLASH PROGRAM READ

```

BANKSEL EEADRL           ;
MOVLW  MS_PROG_EE_ADDR   ;
MOVWF  EEADRH           ;MS Byte of Program Address to read
MOVLW  LS_PROG_EE_ADDR   ;
MOVWF  EEADRL           ;LS Byte of Program Address to read
BANKSEL EECON1           ;
BSF    EECON1, EEPGD     ;Point to PROGRAM memory
BSF    EECON1, RD        ;EE Read
                                ;First instruction after BSF EECON1,RD executes normally
NOP
NOP                                ;Any instructions here are ignored as program
                                ;memory is read in second cycle after BSF EECON1,RD
;
BANKSEL EEDATL           ;
MOVF   EEDATL, W         ;W = LS Byte of Program Memory
MOVWF  LOWPMBYTE         ;
MOVF   EEDATH, W         ;W = MS Byte of Program EEDATL
MOVWF  HIGHPMBYTE       ;
BCF    STATUS, RP1      ;Bank 0
```



## EXAMPLE 23-4: FLASH PROGRAM MEMORY READ

```

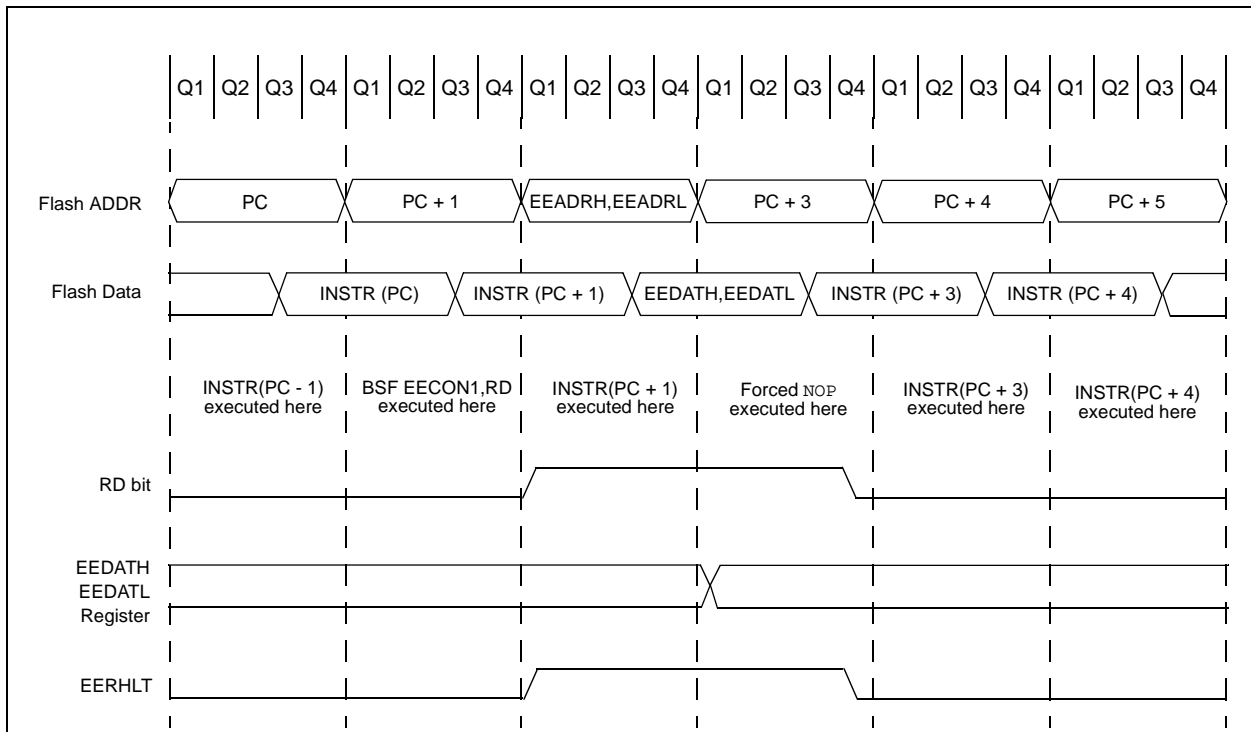
* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI : PROG_ADDR_LO
* data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO

  BANKSEL  EEADRL          ; Select Bank for EEPROM registers
  MOVLW    PROG_ADDR_LO    ;
  MOVWF    EEADRL          ; Store LSB of address
  MOVLW    PROG_ADDR_HI    ;
  MOVWF    EEADRH         ; Store MSB of address

  BCF      EECON1,CFG5      ; Select Configuration Space
  BSF      EECON1,EEPGD     ; Select Program Memory
  BCF      INTCON,GIE       ; Disable interrupts
  BSF      EECON1,RD        ; Initiate read
  NOP      ; Executed (Figure 23-1)
  NOP      ; Ignored (Figure 23-1)
  BSF      INTCON,GIE       ; Restore interrupts

  MOVF     EEDATL,W         ; Get LSB of word
  MOVWF    PROG_DATA_LO    ; Store in user location
  MOVF     EEDATH,W         ; Get MSB of word
  MOVWF    PROG_DATA_HI    ; Store in user location
  
```

**FIGURE 23-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION**



# PIC16F193X/LF193X

---

## 23.2 Erasing Program Memory

While executing code, program memory can only be erased by rows. A row consists of 32 words where the  $EEADR\langle L \rangle = 0000$ . To erase a row:

1. Load the  $EEADRH$  and  $EEADRL$  registers with the address of new row to be erased.
2. Clear the  $CFG\langle S \rangle$  bit of the  $EECON1$  register.
3. Set the  $EEP\langle GD \rangle$  bit of the  $EECON1$  register.
4. Set the  $FREE$  bit of the  $EECON1$  register.
5. Write 55h, then AAh, to  $EECON2$  (Flash programming unlock sequence).
6. Set control bit  $WR$  of the  $EECON1$  register to begin the write operation.

## 23.3 Writing to Flash Program Memory

Before writing, program memory should be erased using the Erase Program Memory command.

No automatic erase occurs upon the initiation of the write; if the program Flash needs to be erased before writing, the row (32 words) must be erased previously.

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits  $WRT\langle 1:0 \rangle$  of the Configuration Word Register 2. Flash program memory must be written in eight-word blocks. See Figure 23-2 for more details. A block consists of eight words with sequential addresses, with a lower boundary defined by an address, where  $EEADR\langle L \rangle = 000$ . All block writes to program memory are done as 32-word erase by eight-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

When the  $LWLO$  bit is '1', the write sequence will only load the buffer register and will not actually initiate the write to program Flash:

1. Set the  $EEP\langle GD \rangle$ ,  $WREN$  and  $LWLO$  bits of the  $EECON1$  register.
2. Write 55h, then AAh, to  $EECON2$  (Flash programming unlock sequence).
3. Set control bit  $WR$  of the  $EECON1$  register to begin the write operation.

To write program data, it must first be loaded into the buffer registers (see Figure 23-1). This is accomplished by first writing the destination address to  $EEADRL$  and  $EEADRH$  and then writing the data to  $EEDATA$  and  $EEDATH$ . After the address and data have been set up, then the following sequence of events must be executed:

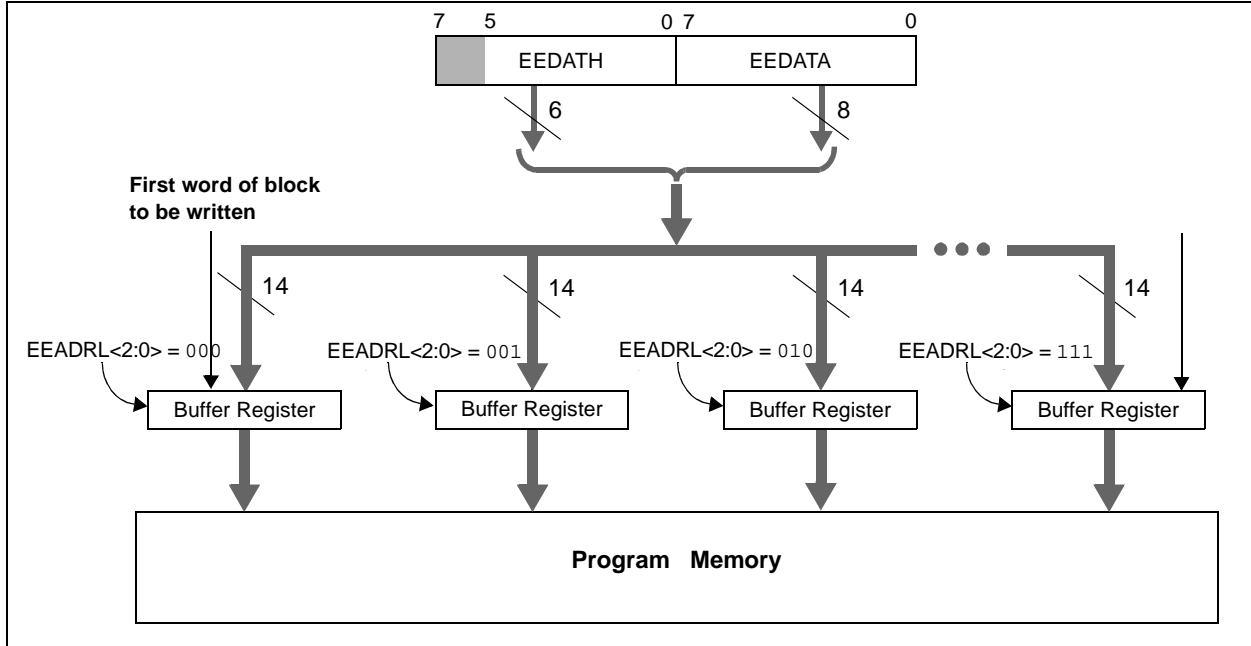
1. Set the  $EEP\langle GD \rangle$  control bit of the  $EECON1$  register.
2. Set the  $LWLO$  bit of the  $EECON1$  register.
3. Write 55h, then AAh, to  $EECON2$  (Flash programming sequence).
4. Set the  $WR$  control bit of the  $EECON1$  register.

Up to eight buffer register locations can be written to with correct data. If less than eight words are being written to in the block of eight words, then the data for the unprogrammed words should be set to all ones.

After the "BSF  $EECON1, WR$ " instruction, the processor requires two cycles to set up the erase/write operation. The user must place two  $NOP$  instructions after the  $WR$  bit is set. Since data is being written to buffer registers, the writing of the first seven words of the block appears to occur immediately. The processor will halt internal operations for the typical 2 ms, only during the cycle in which the erase takes place (i.e., the last word of the sixteen-word block erase). This is not Sleep mode as the clocks and peripherals will continue to run. After the eight-word write cycle, the processor will resume operation with the third instruction after the  $EECON1$  write instruction.

An example of the complete eight-word write sequence is shown in Example 23-5. The initial address is loaded into the  $EEADRH$  and  $EEADRL$  register pair; the eight words of data are loaded using indirect addressing.

**FIGURE 23-2: BLOCK WRITES TO 8K FLASH PROGRAM MEMORY**



# PIC16F193X/LF193X

## EXAMPLE 23-5: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
; 1. A valid starting address (the least significant bits = 00) is loaded in ADDRH:ADDRL
; 2. The 8 bytes of data are loaded, starting at the address in DATADDR
; 3. ADDRH, ADDRL and DATADDR are all located in shared data memory 0x70 - 0x7f
;
      BANKSEL EEADRH          ; Bank 3
      MOVF   ADDRH,W          ; Load initial address
      MOVWF  EEADRH          ;
      MOVF   ADDRL,W         ;
      MOVWF  EEADRL         ;
      MOVF   DATAADDRL,W    ; Load initial data address
      MOVWF  FSR0L          ;
      MOVF   DATAADDRH,W    ; Load initial data address
      MOVWF  FSR0H          ;

LOOP  MOVIW  INDF0++         ; Load first data byte into lower
      MOVWF  EEDATL         ;
      MOVIW  INDF0++         ; Load second data byte into upper
      MOVWF  EEDATH         ;
      BSF   EECON1,EEPGD    ; Point to program memory
      BCF   EECON1,CFG5     ; Not configuration space
      BSF   EECON1,WREN     ; Enable writes
      BSF   EECON1,LWLO     ; Only Load Write Latches

      MOVLW  55h            ; Start of required write sequence:
      MOVWF  EECON2         ; Write 55h
      MOVLW  AAh            ;
      MOVWF  EECON2         ; Write AAh
      BSF   EECON1,WR       ; Set WR bit to begin write

      NOP                    ; Any instructions here are ignored as processor
                          ; halts to begin write sequence
      NOP                    ; processor will stop here and wait for write complete
                          ; after write processor continues with 3rd instruction

      MOVF   EEADR,W        ; Check if lower two bits of address are '00'
      XORLW  0x08           ; Check if we're on the last of 8 addresses
      ANDLW  0x08           ;
      BTFSC  STATUS,Z       ; Exit if last of eight words,
      GOTO   START_WRITE    ;

      INCF   EEADR,F        ; Still loading latches Increment address
      GOTO   LOOP           ; Write next latches

START_WRITE
      BCF   EECON1,LWLO     ; No more Latches only; Actually start write

      MOVLW  55h            ; Start of required write sequence:
      MOVWF  EECON2         ; Write 55h
      MOVLW  AAh            ;
      MOVWF  EECON2         ; Write AAh
      BSF   EECON1,WR       ; Set WR bit to begin write

      NOP                    ; Any instructions here are ignored as processor
                          ; halts to begin write sequence
      NOP                    ; processor will stop here and wait for write complete
                          ; after write processor continues with 3rd instruction
      BCF   EECON1,WREN     ; Disable writes
```

## 23.4 Configuration Word and Device ID Access

Instead of accessing program memory or EEPROM data memory, the User ID's, Device ID/Revision ID and Configuration Words can be accessed when  $CFG5 = 1$ . This is the region that would be pointed to by  $PC<15> = 1$ , but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 23-1.

When read access is initiated on an unallowed address, the EEDATH:EEDATL registers are cleared.

Writes can be disabled via the WRT Configuration bits. Refer to the Configuration Word 2 register.

**TABLE 23-1: PFM AND FUSE ACCESS VIA EECON1/EEDATH:EEDATL REGISTERS (WHEN  $CFG5 = 1$ )**

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8006h	Device ID/Revision ID	Yes	No
8007h-8008h	Configuration Words 1 and 2	Yes	No

### EXAMPLE 23-3: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI : PROG_ADDR_LO
* data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO

  BANKSEL  EEADRL          ; Select Bank 2
  MOVLW    PROG_ADDR_LO   ;
  MOVWF    EEADRL         ; Store LSB of address
  MOVLW    PROG_ADDR_HI   ;
  MOVWL    EEADRH         ; Store MSB of address

  BCF      EECON1,CFG5    ; Deselect Configuration Space
  BSF      EECON1,EEPGD   ; Select Program Memory
  BCF      INTCON,GIE     ; Disable interrupts
  BSF      EECON1,RD      ; Initiate read
  NOP      ; Executed (Figure 23-1)
  NOP      ; Ignored (Figure 23-1)
  BSF      INTCON,GIE     ; Restore interrupts

  MOVF     EEDATL,W       ; Get LSB of word
  MOVWF    PROG_DATA_LO   ; Store in user location
  MOVF     EEDATH,W       ; Get MSB of word
  MOVWF    PROG_DATA_HI   ; Store in user location
```

# PIC16F193X/LF193X

## 23.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM or program memory should be verified (see Example 23-6) to the desired value to be written.

### EXAMPLE 23-6: WRITE VERIFY

```
BANKSEL EEDATL ;
MOVWF EEDATL, W ;EEDATL not changed
                ;from previous write
BSF EECON1, RD ;YES, Read the
                ;value written
XORWF EEDATL, W ;
BTFS STATUS, Z ;Is data the same
GOTO WRITE_ERR ;No, handle error
:              ;Yes, continue
```

### 23.5.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM (specification D124) without exceeding the total number of write cycles to a single byte (specifications D120 and D120A). If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

## 23.6 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

## 23.7 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the  $\overline{CPD}$  bit in the Configuration Word Register 1 (Register 10-1) to '0'.

When the data memory is code-protected, only the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from replacing your program with a program that will access the contents of the data EEPROM.

**TABLE 23-2: SUMMARY OF REGISTERS ASSOCIATED WITH DATA EEPROM**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
EECON1	EEPGD	CFGFS	LWLO	FREE	WRERR	WREN	WR	RD	323
EECON2	EEPROM Control Register 2 (not a physical register)								324*
EEADRL	EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0	322
EEADRH	—	EEADRH6	EEADRH5	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0	322
EEDATL	EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0	322
EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	322
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	73
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	78

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', α = value depends upon condition. Shaded cells are not used by data EEPROM module.

\* Page provides register information.

## 24.0 POWER-DOWN MODE (SLEEP)

The Power-down mode is entered by executing a `SLEEP` instruction.

If the Watchdog Timer is enabled:

- $\overline{\text{WDT}}$  will be cleared but keeps running.
- $\overline{\text{PD}}$  bit of the STATUS register is cleared.
- $\overline{\text{TO}}$  bit of the STATUS register is set.
- Oscillator driver is turned off.
- Timer1 oscillator is unaffected
- 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 VDD or VSS, with no external circuitry drawing current from the I/O pin. 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 VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

Enabled Resets remain functional during Sleep.

### 24.1 Wake-up from Sleep

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

1. External Reset input on  $\overline{\text{MCLR}}$  pin, if enabled.
2. BOR Reset, if enabled.
3. Watchdog Timer wake-up (if WDT was enabled).
4. Any external interrupt.
5. Certain peripheral interrupts (see individual peripheral for more information).

The first two events will cause a device Reset. The last three events are considered a continuation of program execution. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits in the STATUS register can be used to determine the cause of device Reset. The  $\overline{\text{PD}}$  bit, which is set on power-up, is cleared when Sleep is invoked.  $\overline{\text{TO}}$  bit is cleared if WDT wake-up occurred.

Certain 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 prefetched. 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.

# PIC16F193X/LF193X

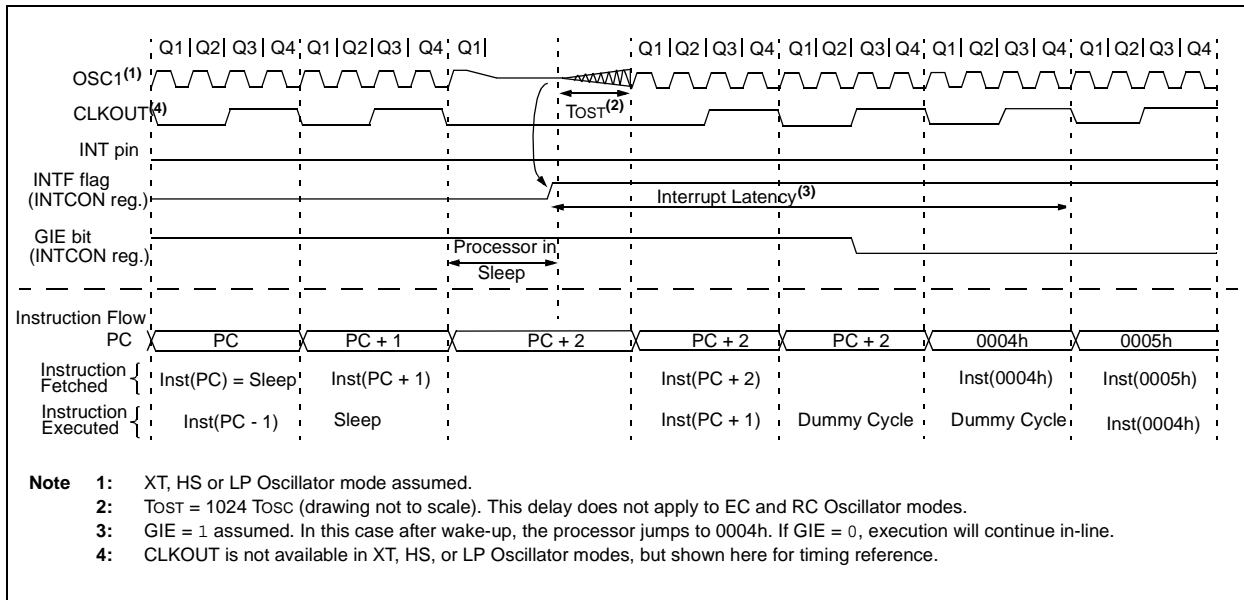
## 24.2 Wake-up Using Interrupts

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`.

**FIGURE 24-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



**TABLE 24-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	104
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	104
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	104
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	73
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	74
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	75
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	77
PIR2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	78

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used in Power-down mode.



## 25.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

In Program/Verify mode the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the “PIC16193X/PIC16LF193X Memory Programming Specification” (DS41360A)

### 25.1 High-voltage Programming Mode

The device is placed into high-voltage Program/Verify mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on MCLR/VPP to VIH.

**Note:** The ICD 2 produces a VPP voltage greater than the maximum VPP specification of the PIC16F193X/LF193X. When using this programmer, an external circuit is required to keep the VPP voltage within the device specifications.

### 25.2 Low-Voltage Programming Mode

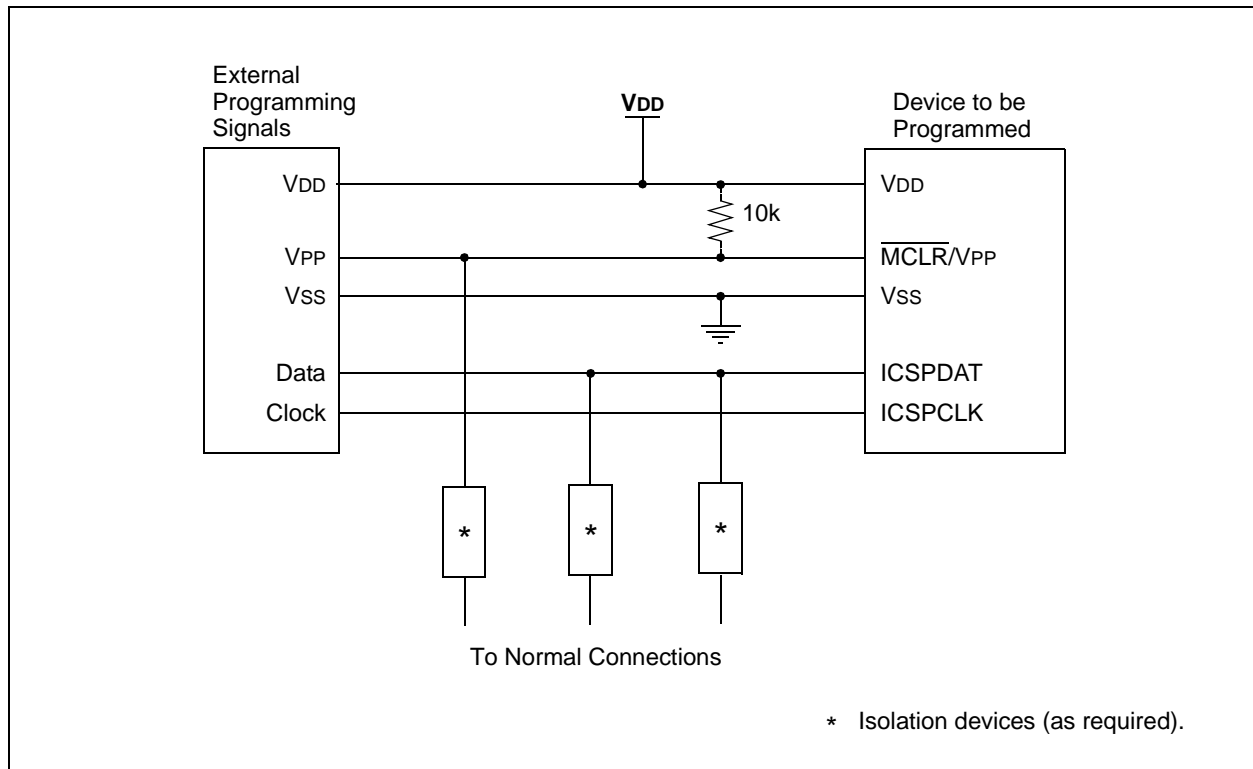
The Low-Voltage Programming mode allows the PIC16F193X/LF193X devices to be programmed using VDD only, without high voltage. When the LVP bit of the Configuration Word 2 register is set to ‘1’, the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to ‘0’.

Entry into the Low-Voltage ICSP Program/Verify modes requires the following steps:

1.  $\overline{\text{MCLR}}$  is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete,  $\overline{\text{MCLR}}$  must be held at VIL for as long as Program/Verify mode is to be maintained.

**FIGURE 25-1: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING**



# PIC16F193X/LF193X

---

NOTES:

## 26.0 INSTRUCTION SET SUMMARY

Each PIC16 instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 26-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the MSb of the appropriate file select register is set.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

## 26.1 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.

**TABLE 26-1: OPCODE FIELD DESCRIPTIONS**

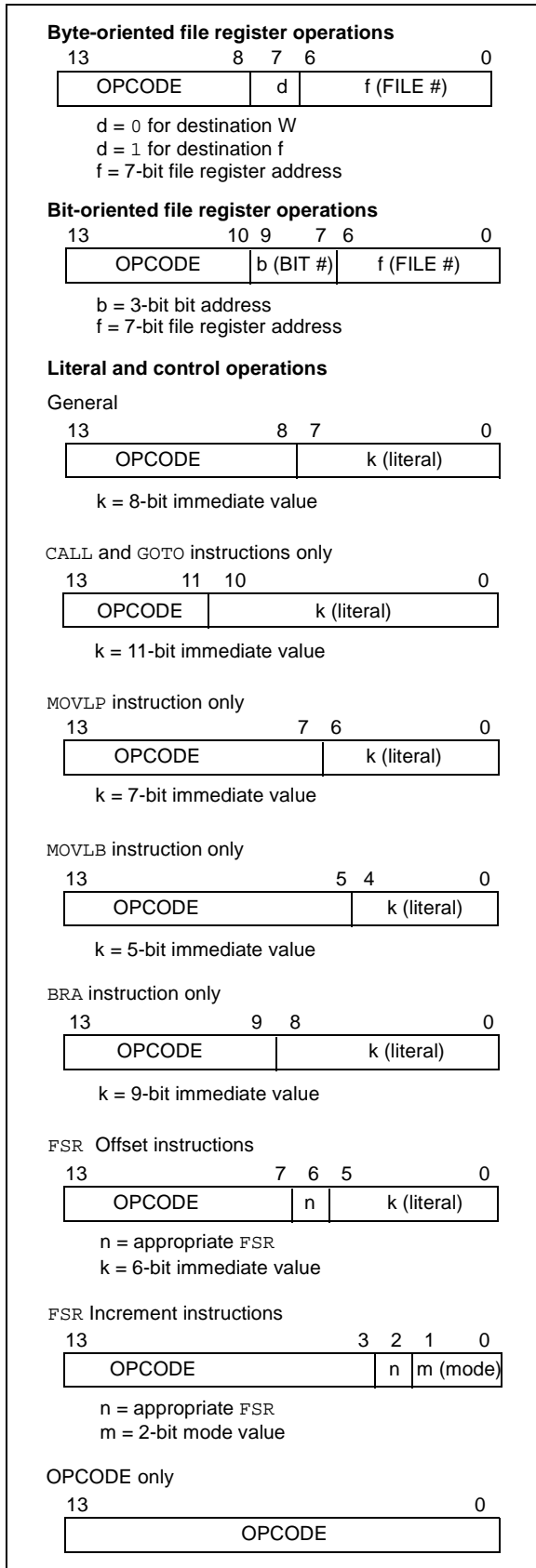
Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility 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.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

**TABLE 26-2: ABBREVIATION DESCRIPTIONS**

Field	Description
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

# PIC16F193X/LF193X

**FIGURE 26-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16F193X/LF193X

**TABLE 26-3: PIC16F193X/LF193X ENHANCED INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb	LSb					
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	–	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	Z	2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	Z	2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
<b>BYTE ORIENTED SKIP OPERATIONS</b>									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
<b>BIT-ORIENTED SKIP OPERATIONS</b>									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
<b>LITERAL OPERATIONS</b>									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

**Note 1:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

# PIC16F193X/LF193X

**TABLE 26-3: PIC16F193X/LF193X ENHANCED INSTRUCTION SET (CONTINUED)**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes
			MSb	LSb				
<b>CONTROL OPERATIONS</b>								
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk	
BRW	–	Relative Branch with W	2	00	0000	0000	1011	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk	
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	
RETFIE	k	Return from interrupt	2	00	0000	0000	1001	
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk	
RETURN	–	Return from Subroutine	2	00	0000	0000	1000	
<b>INHERENT OPERATIONS</b>								
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$ , $\overline{PD}$
NOP	–	No Operation	1	00	0000	0000	0000	
OPTION	–	Load OPTION_REG register with W	1	00	0000	0110	0010	
RESET	–	Software device Reset	1	00	0000	0000	0001	
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$ , $\overline{PD}$
TRIS	f	Load TRIS register with W	1	00	0000	0110	01kk	
<b>C-COMPILER OPTIMIZED</b>								
ADDFSR	n, k	Add Literal to FSRn	1	11	0001	0nkk	kkkk	
MOVIW	mm n	Move INDFn to W, with pre/post inc/dec	1	00	0000	0001	0mmn	Z
	n mm	Move INDFn to W, with pre/post inc/dec	1	00	0000	0001	0nmm	Z
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	kkkk	Z
MOVWI	mm n	Move W to INDFn, with pre/post inc/dec	1	00	0000	0001	1mmn	Z
	n mm	Move W to INDFn, with pre/post inc/dec	1	00	0000	0001	1nmm	Z
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk	kkkk	Z

**Note 1:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

## 26.2 Instruction Descriptions

<b>ADDFSR</b>	<b>Add Literal to FSRn</b>
Syntax:	[ <i>label</i> ] ADDFSR n, k
Operands:	-32 ≤ k ≤ 31 n ∈ [ 0, 1 ]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair.  FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap around.

<b>ANDLW</b>	<b>AND literal with W</b>
Syntax:	[ <i>label</i> ] ANDLW k
Operands:	0 ≤ k ≤ 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.

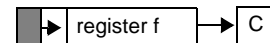
<b>ADDLW</b>	<b>Add literal and W</b>
Syntax:	[ <i>label</i> ] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

<b>ANDWF</b>	<b>AND W with f</b>
Syntax:	[ <i>label</i> ] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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'.

<b>ADDWF</b>	<b>Add W and f</b>
Syntax:	[ <i>label</i> ] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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'.

<b>ASRF</b>	<b>Arithmetic Right Shift</b>
Syntax:	[ <i>label</i> ] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>) → dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

<b>ADDWFC</b>	<b>ADD W and CARRY bit to f</b>
Syntax:	[ <i>label</i> ] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.



# PIC16F193X/LF193X

---

<b>BCF</b>	<b>Bit Clear f</b>
Syntax:	[ <i>label</i> ] BCF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$0 \rightarrow (f<b>)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

<b>BTFSC</b>	<b>Bit Test f, Skip if Clear</b>
Syntax:	[ <i>label</i> ] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
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 2-cycle instruction.

<b>BRA</b>	<b>Relative Branch</b>
Syntax:	[ <i>label</i> ] BRA k
Operands:	$-256 \leq k \leq 255$
Operation:	$(PC) + k \rightarrow PC$
Status Affected:	None
Description:	Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 1 + k$ . This instruction is a two-cycle instruction.

<b>BTFSS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	[ <i>label</i> ] BTFSS f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
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 2-cycle instruction.

<b>BRW</b>	<b>Relative Branch with W</b>
Syntax:	[ <i>label</i> ] BRW
Operands:	None
Operation:	$(PC) + (W) \rightarrow PC$
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 1 + (W)$ . This instruction is a two-cycle instruction.

<b>BSF</b>	<b>Bit Set f</b>
Syntax:	[ <i>label</i> ] BSF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$1 \rightarrow (f<b>)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.



# PIC16F193X/LF193X

## CALL Call Subroutine

**Syntax:** [ *label* ] CALL *k*

**Operands:**  $0 \leq k \leq 2047$

**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.

## CALLW Subroutine Call With W

**Syntax:** [ *label* ] CALLW

**Operands:** None

**Operation:** (PC) + 1 → TOS,  
(W) → PC<7:0>,  
(PCLATH<6:0>) → PC<14:8>

**Status Affected:** None

**Description:** Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a two-cycle instruction.

## CLRF Clear f

**Syntax:** [ *label* ] CLRF *f*

**Operands:**  $0 \leq f \leq 127$

**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 →  $\overline{TO}$   
1 →  $\overline{PD}$

**Status Affected:**  $\overline{TO}$ ,  $\overline{PD}$

**Description:** CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits  $\overline{TO}$  and  $\overline{PD}$  are set.

## COMF Complement f

**Syntax:** [ *label* ] COMF *f*,*d*

**Operands:**  $0 \leq f \leq 127$   
*d* ∈ [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:**  $0 \leq f \leq 127$   
*d* ∈ [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'.

# PIC16F193X/LF193X

---

## **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 2-cycle instruction.

## **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 2-cycle instruction.

## **GOTO**      **Unconditional Branch**

---

Syntax:      [ *label* ] GOTO k

Operands:     $0 \leq k \leq 2047$

Operation:     $k \rightarrow PC<10:0>$   
               $PCLATH<4:3> \rightarrow PC<12:11>$

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.

## **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.

## **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'.

## **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'.

**LSLF**                      **Logical Left Shift**

---

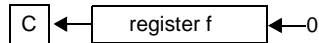
Syntax:                    `[ label ] LSLF f {,d}`

Operands:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation:                 $(f<7>) \rightarrow C$   
 $(f<6:0>) \rightarrow \text{dest}<7:1>$   
 $0 \rightarrow \text{dest}<0>$

Status Affected:        C, Z

Description:             The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



**LSRF**                      **Logical Right Shift**

---

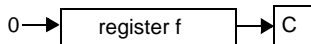
Syntax:                    `[ label ] LSRF f {,d}`

Operands:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation:                 $0 \rightarrow \text{dest}<7>$   
 $(f<7:1>) \rightarrow \text{dest}<6:0>$ ,  
 $(f<0>) \rightarrow C$ ,

Status Affected:        C, Z

Description:             The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



**MOVF**                      **Move f**

---

Syntax:                    `[ label ] MOVF f,d`

Operands:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation:                 $(f) \rightarrow (\text{dest})$

Status Affected:        Z

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  
 $W = \text{value in FSR register}$   
 $Z = 1$

# PIC16F193X/LF193X

## MOVIW Move INDFn to W

Syntax: [ *label* ] MOVIW ++INDFn  
 [ *label* ] MOVIW --INDFn  
 [ *label* ] MOVIW INDFn++  
 [ *label* ] MOVIW INDFn--  
 [ *label* ] MOVIW [k]INDFn  
 [ *label* ] MOVIW INDFn

Operands:  $n \in [0, 1]$   
 $mm \in [00, 01, 10, 11]$ .  
 $-32 \leq k \leq 31$   
 If not present,  $k = 0$ .

Operation:  $INDFn \rightarrow W$   
 Effective address is determined by

- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:

- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

mm	Mode	Syntax
00	Preincrement	++INDFn
01	Predecrement	--INDFn
10	Postincrement	INDFn++
11	Postdecrement	INDFn--

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

## MOVLB Move literal to BSR

Syntax: [ *label* ] MOVLB k

Operands:  $0 \leq k \leq 15$

Operation:  $k \rightarrow BSR$

Status Affected: None

Description: The five-bit literal 'k' is loaded into the Bank Select Register (BSR).

## MOVLP Move literal to PCLATH

Syntax: [ *label* ] MOVLP k

Operands:  $0 \leq k \leq 127$

Operation:  $k \rightarrow PCLATH$

Status Affected: None

Description: The seven-bit literal 'k' is loaded into the PCLATH register.

## MOVLW Move literal to W

Syntax: [ *label* ] MOVLW k

Operands:  $0 \leq k \leq 255$

Operation:  $k \rightarrow (W)$

Status Affected: None

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:  $0 \leq f \leq 127$

Operation:  $(W) \rightarrow (f)$

Status Affected: None

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

**MOVWI**                      **Move W to INDFn**

---

Syntax:                      [ *label* ] MOVWI ++INDFn  
                                  [ *label* ] MOVWI --INDFn  
                                  [ *label* ] MOVWI INDFn++  
                                  [ *label* ] MOVWI INDFn--  
                                  [ *label* ] MOVWI [k]INDFn  
                                  [ *label* ] MOVWI INDFn

Operands:                      n ∈ [0,1]  
                                  mm ∈ [00, 01, 10, 11].  
                                  -32 ≤ k ≤ 31  
                                  If not present, k = 0.

Operation:                      W → INDFn  
                                  Effective address is determined by

- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:

- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected:              None

mm	Mode	Syntax
00	Preincrement	++INDFn
01	Predecrement	--INDFn
10	Postincrement	INDFn++
11	Postdecrement	INDFn--

Description:                      This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

**NOP**                              **No Operation**

---

Syntax:                              [ *label* ] NOP

Operands:                              None

Operation:                              No operation

Status Affected:                      None

Description:                              No operation.

Words:                                      1

Cycles:                                      1

Example:                                  NOP

**OPTION**                              **Load OPTION\_REG Register with W**

---

Syntax:                              [ *label* ] OPTION

Operands:                              None

Operation:                              (W) → OPTION\_REG

Status Affected:                      None

Description:                              Move data from W register to OPTION\_REG register.

**RESET**                              **Software Reset**

---

Syntax:                              [ *label* ] RESET

Operands:                              None

Operation:                              Execute a device Reset. Resets the nRI flag of the PCON register.

Status Affected:                      None

Description:                              This instruction provides a way to execute a hardware Reset by software.

# PIC16F193X/LF193X

## RETFIE      Return from Interrupt

**Syntax:**            [ *label* ] RETFIE

**Operands:**        None

**Operation:**        TOS → PC,  
                          1 → GIE

**Status Affected:**    None

**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

```

## 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.

## RETLW      Return with literal in W

**Syntax:**            [ *label* ] RETLW *k*

**Operands:**         $0 \leq k \leq 255$

**Operation:**        *k* → (W);  
                          TOS → PC

**Status Affected:**    None

**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 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

## RLF      Rotate Left f through Carry

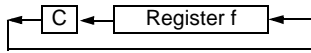
**Syntax:**            [ *label* ] RLF *f*,*d*

**Operands:**         $0 \leq f \leq 127$   
                          *d* ∈ [0,1]

**Operation:**        See description below

**Status Affected:**    C

**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

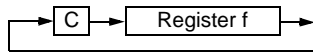
**Syntax:** [ *label* ] RRF f,d

**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 Enter Sleep mode

**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:**  $k - (W) \rightarrow (W)$

**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.

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

## SUBWF Subtract W from f

**Syntax:** [ *label* ] SUBWF f,d

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**  $(f) - (W) \rightarrow (\text{destination})$

**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'.

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

## SUBWFB Subtract W from f with Borrow

**Syntax:** SUBWFB f {,d}

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**  $(f) - (W) - (\overline{B}) \rightarrow \text{dest}$

**Status Affected:** C, DC, Z

**Description:** Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

# PIC16F193X/LF193X

---

## SWAPF Swap Nibbles in f

---

Syntax: [ *label* ] SWAPF f,d

Operands:  $0 \leq f \leq 127$   
 $d \in [0,1]$

Operation: (f<3:0>) → (destination<7:4>),  
(f<7:4>) → (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'.

## TRIS Load TRIS Register with W

---

Syntax: [ *label* ] TRIS f

Operands:  $5 \leq f \leq 7$

Operation: (W) → TRIS register 'f'

Status Affected: None

Description: Move data from W register to TRIS register.  
When 'f' = 5, TRISA is loaded.  
When 'f' = 6, TRISB is loaded.  
When 'f' = 7, TRISC is loaded.

## 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'.



## 27.0 DEVELOPMENT SUPPORT

The PIC<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM<sup>™</sup> Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK<sup>™</sup> Object Linker/  
MPLIB<sup>™</sup> Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE<sup>™</sup> In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART<sup>®</sup> Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICKit<sup>™</sup> 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

## 27.1 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<sup>®</sup> operating system-based application that contains:

- A single graphical interface to all 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
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU 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 increased flexibility and power.

# PIC16F193X/LF193X

---

## 27.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC 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

## 27.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. 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.

## 27.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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

## 27.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its 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

## 27.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 27.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC 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 architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC 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.

## 27.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 27.9 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 PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. 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™ (ICSP™) 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, and 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 PIC devices.

## 27.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer 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 PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The 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.

# PIC16F193X/LF193X

---

## 27.11 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 PIC devices in DIP packages 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.

## 27.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 27.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) for the complete list of demonstration, development and evaluation kits.

## 28.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings<sup>(†)</sup>

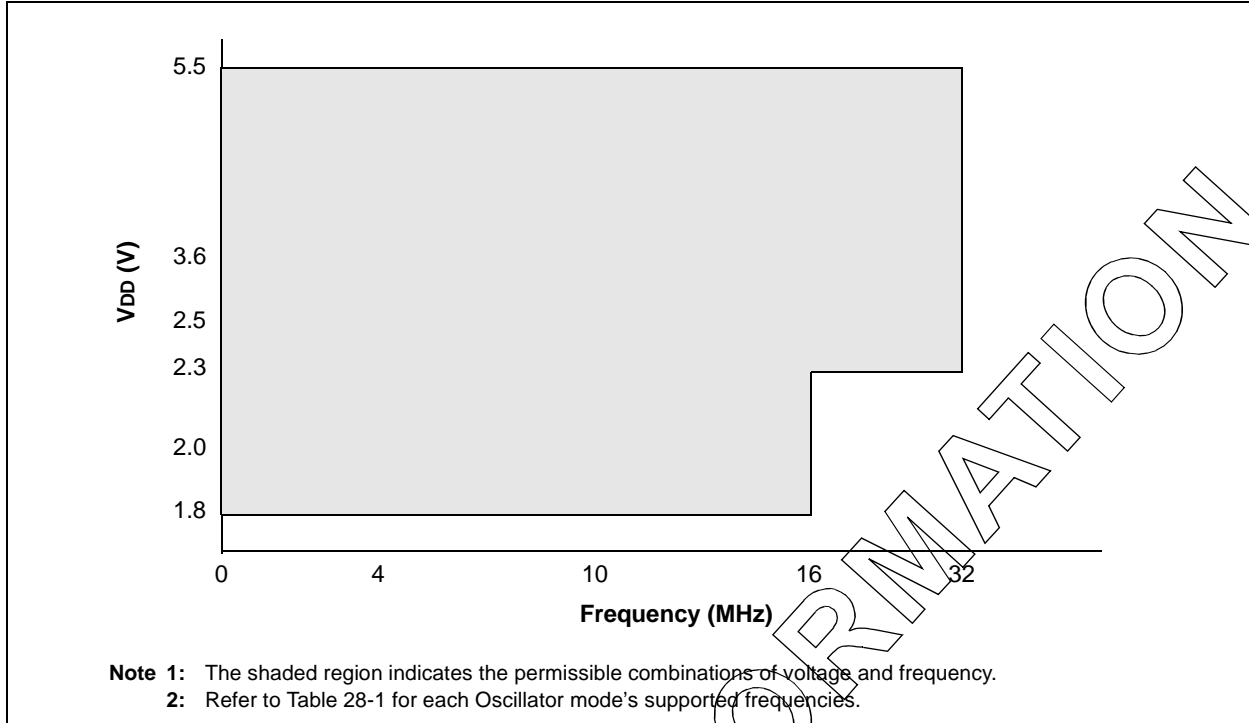
Ambient temperature under bias .....	-40°C to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to Vss, PIC16F193X .....	-0.3V to +6.5V
Voltage on VDD with respect to Vss, PIC16LF193X .....	-0.3V to +4.0V
Voltage on MCLR with respect to Vss .....	-0.3V to +9.0V
Voltage on all other pins with respect to Vss .....	-0.3V to (VDD + 0.3V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of Vss pin .....	95 mA
Maximum current into VDD pin .....	70 mA
Clamp current, I <sub>K</sub> (V <sub>PIN</sub> < 0 or V <sub>PIN</sub> > VDD) .....	± 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 all ports <sup>(2)</sup> , -40°C ≤ TA ≤ +85°C for industrial .....	200 mA
Maximum current sunk by all ports <sup>(2)</sup> , -40°C ≤ TA ≤ +125°C for extended .....	90 mA
Maximum current sourced by all ports <sup>(2)</sup> , 40°C ≤ TA ≤ +85°C for industrial .....	140 mA
Maximum current sourced by all ports <sup>(2)</sup> , -40°C ≤ TA ≤ +125°C for extended .....	65 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times (I_{DD} - \sum I_{OH}) + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ .

† 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 above maximum rating conditions for extended periods may affect device reliability.

# PIC16F193X/LF193X

**FIGURE 28-1: PIC16F193X VOLTAGE FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**



**FIGURE 28-2: PIC16LF193X VOLTAGE FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**

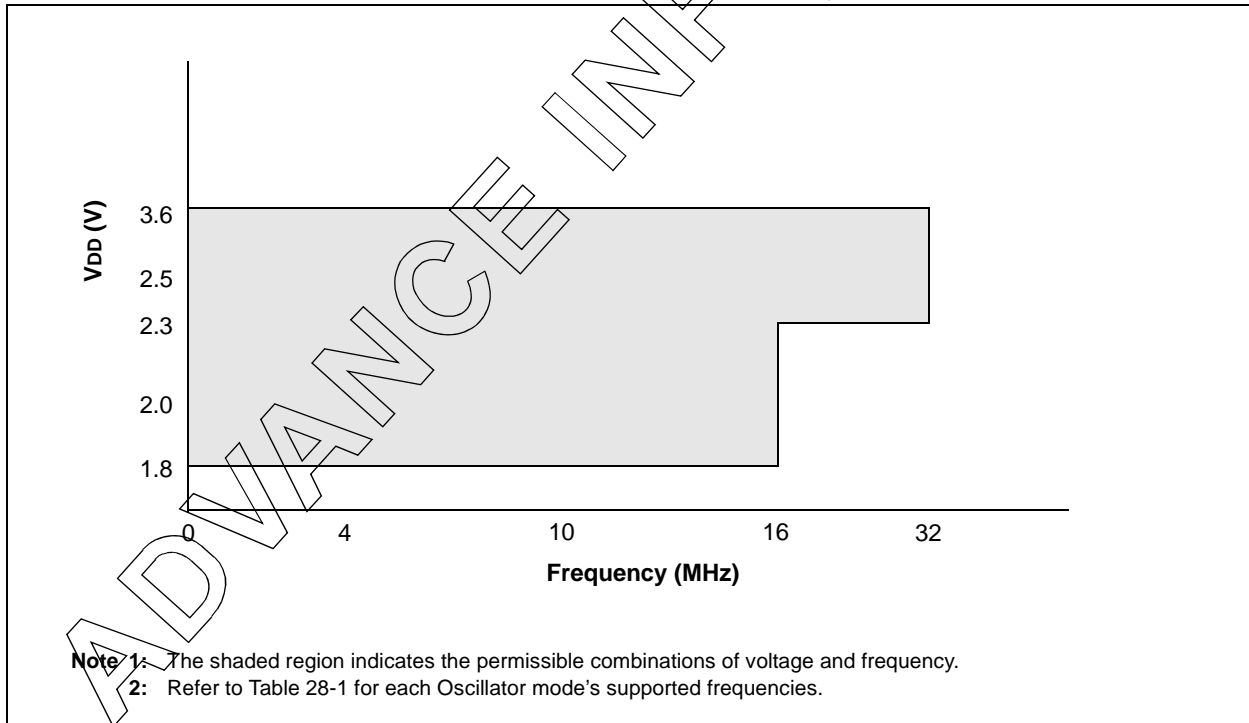
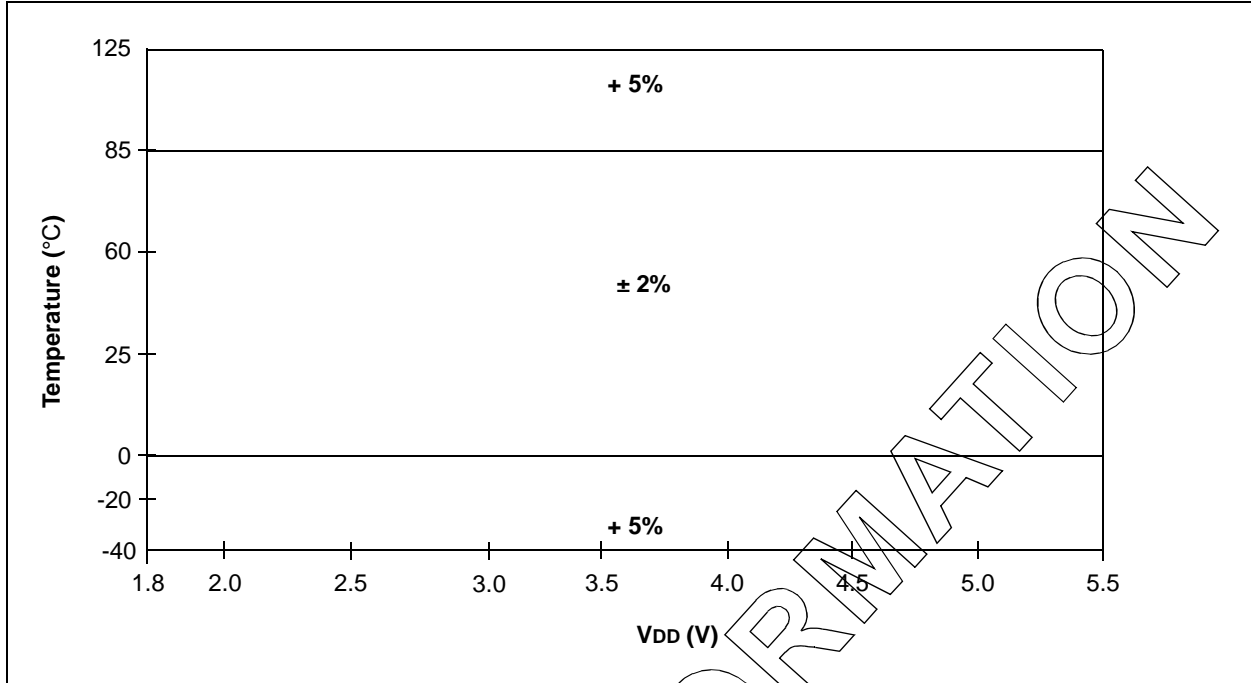


FIGURE 28-3: HFINTOSC FREQUENCY ACCURACY OVER DEVICE V<sub>DD</sub> AND TEMPERATURE



ADVANCE INFORMATION

# PIC16F193X/LF193X

## 28.1 DC Characteristics: PIC16F193X/LF193X-I/E (Industrial, Extended)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F193X		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
D001	VDD	<b>Supply Voltage</b>					
		PIC16LF193X	1.8 2.3	— —	3.6 3.6	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (NOTE 2)
D001		PIC16F193X	1.8 2.3	— —	5.5 5.5	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (NOTE 2)
D002*	VDR	<b>RAM Data Retention Voltage<sup>(1)</sup></b>					
		PIC16LF193X	1.5	—	—	V	Device in Sleep mode
D002*		PIC16F193X	1.7	—	—	V	Device in Sleep mode
	VPOR*	<b>Power-on Reset Release Voltage</b>					
	VPORR*	<b>Power-on Reset Rearm Voltage</b>					
		PIC16LF193X	—	0.8	—	V	Device in Sleep mode
		PIC16F193X	—	1.7	—	V	Device in Sleep mode
	VADFVR	<b>Fixed Voltage Reference Voltage for ADC (calibrated)</b>	0.984	1.024	1.064	V	FVRV = 00 (1x), VDD ≥ 2.5V 125°C FVRV = 01 (2x), VDD ≥ 2.5V 125°C FVRV = 10 (4x), VDD ≥ 4.75V 125°C
			0.974	2.048	1.064		
			1.968	2.048	2.158		
			1.938	4.096	2.148		
			3.966	4.096	4.226		
			3.936	4.226	4.226		
	VCDAFVR	<b>Fixed Voltage Reference Voltage for Comparator and DAC</b>	0.984	1.024	1.064	V	FVRV = 00 (1x), VDD ≥ 2.5V 125°C FVRV = 01 (2x), VDD ≥ 2.5V 125°C FVRV = 10 (4x), VDD ≥ 4.75V 125°C
			0.974	2.048	1.064		
			1.968	2.048	2.158		
			1.938	4.096	2.148		
			3.966	4.096	4.226		
			3.936	4.226	4.226		
	VFVR_REF	<b>Fixed Voltage Reference Voltage for LCD Bias</b>	0.984	1.024	1.064	V	FVRV = 00 (1x), VDD ≥ 2.5V 125°C
			0.974	—	1.064		
D004*	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See Section 3.2 "Power-on Reset (POR)" for details.

\* These parameters are characterized but not tested.

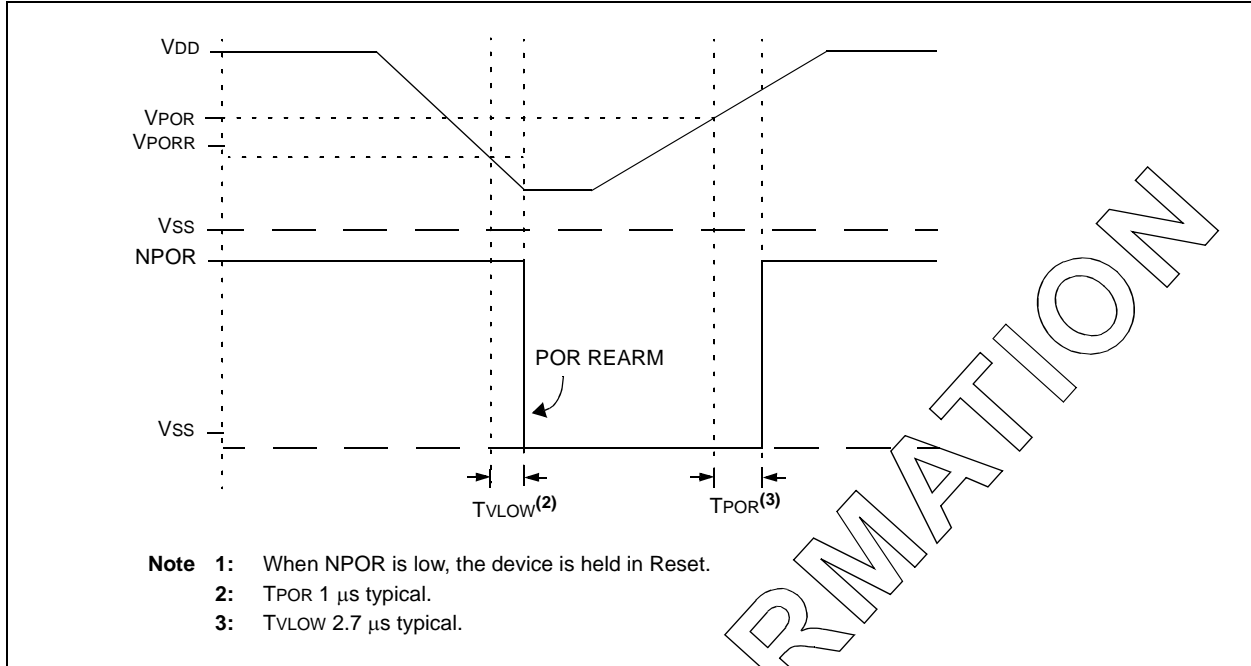
† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**Note 2:** PLL required for 32 MHz operation.



**FIGURE 28-4: POR AND POR REARM WITH SLOW RISING V<sub>DD</sub>**



ADVANCE INFORMATION

# PIC16F193X/LF193X

## 28.2 DC Characteristics: PIC16F193X/LF193X-I/E (Industrial, Extended)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F193X		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
<b>Supply Current (IDD)<sup>(1, 2)</sup></b>							
D009	LDO Regulator	—	350	TBD	μA	—	HS, EC OR INTOSC/INTOSCIO (8-16 MHz) Clock modes with all VCAP pins disabled
		—	50	TBD	μA	—	All VCAP pins disabled
		—	30	TBD	μA	—	VCAP enabled on RA0, RA5 or RA6
		—	5	TBD	μA	—	LP Clock mode and Sleep (requires FVR and BOR to be disabled)
D010		—	7.0	TBD	μA	1.8	FOSC = 32 kHz LP Oscillator mode (Note 4), -40°C ≤ TA ≤ +85°C
		—	9.0	TBD	μA	3.0	
D010		—	9.5	TBD	μA	1.8	FOSC = 32 kHz LP Oscillator mode (Note 4), -40°C ≤ TA ≤ +85°C
		—	12.5	TBD	μA	3.0	
		—	13.5	TBD	μA	5.0	
D011*		—	7.0	TBD	μA	1.8	FOSC = 32 kHz LP Oscillator mode
		—	9.0	TBD	μA	3.0	
D011*		—	9.5	TBD	μA	1.8	FOSC = 32 kHz LP Oscillator mode (Note 4)
		—	12.5	TBD	μA	3.0	
		—	13.5	TBD	μA	5.0	
D011A*		—	150	TBD	μA	1.8	FOSC = 1 MHz XT Oscillator mode
		—	270	TBD	μA	3.0	
D011A*		—	160	TBD	μA	1.8	FOSC = 1 MHz XT Oscillator mode (Note 5)
		—	260	TBD	μA	3.0	
		—	390	TBD	μA	5.0	
D012		—	430	TBD	μA	1.8	FOSC = 4 MHz XT Oscillator mode
		—	750	TBD	μA	3.0	
D012		—	450	TBD	μA	1.8	FOSC = 4 MHz XT Oscillator mode (Note 5)
		—	770	TBD	μA	3.0	
		—	930	TBD	μA	5.0	
D013*		—	180	TBD	μA	1.8	FOSC = 1 MHz EC Oscillator mode
		—	350	TBD	μA	3.0	

\* These parameters are characterized but not tested.

**Legend:** TBD = To Be Determined

- 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; 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:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula  $I_R = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.
- 4:** FVR and BOR are disabled.
- 5:** 0.1 μF capacitor on VCAP (RA0).

# PIC16F193X/LF193X

## 28.2 DC Characteristics: PIC16F193X/LF193X-I/E (Industrial, Extended) (Continued)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
PIC16F193X		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D013*		—	200	TBD	$\mu\text{A}$	1.8	Fosc = 1 MHz EC Oscillator mode (Note 5)
		—	370	TBD	$\mu\text{A}$	3.0	
		—	450	TBD	$\mu\text{A}$	5.0	
D014	Supply Current (I <sub>DD</sub> ) <sup>(1, 2)</sup>	—	450	TBD	$\mu\text{A}$	1.8	Fosc = 4 MHz EC Oscillator mode
		—	830	TBD	$\mu\text{A}$	3.0	
		—	475	TBD	$\mu\text{A}$	1.8	
D014		—	850	TBD	$\mu\text{A}$	3.0	
		—	980	TBD	$\mu\text{A}$	5.0	
		D015		—	130	TBD	$\mu\text{A}$
—	190			TBD	$\mu\text{A}$	3.0	
D015				—	150	TBD	$\mu\text{A}$
		—	210	TBD	$\mu\text{A}$	3.0	
		—	270	TBD	$\mu\text{A}$	5.0	
D016*		—	980	TBD	$\mu\text{A}$	1.8	Fosc = 8 MHz HFINTOSC mode
		—	1780	TBD	$\mu\text{A}$	3.0	
D016*		—	1.0	TBD	mA	1.8	Fosc = 8 MHz HFINTOSC mode (Note 5)
		—	1.8	TBD	mA	3.0	
		—	2.0	TBD	mA	5.0	
D017		—	1.5	TBD	mA	1.8	Fosc = 16 MHz HFINTOSC mode
		—	2.8	TBD	mA	3.0	
D017		—	1.7	TBD	mA	1.8	Fosc = 16 MHz HFINTOSC mode (Note 5)
		—	2.9	TBD	mA	3.0	
		—	3.1	TBD	mA	5.0	
D018		—	410	TBD	$\mu\text{A}$	1.8	Fosc = 4 MHz EXTRC mode (Note 3, Note 5)
		—	710	TBD	$\mu\text{A}$	3.0	
D018		—	430	TBD	$\mu\text{A}$	1.8	Fosc = 4 MHz EXTRC mode (Note 3, Note 5)
		—	730	TBD	$\mu\text{A}$	3.0	
		—	860	TBD	$\mu\text{A}$	5.0	
D019		—	5.3	TBD	mA	3.0	Fosc = 32 MHz HS Oscillator mode
		—	6.0	TBD	mA	3.6	

\* These parameters are characterized but not tested.

**Legend:** TBD = To Be Determined

**Note 1:** The test conditions for all I<sub>DD</sub> measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V<sub>DD</sub>; MCLR = V<sub>DD</sub>; 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:** For RC oscillator configurations, current through R<sub>EXT</sub> is not included. The current through the resistor can be extended by the formula  $I_R = V_{DD}/2R_{EXT}$  (mA) with R<sub>EXT</sub> in k $\Omega$ .

**4:** FVR and BOR are disabled.

**5:** 0.1  $\mu\text{F}$  capacitor on VCAP (RA0).

# PIC16F193X/LF193X

## 28.2 DC Characteristics: PIC16F193X/LF193X-I/E (Industrial, Extended) (Continued)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F193X		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D019		—	5.3	TBD	mA	3.0	FOSC = 32 MHz
		—	6.0	TBD	mA	5.0	HS Oscillator mode (Note 5)

\* These parameters are characterized but not tested.

**Legend:** TBD = To Be Determined

- 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; 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:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula  $I_R = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.
- 4:** FVR and BOR are disabled.
- 5:** 0.1 μF capacitor on VCAP (RA0).

ADVANCE INFORMATION

# PIC16F193X/LF193X

## 28.3 DC Characteristics: PIC16F193X/LF193X-I/E (Power-Down)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated)						
		Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended						
PIC16F193X		Standard Operating Conditions (unless otherwise stated)						
		Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended						
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
<b>Power-down Base Current (IPD)<sup>(2)</sup></b>								
D020		—	0.06	TBD	TBD	μA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	0.08	TBD	TBD	μA	3.0	
D020		—	3.1	TBD	TBD	μA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	3.6	TBD	TBD	μA	3.0	
		—	4.5	TBD	TBD	μA	5.0	
D021		—	0.5	TBD	TBD	μA	1.8	LPWDT Current (Note 1)
		—	0.8	TBD	TBD	μA	3.0	
D021		—	3.8	TBD	TBD	μA	1.8	LPWDT Current (Note 1)
		—	4.3	TBD	TBD	μA	3.0	
		—	5.3	TBD	TBD	μA	5.0	
D021A		—	8.5	TBD	TBD	μA	1.8	FVR current (Note 3)
		—	8.5	TBD	TBD	μA	3.0	
D021A		—	32	TBD	TBD	μA	1.8	FVR current (Note 3, Note 5)
		—	39	TBD	TBD	μA	3.0	
		—	70	TBD	TBD	mA	5.0	
D022		—	—	TBD	TBD	μA	1.8	BOR Current (Note 1, Note 3)
		—	7.5	TBD	TBD	μA	3.0	
D022		—	—	TBD	TBD	μA	1.8	BOR Current (Note 1, Note 3, Note 5)
		—	34	TBD	TBD	μA	3.0	
		—	67	TBD	TBD	μA	5.0	
D026		—	0.6	TBD	TBD	μA	1.8	T1OSC Current (Note 1)
		—	1.8	TBD	TBD	μA	3.0	
D026		—	4.5	TBD	TBD	μA	1.8	T1OSC Current (Note 1)
		—	6	TBD	TBD	μA	3.0	
		—	7	TBD	TBD	μA	5.0	

\* These parameters are characterized but not tested.

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

**Legend:** TBD = To Be Determined

- Note 1:** 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.
- 2:** 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.
- 3:** Fixed Voltage Reference is automatically enabled whenever the BOR is enabled.
- 4:** A/D oscillator source is FRC.
- 5:** 0.1 μF capacitor on VCAP (RA0).

# PIC16F193X/LF193X

## 28.3 DC Characteristics: PIC16F193X/LF193X-I/E (Power-Down) (Continued)

PIC16LF193X		Standard Operating Conditions (unless otherwise stated)						
		Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended						
PIC16F193X		Standard Operating Conditions (unless otherwise stated)						
		Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended						
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
<b>Power-down Base Current (IPD)<sup>(2)</sup></b>								
D027		—	0.1	TBD	TBD	μA	1.8	A/D Current ( <b>Note 1, Note 4</b> ), no conversion in progress
		—	0.1	TBD	TBD	μA	3.0	
D027		—	3.5	TBD	TBD	μA	1.8	A/D Current ( <b>Note 1, Note 4</b> ), no conversion in progress
		—	4	TBD	TBD	μA	3.0	
		—	4.5	TBD	TBD	μA	5.0	
D027A		—	250	TBD	TBD	μA	1.8	A/D Current ( <b>Note 1, Note 4</b> ), conversion in progress
		—	250	TBD	TBD	μA	3.0	
D027A		—	280	TBD	TBD	μA	1.8	A/D Current ( <b>Note 1, Note 4, Note 5</b> ), conversion in progress
		—	280	TBD	TBD	μA	3.0	
		—	280	TBD	TBD	μA	5.0	
D028		—	3.5	TBD	TBD	μA	1.8	Cap Sense
		—	7	TBD	TBD	μA	3.0	
D028		—	3.5	TBD	TBD	μA	1.8	Cap Sense
		—	7	TBD	TBD	μA	3.0	
		—	32	TBD	TBD	μA	5.0	
D029		—	1	TBD	TBD	μA	3.6	LCD Bias Ladder, Low-power
		—	10	TBD	TBD	μA	3.6	LCD Bias Ladder, Medium-power
		—	100	TBD	TBD	μA	3.6	LCD Bias Ladder, High-power
D029		—	1	TBD	TBD	μA	5.0	LCD Bias Ladder, Low-power
		—	10	TBD	TBD	μA	5.0	LCD Bias Ladder, Medium-power
		—	100	TBD	TBD	μA	5.0	LCD Bias Ladder, High-power

\* These parameters are characterized but not tested.

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

**Legend:** TBD = To Be Determined

- Note 1:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral 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.
- Note 2:** 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.
- Note 3:** Fixed Voltage Reference is automatically enabled whenever the BOR is enabled.
- Note 4:** A/D oscillator source is FRC.
- Note 5:** 0.1 μF capacitor on VCAP (RA0).

# PIC16F193X/LF193X

## 28.4 DC Characteristics: PIC16F193X/LF193X-I/E

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D030 D030A D031 D032 D033A	V <sub>IL</sub>	<b>Input Low Voltage</b>					
		I/O PORT:					
		with TTL buffer	—	—	0.8	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
		with Schmitt Trigger buffer	—	—	$0.15 V_{DD}$	V	$1.8\text{V} \leq V_{DD} \leq 4.5\text{V}$
		with I <sup>2</sup> C™ levels	—	—	$0.2 V_{DD}$	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$
		with SMBus™ levels	—	—	$0.3 V_{DD}$	V	
		MCLR, OSC1 (RC mode) <sup>(1)</sup>	—	—	0.8	V	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$
D040 D040A D041 D042 D043A D043B	V <sub>IH</sub>	<b>Input High Voltage</b>					
I/O ports:							
with TTL buffer		2.0	—	—	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$	
with Schmitt Trigger buffer		$0.25 V_{DD} + 0.8$	—	—	V	$1.8\text{V} \leq V_{DD} \leq 4.5\text{V}$	
with I <sup>2</sup> C™ levels		$0.8 V_{DD}$	—	—	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$	
with SMBus™ levels		$0.7 V_{DD}$	—	—	V		
MCLR		2.1	—	—	V	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	
D060 D061 D063	I <sub>IL</sub>	<b>Input Leakage Current<sup>(2)</sup></b>					
I/O ports		—	± 5	± 100	nA	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at high-impedance 125°C	
MCLR <sup>(3)</sup>		—	± 50	± 200	nA	$V_{SS} \leq V_{PIN} \leq V_{DD}$	
D070*	I <sub>PUR</sub>	<b>Weak Pull-up Current</b>					
		25 25	100 140	200 300	μA	$V_{DD} = 3.3\text{V}$ , $V_{PIN} = V_{SS}$ $V_{DD} = 5.0\text{V}$ , $V_{PIN} = V_{SS}$	
D080	V <sub>OL</sub>	<b>Output Low Voltage<sup>(4)</sup></b>					
I/O ports		—	—	0.6	V	$I_{OH} = 8\text{mA}$ , $V_{DD} = 5\text{V}$ $I_{OH} = 6\text{mA}$ , $V_{DD} = 3.3\text{V}$ $I_{OH} = 3\text{mA}$ , $V_{DD} = 1.8\text{V}$	

**Legend:** TBD = To Be Determined

\* These parameters are characterized but not tested.

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

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN 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 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.

**4:** Including OSC2 in CLKOUT mode.

# PIC16F193X/LF193X

## 28.4 DC Characteristics: PIC16F193X/LF193X-I/E (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D090	VOH	<b>Output High Voltage<sup>(4)</sup></b>					
		I/O ports	$V_{DD} - 0.7$	—	—	V	IOH = 3.5mA, VDD = 5V IOH = 3mA, VDD = 3.3V IOH = 2mA, VDD = 1.8V
<b>Capacitive Loading Specs on Output Pins</b>							
D101*	COSC2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101A*	CIO	All I/O pins	—	—	50	pF	
<b>Vcap Capacitor Charging</b>							
D102		Charging current	—	200	—	μA	
D102A		Source/sink capability when charging complete	—	0.0	—	mA	

**Legend:** TBD = To Be Determined

\* These parameters are characterized but not tested.

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

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN 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 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.

**4:** Including OSC2 in CLKOUT mode.



## 28.5 Memory Programming Requirements

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$				
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
<b>Program Memory Programming Specifications</b>							
D110	VIHH	Voltage on $\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$ pin	8.0	—	9.0	V	(Note 3, Note 4)
D111	IDDP	Supply Current during Programming	—	—	10	mA	
D112		VDD for Bulk Erase	2.7	—	VDD max.	V	
D113	VPEW	VDD for Write or Row Erase	VDD min.	—	VDD max.	V	
D114	IPPPGM	Current on $\overline{\text{MCLR}}/\text{VPP}$ during Erase/Write	—	—	1.0	mA	
D115	IDDPGM	Current on VDD during Erase/Write	—	—	5.0	mA	
<b>Data EEPROM Memory</b>							
D116	ED	Byte Endurance	—	100K	—	E/W	-40°C to +85°C
D117	VDRW	VDD for Read/Write	VDD min.	—	VDD max.	V	
D118	TDEW	Erase/Write Cycle Time	—	4.0	5.0	ms	Provided no other specifications are violated
D119	TRETD	Characteristic Retention	40	—	—	Year	
D120	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(2)</sup>	1M	10M	—	E/W	
<b>Program Flash Memory</b>							
D121	EP	Cell Endurance	—	10K	—	E/W	-40°C to +85°C (Note 1)
D122	VPR	VDD for Read	VDD min.	—	VDD max.	V	
D123	TIW	Self-timed Write Cycle Time	—	2	2.5	ms	Provided no other specifications are violated
D124	TRETD	Characteristic Retention	40	—	—	Year	

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

**Note 1:** Self-write and Block Erase.

**Note 2:** Refer to Section 23.5.1 "Using the Data EEPROM" for a more detailed discussion on data EEPROM endurance.

**Note 3:** Required only if single-supply programming is disabled.

**Note 4:** The MPLAB ICD 2 does not support variable VPP output. Circuitry to limit the ICD 2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD 2.

# PIC16F193X/LF193X

## 28.6 Thermal Considerations

Standard Operating Conditions (unless otherwise stated)					
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
Param No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	$\theta_{JA}$	Thermal Resistance Junction to Ambient	60	$^{\circ}\text{C}/\text{W}$	28-pin SPDIP package
			80	$^{\circ}\text{C}/\text{W}$	28-pin SOIC package
			90	$^{\circ}\text{C}/\text{W}$	28-pin SSOP package
			27.5	$^{\circ}\text{C}/\text{W}$	28-pin QFN 6x6mm package
			47.2	$^{\circ}\text{C}/\text{W}$	40-pin PDIP package
			46	$^{\circ}\text{C}/\text{W}$	44-pin TQFP package
			24.4	$^{\circ}\text{C}/\text{W}$	44-pin QFN 8x8mm package
TH02	$\theta_{JC}$	Thermal Resistance Junction to Case	31.4	$^{\circ}\text{C}/\text{W}$	28-pin SPDIP package
			24	$^{\circ}\text{C}/\text{W}$	28-pin SOIC package
			24	$^{\circ}\text{C}/\text{W}$	28-pin SSOP package
			24	$^{\circ}\text{C}/\text{W}$	28-pin QFN 6x6mm package
			24.7	$^{\circ}\text{C}/\text{W}$	40-pin PDIP package
			14.5	$^{\circ}\text{C}/\text{W}$	44-pin TQFP package
			20	$^{\circ}\text{C}/\text{W}$	44-pin QFN 8x8mm package
TH03	$T_{JMAX}$	Maximum Junction Temperature	150	$^{\circ}\text{C}$	
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \sum (I_{OL} \times V_{OL}) + \sum (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	Derated Power	—	W	$P_{DER} = P_{D_{MAX}} (T_J - T_A) / \theta_{JA}^{(2)}$

**Note 1:**  $I_{DD}$  is current to run the chip alone without driving any load on the output pins.

**2:**  $T_A$  = Ambient Temperature

**3:**  $T_J$  = Junction Temperature

ADVANCE INFORMATION

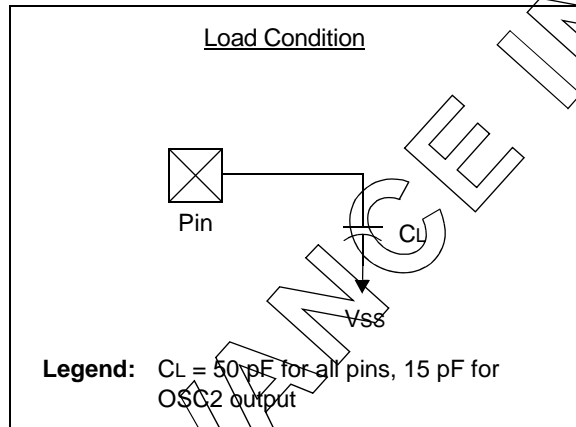
## 28.7 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

<b>T</b>	F      Frequency	T      Time
Lowercase letters (pp) and their meanings:		
<b>pp</b>	cc      CCP1 ck      CLKOUT cs $\overline{CS}$ di      SDI do      SDO dt      Data in io      I/O PORT mc      MCLR	osc      OSC1 rd $\overline{RD}$ rw $\overline{RD}$ or $\overline{WR}$ sc      SCK ss $\overline{SS}$ t0      T0CKI t1      T1CKI wr $\overline{WR}$
Uppercase letters and their meanings:		
<b>S</b>	F      Fall H      High I      Invalid (High-impedance) L      Low	P      Period R      Rise V      Valid Z      High-impedance

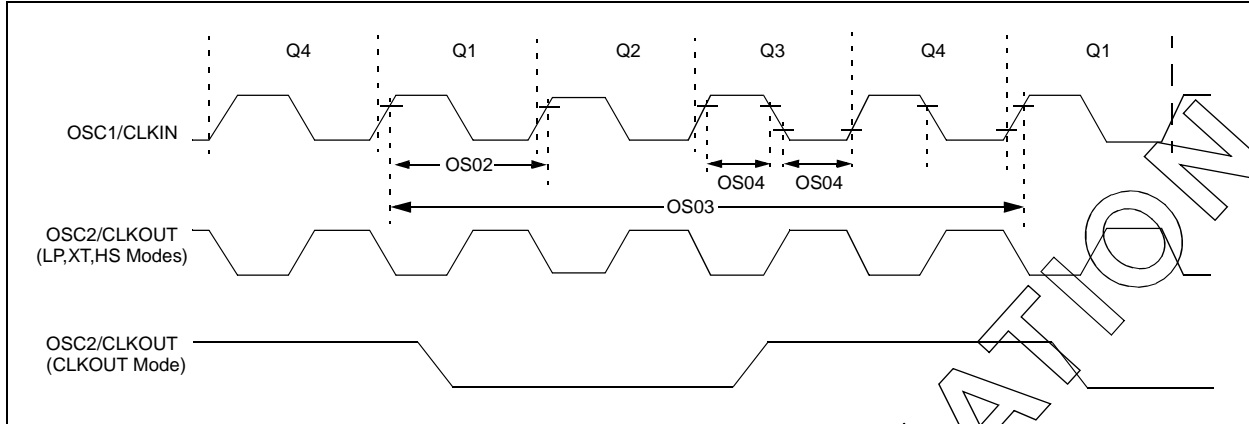
**FIGURE 28-5: LOAD CONDITIONS**



# PIC16F193X/LF193X

## 28.8 AC Characteristics: PIC16F193X/LF193X-I/E

**FIGURE 28-6: CLOCK TIMING**



**TABLE 28-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS01	Fosc	External CLKIN Frequency <sup>(1)</sup>	DC	—	1	MHz	EC Oscillator mode (low)
			DC	—	4	MHz	EC Oscillator mode (medium)
			DC	—	32	MHz	EC Oscillator mode (high)
	Oscillator Frequency <sup>(1)</sup>	—	32,768	—	kHz	LP Oscillator mode	
		0.1	—	4	MHz	XT Oscillator mode	
		1	—	4	MHz	HS Oscillator mode, $V_{DD} \leq 2.3\text{V}$	
1		—	20	MHz	HS Oscillator mode, $V_{DD} > 2.3\text{V}$		
OS02	Tosc	External CLKIN Period <sup>(1)</sup>	27	—	$\infty$	$\mu\text{s}$	LP Oscillator mode
			250	—	$\infty$	ns	XT Oscillator mode
			50	—	$\infty$	ns	HS Oscillator mode
			31.25	—	$\infty$	ns	EC Oscillator mode
	Oscillator Period <sup>(1)</sup>	—	30.5	—	$\mu\text{s}$	LP Oscillator mode	
250		—	10,000	ns	XT Oscillator mode		
50		—	1,000	ns	HS Oscillator mode		
250		—	—	ns	RC Oscillator mode		
OS03	Tcy	Instruction Cycle Time <sup>(1)</sup>	200	Tcy	DC	ns	$T_{CY} = 4/F_{OSC}$
OS04*	TosH, TosL	External CLKIN High, External CLKIN Low	2	—	—	$\mu\text{s}$	LP oscillator
			100	—	—	ns	XT oscillator
			20	—	—	ns	HS oscillator
OS05*	TosR, TosF	External CLKIN Rise, External CLKIN Fall	0	—	$\infty$	ns	LP oscillator
			0	—	$\infty$	ns	XT oscillator
			0	—	$\infty$	ns	HS oscillator

\* These parameters are characterized but not tested.

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

**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 28-2: OSCILLATOR PARAMETERS**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFosc	Internal Calibrated HFINTOSC Frequency <sup>(2)</sup>	$\pm 2\%$	—	16.0	—	MHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
			$\pm 5\%$	—	16.0	—	MHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS08A	MFosc	Internal Calibrated MFINTOSC Frequency <sup>(2)</sup>	$\pm 2\%$	—	500	—	kHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
			$\pm 5\%$	—	500	—	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS10*	Tiosc ST	HFINTOSC and MFINTOSC Wake-up from Sleep Start-up Time	—	—	5	7	$\mu\text{s}$	$V_{DD} = 2.0\text{V}, -40^{\circ}\text{C} \text{ to } +85^{\circ}\text{C}$
			—	—	5	7	$\mu\text{s}$	$V_{DD} = 3.0\text{V}, -40^{\circ}\text{C} \text{ to } +85^{\circ}\text{C}$
			—	—	5	7	$\mu\text{s}$	$V_{DD} = 5.0\text{V}, -40^{\circ}\text{C} \text{ to } +85^{\circ}\text{C}$

\* These parameters are characterized but not tested.

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

**Note 1:** Instruction cycle period (T<sub>cy</sub>) 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 the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**2:** To ensure these oscillator frequency tolerances, V<sub>DD</sub> and V<sub>SS</sub> must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.

**3:** By design.

**TABLE 28-3: PLL CLOCK TIMING SPECIFICATIONS (V<sub>DD</sub> = 2.7V TO 5.5V)**

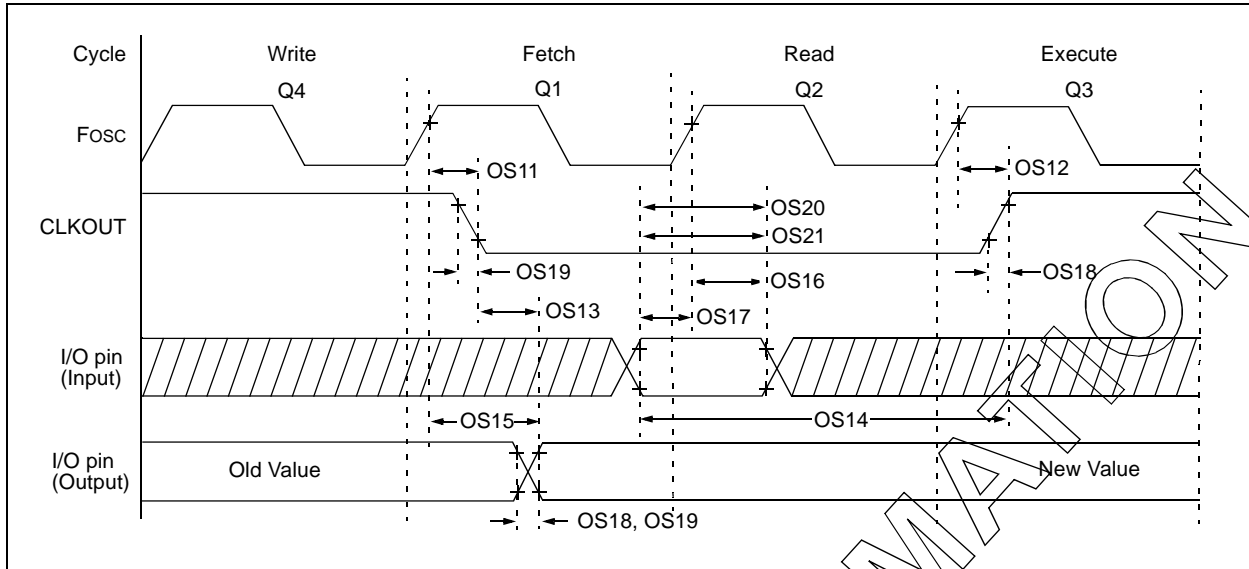
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	$\Delta\text{CLK}$	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

\* 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.

# PIC16F193X/LF193X

FIGURE 28-7: CLKOUT AND I/O TIMING



**TABLE 28-4: CLKOUT AND I/O TIMING PARAMETERS**

Standard Operating Conditions (unless otherwise stated) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ <sup>(1)</sup>	—	—	70	ns	VDD = 3.3-5.0V
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ <sup>(1)</sup>	—	—	72	ns	VDD = 3.3-5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid <sup>(1)</sup>	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑ <sup>(1)</sup>	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	VDD = 3.3-5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	VDD = 3.3-5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18	TioR	Port output rise time <sup>(2)</sup>	—	40 15	72 82	ns	VDD = 1.8V VDD = 3.3-5.0V
OS19	TioF	Port output fall time <sup>(2)</sup>	—	28 15	55 30	ns	VDD = 1.8V VDD = 3.3-5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

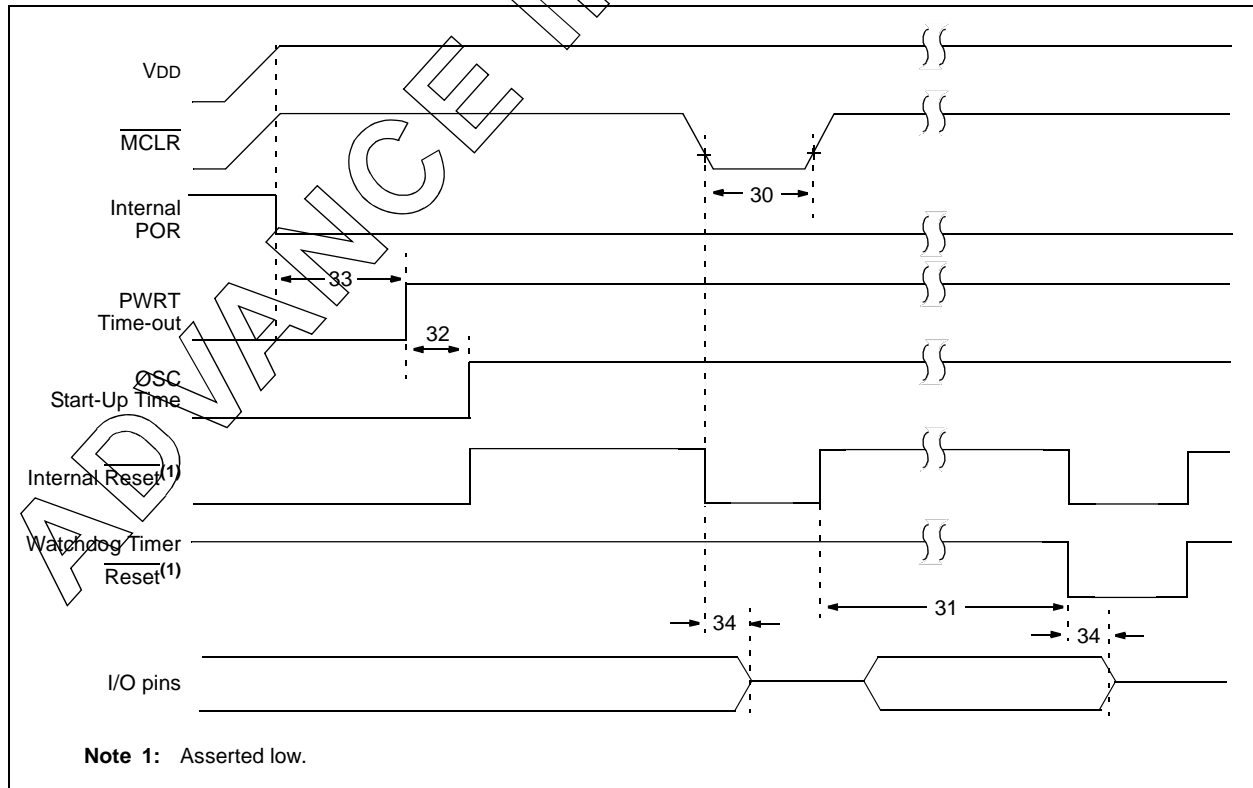
\* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x Tosc.

**2:** Includes OSC2 in CLKOUT mode.

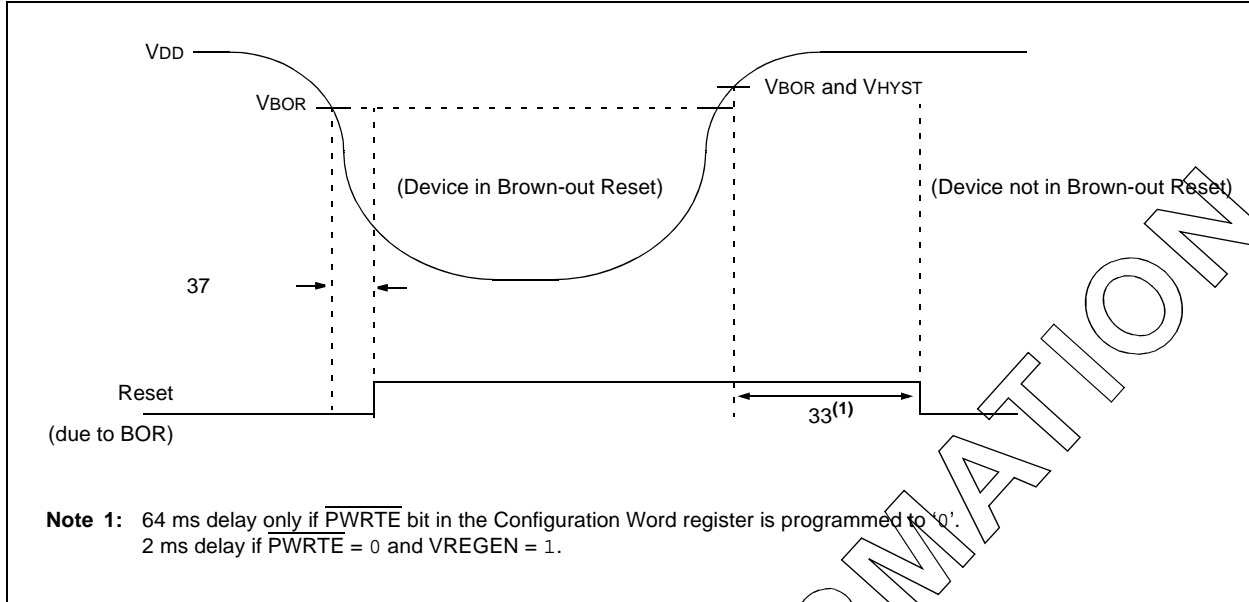
**FIGURE 28-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**Note 1:** Asserted low.

# PIC16F193X/LF193X

FIGURE 28-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



ADVANCE INFORMATION



**TABLE 28-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2	—	—	$\mu\text{s}$	$V_{DD} = 3.3\text{-}5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			5	—	—	$\mu\text{s}$	$V_{DD} = 3.3\text{-}5\text{V}$
31	TWDTP	Low-Power Watchdog Timer Time-out Period (No Prescaler)	10	18	27	ms	$V_{DD} = 3.3\text{V-}5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			10	18	33	ms	$V_{DD} = 3.3\text{V-}5\text{V}$
32	TOST	Oscillator Start-up Timer Period <sup>(1), (2)</sup>	—	1024	—	Tosc	<b>(Note 3)</b>
33*	TPWRT	Power-up Timer Period, $PWRTE = 0$	40	65	140	ms	
34*	TIOZ	I/O high-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0	$\mu\text{s}$	
35	VBOR	Brown-out Reset Voltage	2.40	2.5	2.60	V	BORV=2.5V
			1.80	1.9	2.00	V	BORV=1.9V
36*	VHYST	Brown-out Reset Hysteresis	25	50	75 100	mV	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $-40^{\circ}\text{C}$ to $125^{\circ}\text{C}$
37*	TBORDC	Brown-out Reset DC Response Time	1	3	5 10	$\mu\text{s}$	$V_{DD} \leq V_{BOR}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $V_{DD} \leq V_{BOR}$

\* These parameters are characterized but not tested.

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

**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 the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

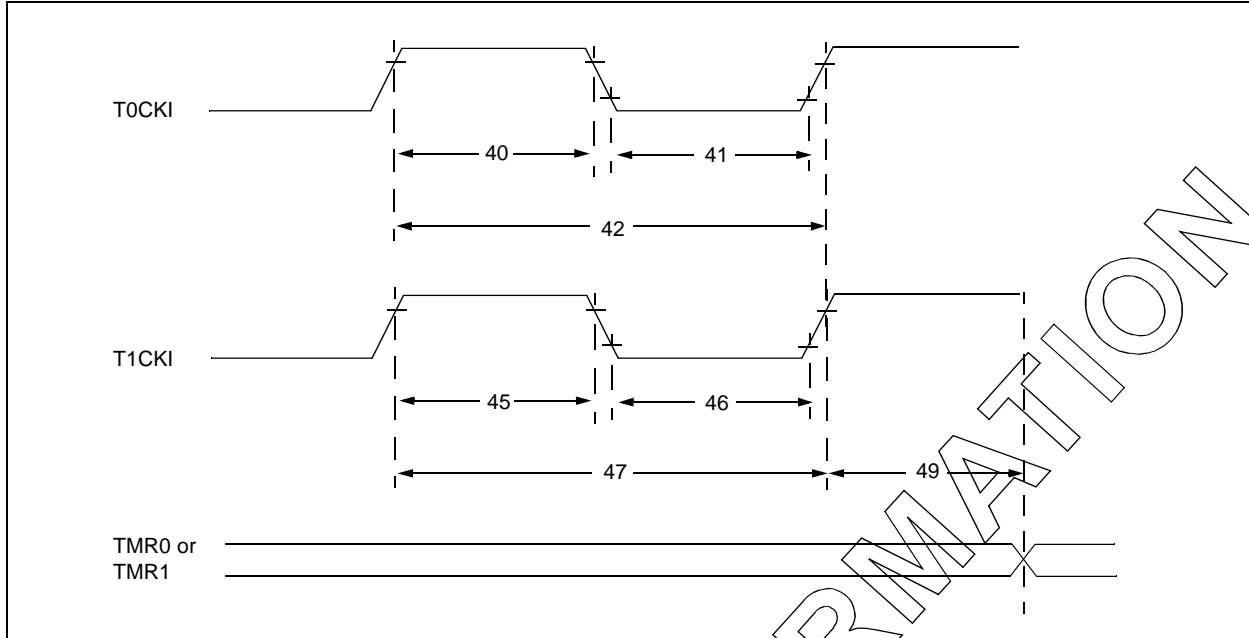
**2:** By design.

**3:** Period of the slower clock.

**4:** To ensure these voltage tolerances,  $V_{DD}$  and  $V_{SS}$  must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.

# PIC16F193X/LF193X

**FIGURE 28-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



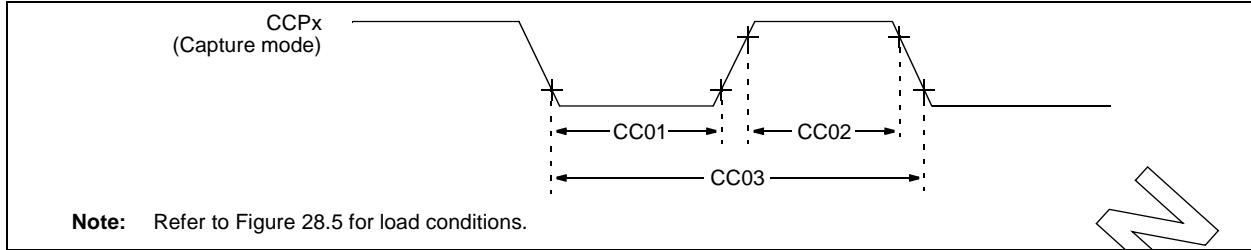
**TABLE 28-6: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	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	$0.5 T_{CY} + 20$	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	Tt1P	T1CKI Input Period	Synchronous	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	—	ns	
48	Ft1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

\* These parameters are characterized but not tested.

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

**FIGURE 28-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)**



**TABLE 28-7: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
CC01*	TcL	CCPx Input Low Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TcH	CCPx Input High Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TcP	CCPx Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)

\* These parameters are characterized but not tested.

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

**TABLE 28-8: PIC16F193X/LF193X A/D CONVERTER (ADC) CHARACTERISTICS:**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	—	$\pm 1$	LSb	$V_{REF} = 3.0\text{V}$
AD03	EDL	Differential Error	—	—	$\pm 1$	LSb	No missing codes $V_{REF} = 3.0\text{V}$
AD04	E0FF	Offset Error	—	—	$\pm 3$	LSb	$V_{REF} = 3.0\text{V}$
AD05	E0N	Gain Error	—	—	$\pm 3$	LSb	$V_{REF} = 3.0\text{V}$
AD06	VREF	Reference Voltage <sup>(3)</sup>	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD09*	IREF	VREF Input Current <sup>(3)</sup>	10	—	1000	$\mu\text{A}$	During VAIN acquisition. Based on differential of VHOLD to VAIN.
			—	—	10	$\mu\text{A}$	During A/D conversion cycle.

\* These parameters are characterized but not tested.

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

**Note 1:** Total Absolute Error includes integral, differential, offset and gain errors.

**Note 2:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**Note 3:** ADC VREF is from external VREF, VDD pin or FVREF, whichever is selected as reference input.

**Note 4:** When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

# PIC16F193X/LF193X

**TABLE 28-9: PIC16F193X/LF193X A/D CONVERSION REQUIREMENTS**

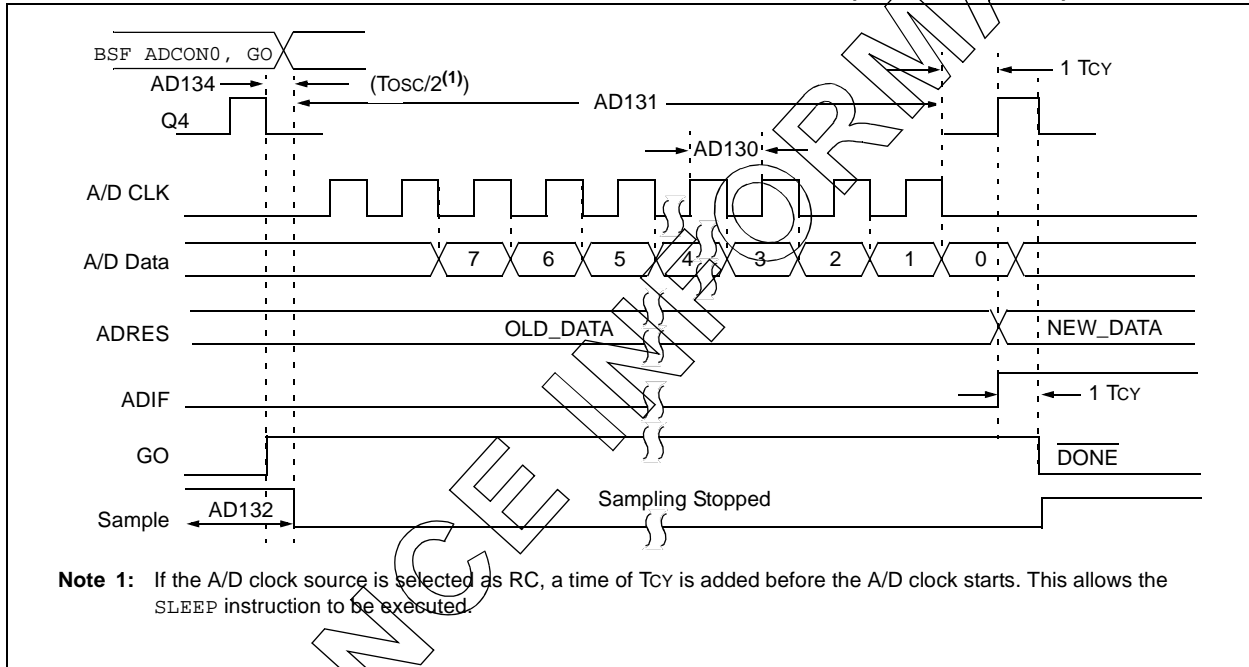
Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD130*	TAD	A/D Clock Period	1.0	—	9.0	$\mu\text{s}$	TOSC-based
		A/D Internal RC Oscillator Period	1.0	1.6	6.0	$\mu\text{s}$	ADCS<1:0> = 11 (ADRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time) <sup>(1)</sup>	—	10.5	—	TAD	Set GO/DONE bit to conversion complete
AD132*	TACQ	Acquisition Time	—	9.5	—	$\mu\text{s}$	

\* These parameters are characterized but not tested.

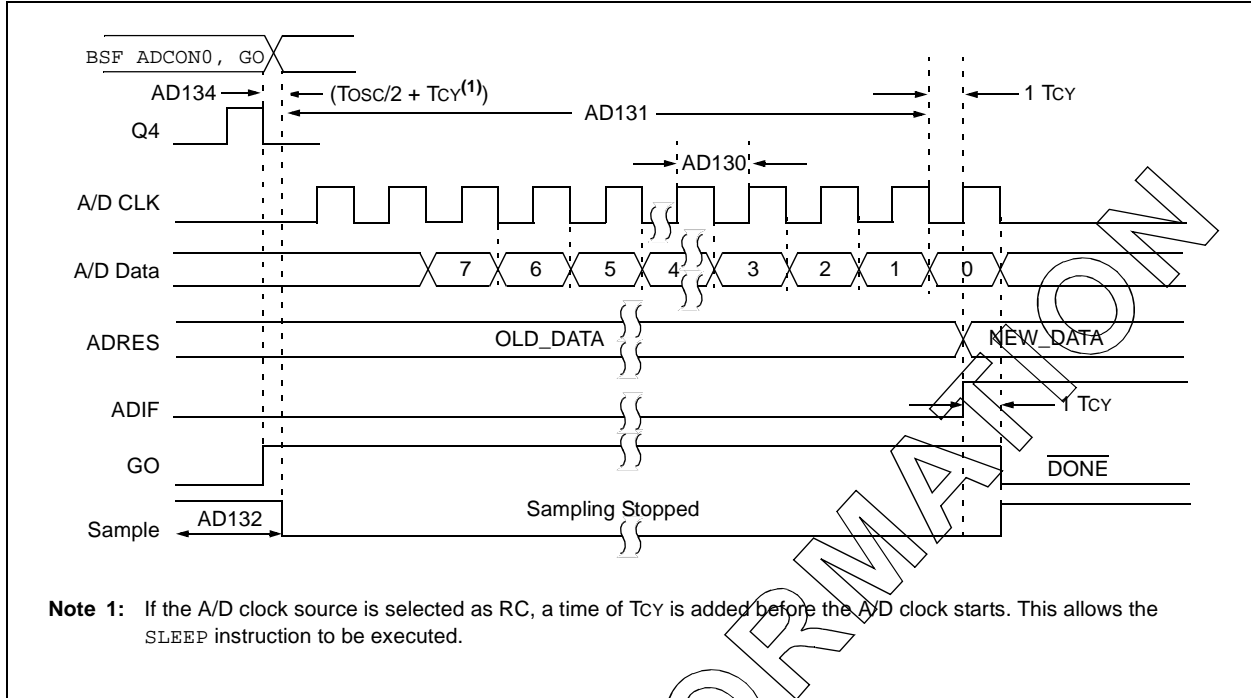
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The ADRES register may be read on the following Tcy cycle.

**FIGURE 28-12: PIC16F193X/LF193X A/D CONVERSION TIMING (NORMAL MODE)**



**FIGURE 28-13: PIC16F193X/LF193X A/D CONVERSION TIMING (SLEEP MODE)**



# PIC16F193X/LF193X

**TABLE 28-10: COMPARATOR SPECIFICATIONS**

Operating Conditions: 1.8V < VDD < 5.5V, -40°C < TA < +125°C (unless otherwise stated).							
Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
CM01	V <sub>IOFF</sub>	Input Offset Voltage	—	±7.5	±15	mV	
CM02	V <sub>ICM</sub>	Input Common Mode Voltage	0	—	V <sub>DD</sub>	V	
CM03	CMRR	Common Mode Rejection Ratio	55	—	—	dB	
CM04	T <sub>RESP</sub>	Response Time	—	150	400	ns	Note 1
CM05	T <sub>MC2OV</sub>	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<sub>DD</sub>/2, while the other input transitions from V<sub>SS</sub> to V<sub>DD</sub>.

**TABLE 28-11: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS**

Operating Conditions: 1.8V < VDD < 5.5V, -40°C < TA < +125°C (unless otherwise stated).							
Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
DAC01*	CLSB	Step Size <sup>(2)</sup>	—	V <sub>DD</sub> /32	—	V	
DAC02*	CACC	Absolute Accuracy	—	—	±1/2	LSb	
DAC03*	CR	Unit Resistor Value (R)	—	TBD	—	Ω	
DAC04*	CST	Settling Time <sup>(1)</sup>	—	—	10	µs	

\* These parameters are characterized but not tested.

**Legend:** TBD = To Be Determined

**Note 1:** Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

**TABLE 28-12: FIXED VOLTAGE REFERENCE (FVR) SPECIFICATIONS**

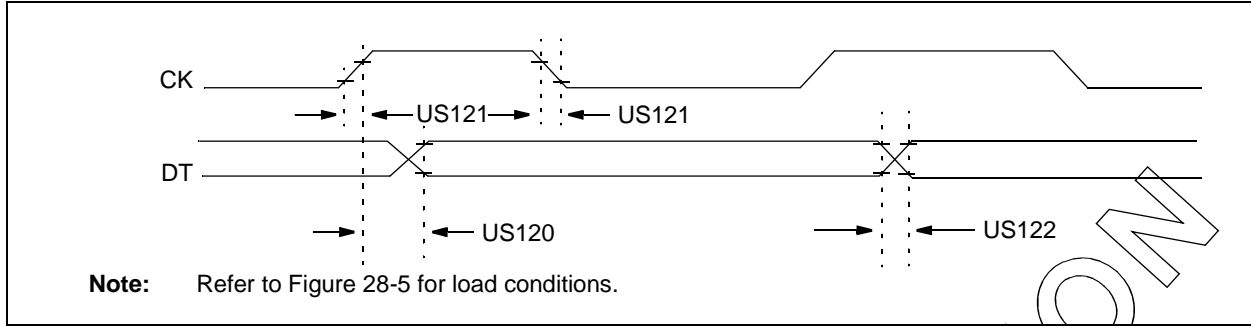
Operating Conditions: 1.8V < VDD < 5.5V, -40°C < TA < +125°C (unless otherwise stated).							
VR Voltage Reference Specifications			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C				
Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
VR01	V <sub>FVR</sub>	<b>Fixed Voltage Reference Voltage (calibrated)</b>	0.984 0.974 1.968 1.938 3.966 3.936	1.024  2.048  4.096	1.064 1.064 2.158 2.148 4.226 4.226	V	FVRV = 00 (1x), V <sub>DD</sub> ≥ 2V 125°C FVRV = 01 (2x), V <sub>DD</sub> ≥ 2.5V 125°C FVRV = 10 (4x), V <sub>DD</sub> ≥ 4.75V 125°C
VR02	TC <sub>VOUT</sub>	Voltage drift temperature coefficient	—	TBD	TBD	ppm/°C	
VR03	ΔV <sub>ROUT</sub> /ΔV <sub>DD</sub>	Voltage drift with respect to V <sub>DD</sub> regulation	—	TBD	—	µV/V	
VR04	T <sub>STABLE</sub>	Settling Time	—	TBD	TBD	µs	

**Legend:** TBD = To Be Determined

\* These parameters are characterized but not tested.

# PIC16F193X/LF193X

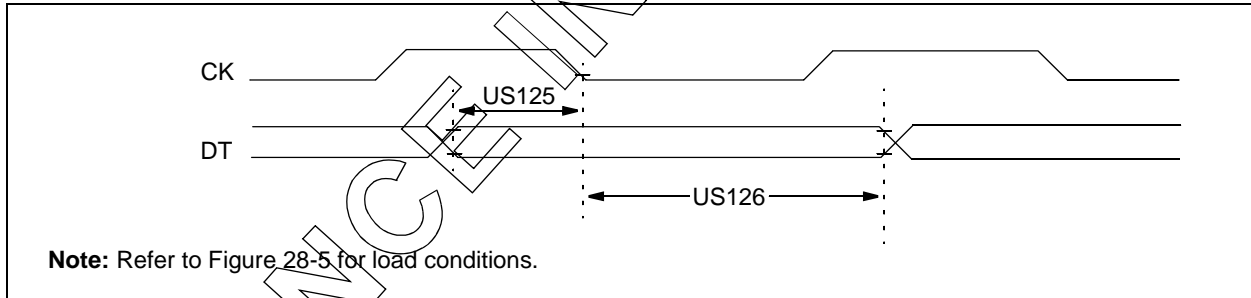
**FIGURE 28-14: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**



**TABLE 28-13: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)						
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$						
Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
US120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	3.0-5.5V	80	ns	
			1.8-5.5V	100	ns	
US121	TCKRF	Clock out rise time and fall time (Master mode)	3.0-5.5V	45	ns	
			1.8-5.5V	50	ns	
US122	TDTRF	Data-out rise time and fall time	3.0-5.5V	45	ns	
			1.8-5.5V	50	ns	

**FIGURE 28-15: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**

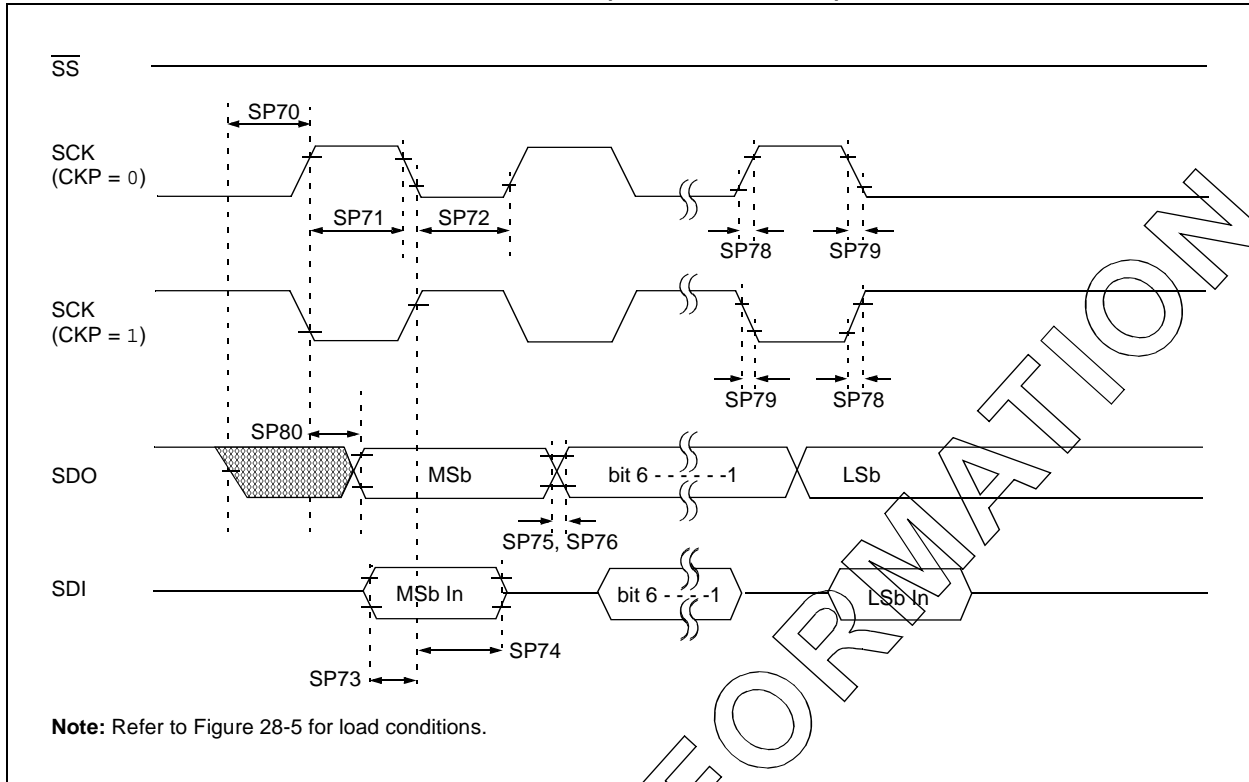


**TABLE 28-14: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

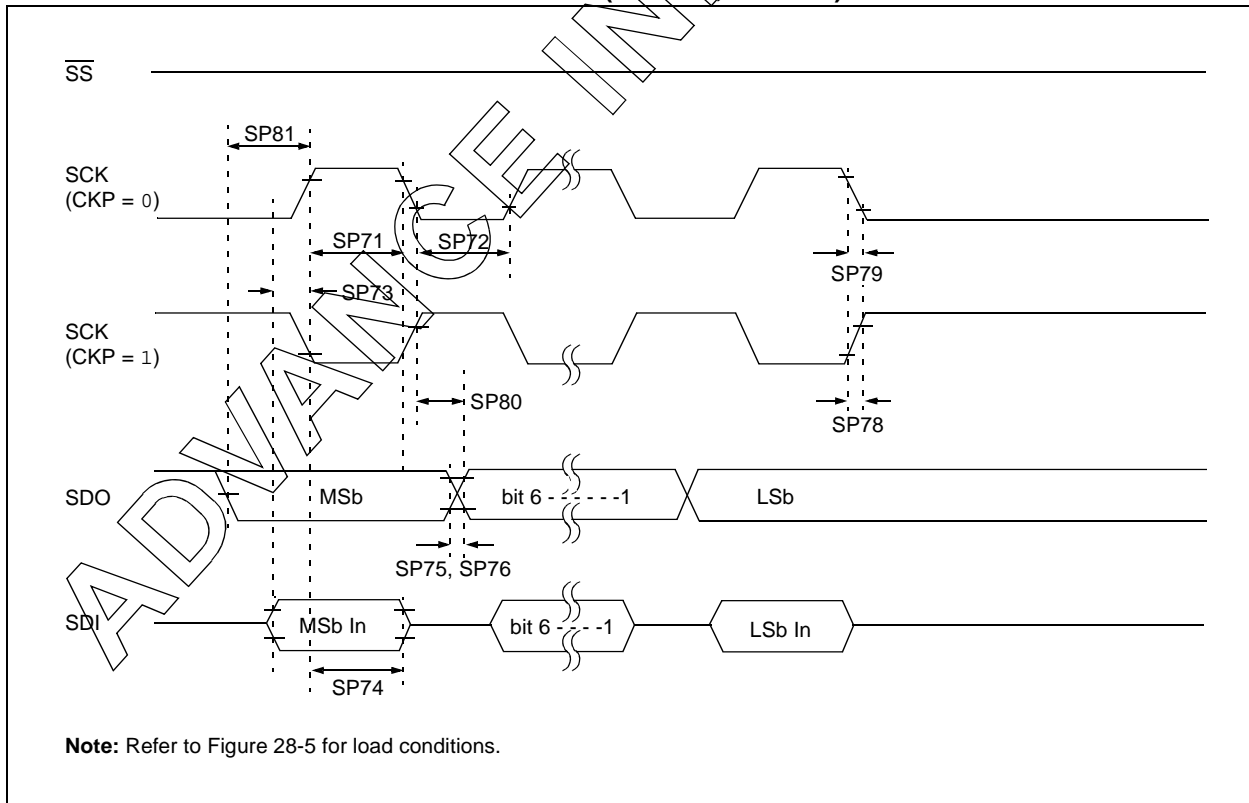
Standard Operating Conditions (unless otherwise stated)						
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$						
Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-hold before CK $\downarrow$ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold after CK $\downarrow$ (DT hold time)	15	—	ns	

# PIC16F193X/LF193X

**FIGURE 28-16: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**

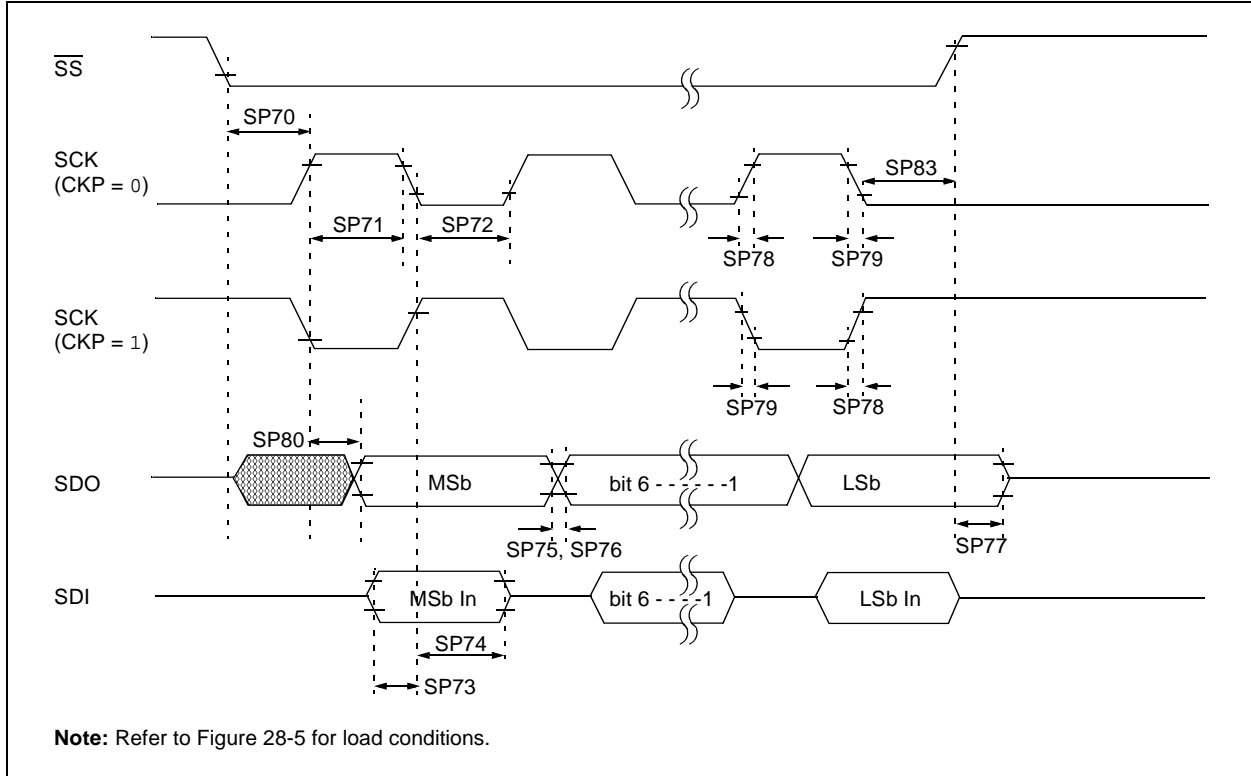


**FIGURE 28-17: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**

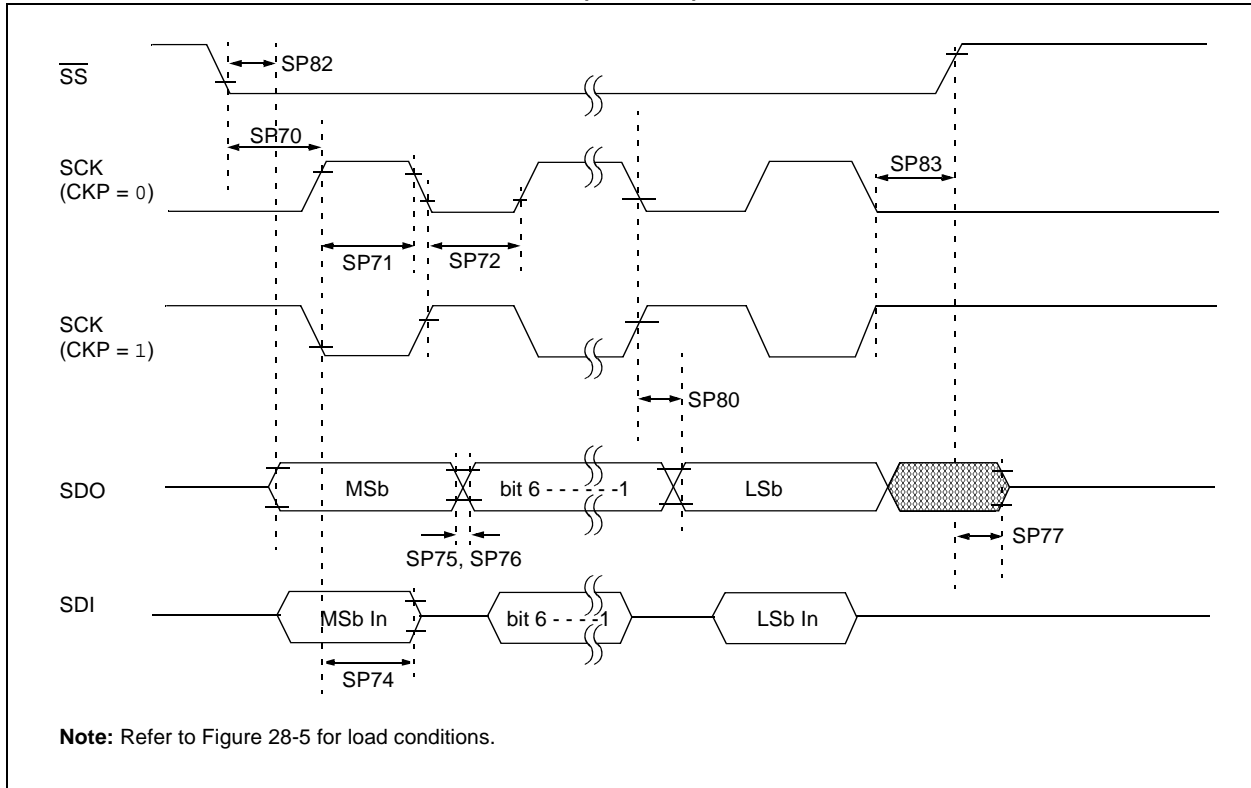




**FIGURE 28-18: SPI SLAVE MODE TIMING (CKE = 0)**



**FIGURE 28-19: SPI SLAVE MODE TIMING (CKE = 1)**



# PIC16F193X/LF193X

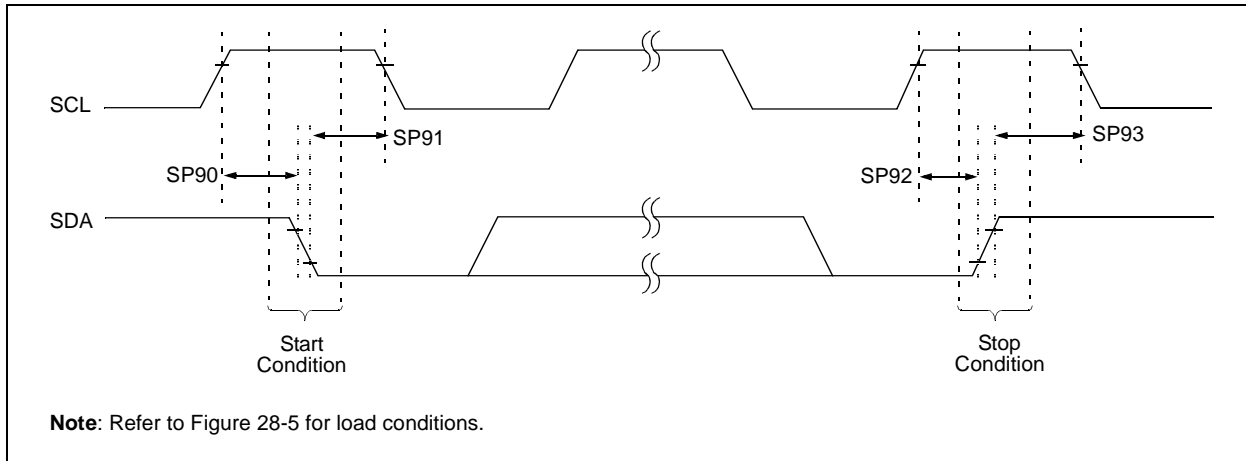
**TABLE 28-15: SPI MODE REQUIREMENTS**

Param No.	Symbol	Characteristic	Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2sch, TssL2scl	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input	Tcy	—	—	ns	
SP71*	Tsch	SCK input high time (Slave mode)	Tcy + 20	—	—	ns	
SP72*	Tscl	SCK input low time (Slave mode)	Tcy + 20	—	—	ns	
SP73*	TdIV2sch, TdIV2scl	Setup time of SDI data input to SCK edge	100	—	—	ns	
SP74*	Tsch2dIL, Tscl2dIL	Hold time of SDI data input to SCK edge	100	—	—	ns	
SP75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP76*	TdoF	SDO data output fall time	—	10	25	ns	
SP77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output high-impedance	10	—	50	ns	
SP78*	TscR	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP79*	TscF	SCK output fall time (Master mode)	—	10	25	ns	
SP80*	Tsch2doV, Tscl2doV	SDO data output valid after SCK edge	3.0-5.5V	—	—	50	ns
			1.8-5.5V	—	—	145	ns
SP81*	TdoV2sch, TdoV2scl	SDO data output setup to SCK edge	Tcy	—	—	ns	
SP82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
SP83*	Tsch2ssH, Tscl2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5Tcy + 40	—	—	ns	

\* These parameters are characterized but not tested.

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

**FIGURE 28-20: I<sup>2</sup>C™ BUS START/STOP BITS TIMING**

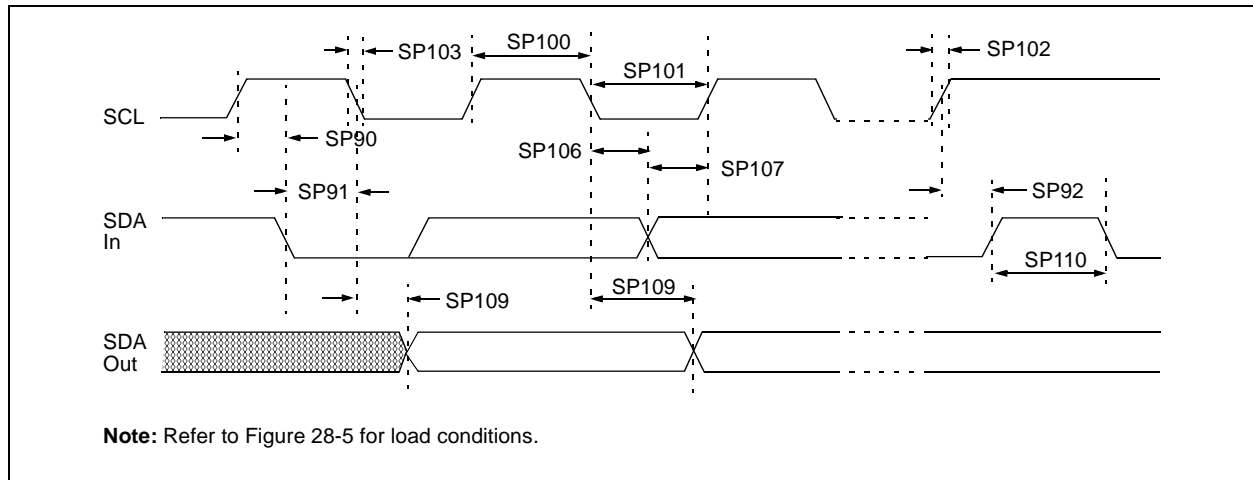


**TABLE 28-16: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS**

Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions	
SP90*	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

\* These parameters are characterized but not tested.

**FIGURE 28-21: I<sup>2</sup>C™ BUS DATA TIMING**



# PIC16F193X/LF193X

**TABLE 28-17: I<sup>2</sup>C™ BUS DATA REQUIREMENTS**

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
SP100*	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	1.5T <sub>CY</sub>	—			
SP101*	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	1.5T <sub>CY</sub>	—			
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1C <sub>B</sub>	250	ns	C <sub>B</sub> is specified to be from 10-400 pF
SP90*	TSU:STA	Start condition setup time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
SP91*	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	
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	<b>(Note 2)</b>
			400 kHz mode	100	—	ns	
SP92*	TSU:STO	Stop condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	<b>(Note 1)</b>
			400 kHz mode	—	—	ns	
SP110*	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	
SP111	C <sub>B</sub>	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<sup>2</sup>C™ bus device can be used in a Standard mode (100 kHz) I<sup>2</sup>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<sup>2</sup>C bus specification), before the SCL line is released.

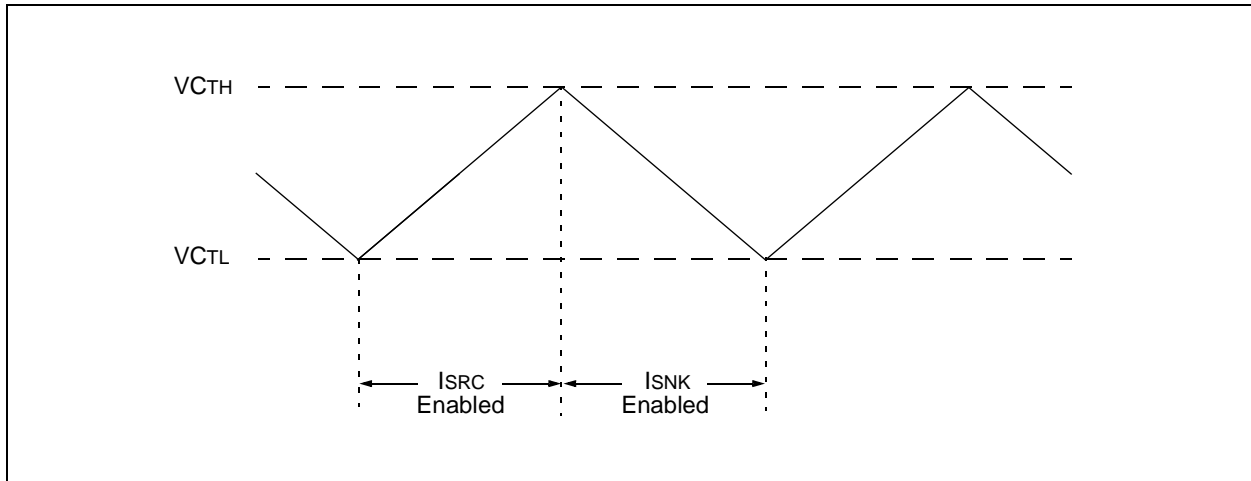
**TABLE 28-18: CAP SENSE OSCILLATOR SPECIFICATIONS**

Param. No.	Symbol	Characteristic		Min.	Typ†	Max.	Units	Conditions
CS01	ISRC	Current Source	High	—	-5.8	—	μs	
			Medium	—	-1.1	—	μs	
			Low	—	-0.2	—	μs	
CS02	ISNK	Current Sink	High	—	6.6	—	μs	
			Medium	—	1.3	—	μs	
			Low	—	0.24	—	μs	
CS03	VCTH	Cap Threshold	High	—	0.8	—	μs	
CS04	VCTL	Cap Threshold	Low	—	0.4	—	μs	

\* These parameters are characterized but not tested.

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

**FIGURE 28-22: CAP SENSE OSCILLATOR**



# PIC16F193X/LF193X

---

NOTES:

## 29.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Graphs and charts are not available at this time.

# PIC16F193X/LF193X

---

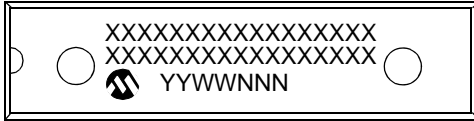
NOTES:



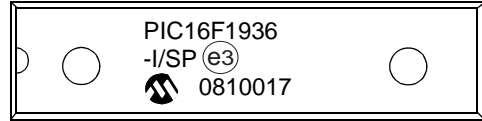
## 30.0 PACKAGING INFORMATION

### 30.1 Package Marking Information

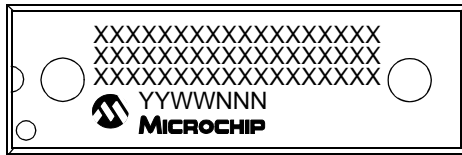
28-Lead SPDIP



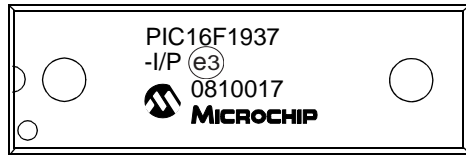
Example



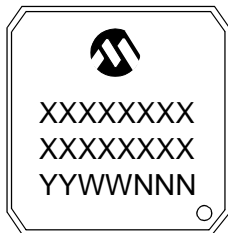
40-Lead PDIP



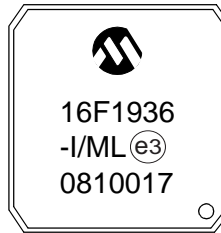
Example



28-Lead QFN



Example



<b>Legend:</b>	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
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.
<b>Note:</b>	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<sup>®</sup> 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.

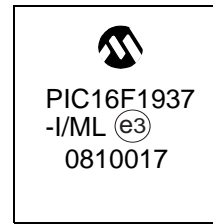
# PIC16F193X/LF193X

## Package Marking Information (Continued)

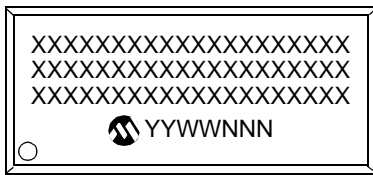
44-Lead QFN



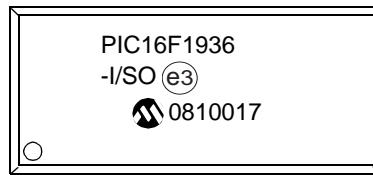
Example



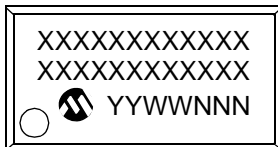
28-Lead SOIC



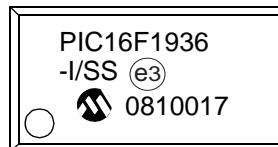
Example



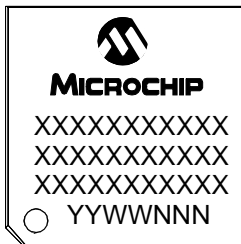
28-Lead SSOP



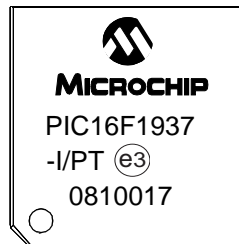
Example



44-Lead TQFP



Example



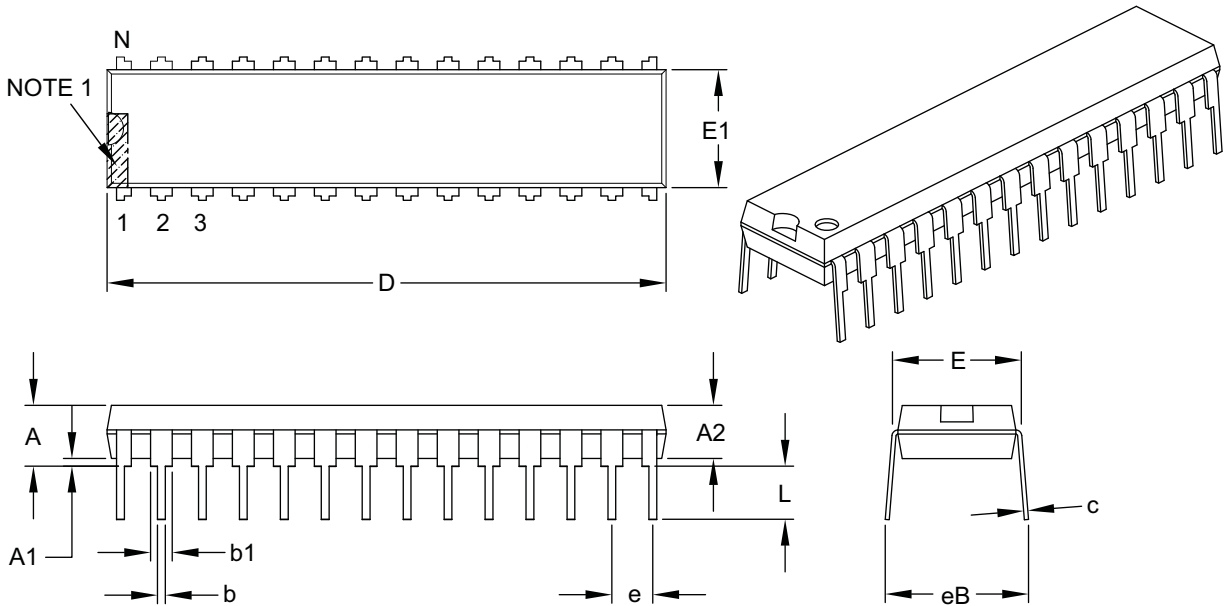
# PIC16F193X/LF193X

## 30.2 Package Details

The following sections give the technical details of the packages.

### 28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

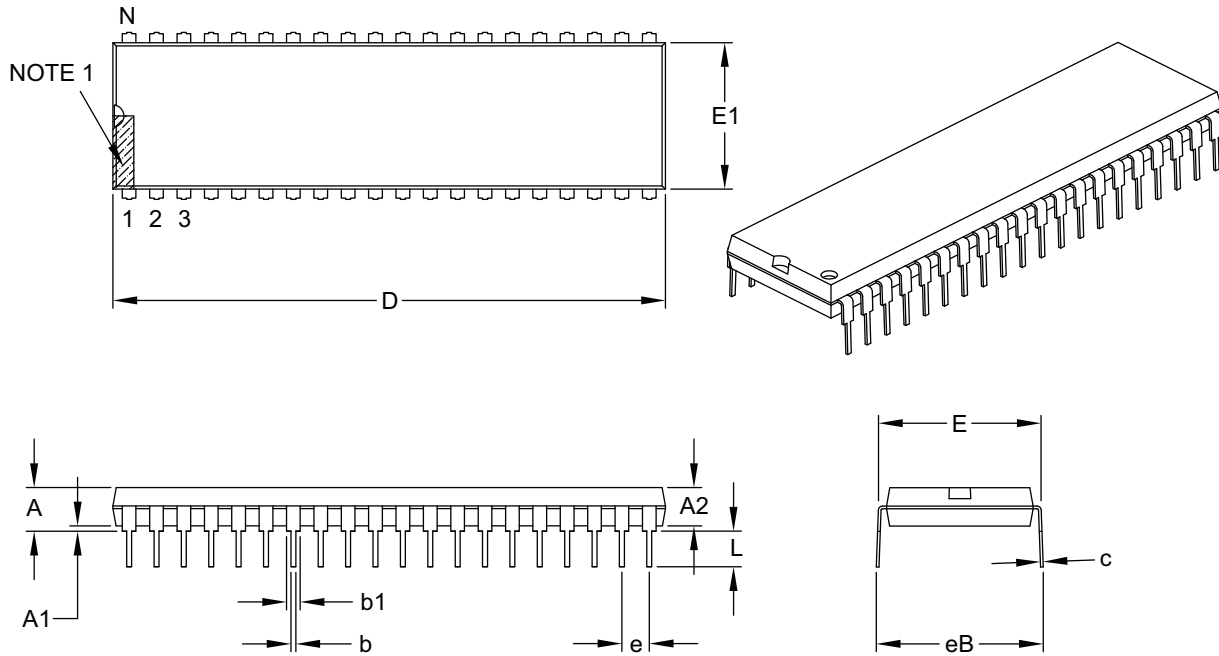
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

# PIC16F193X/LF193X

## 40-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packageing>



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	40		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.250
Molded Package Thickness	A2	.125	–	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.590	–	.625
Molded Package Width	E1	.485	–	.580
Overall Length	D	1.980	–	2.095
Tip to Seating Plane	L	.115	–	.200
Lead Thickness	c	.008	–	.015
Upper Lead Width	b1	.030	–	.070
Lower Lead Width	b	.014	–	.023
Overall Row Spacing §	eB	–	–	.700

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

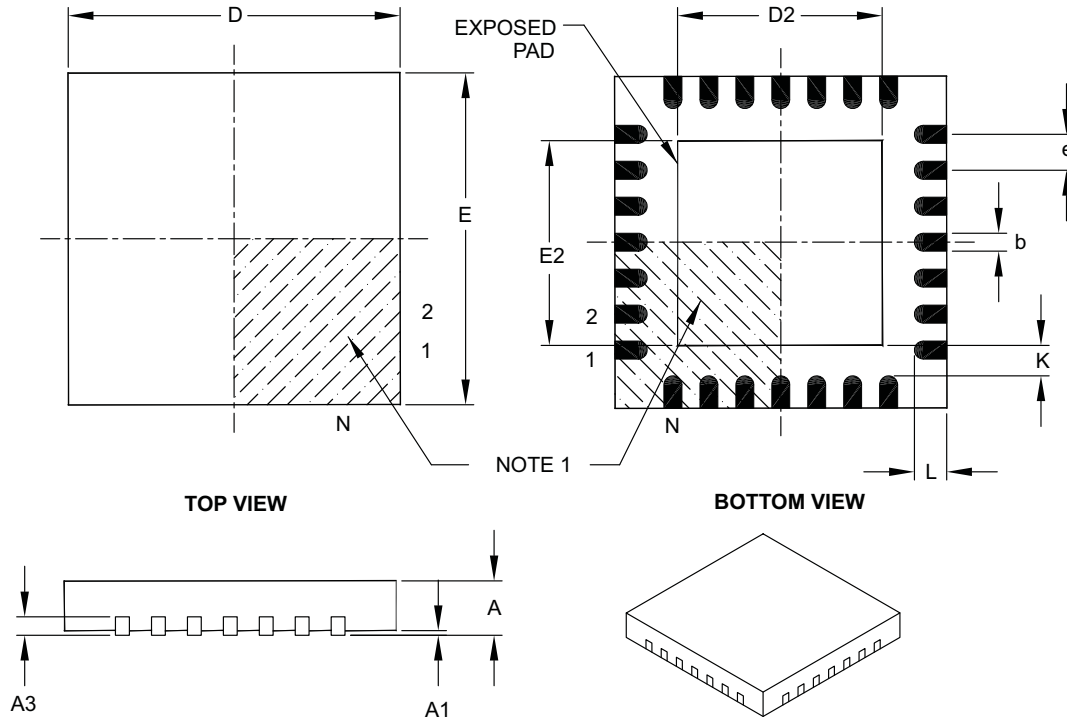
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-016B

# PIC16F193X/LF193X

## 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.20
Contact Width	b	0.23	0.30	0.35
Contact Length	L	0.50	0.55	0.70
Contact-to-Exposed Pad	K	0.20	–	–

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

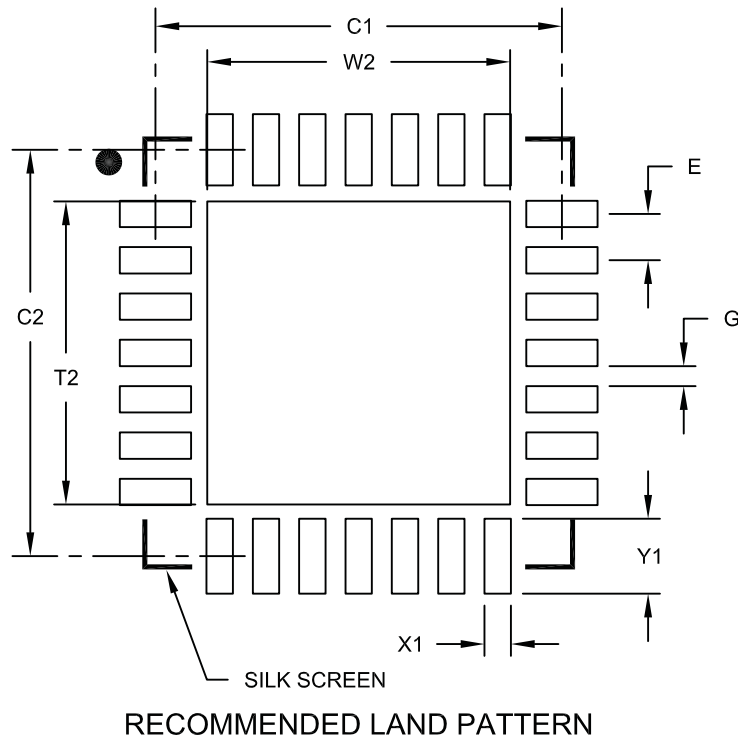
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

# PIC16F193X/LF193X

## 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

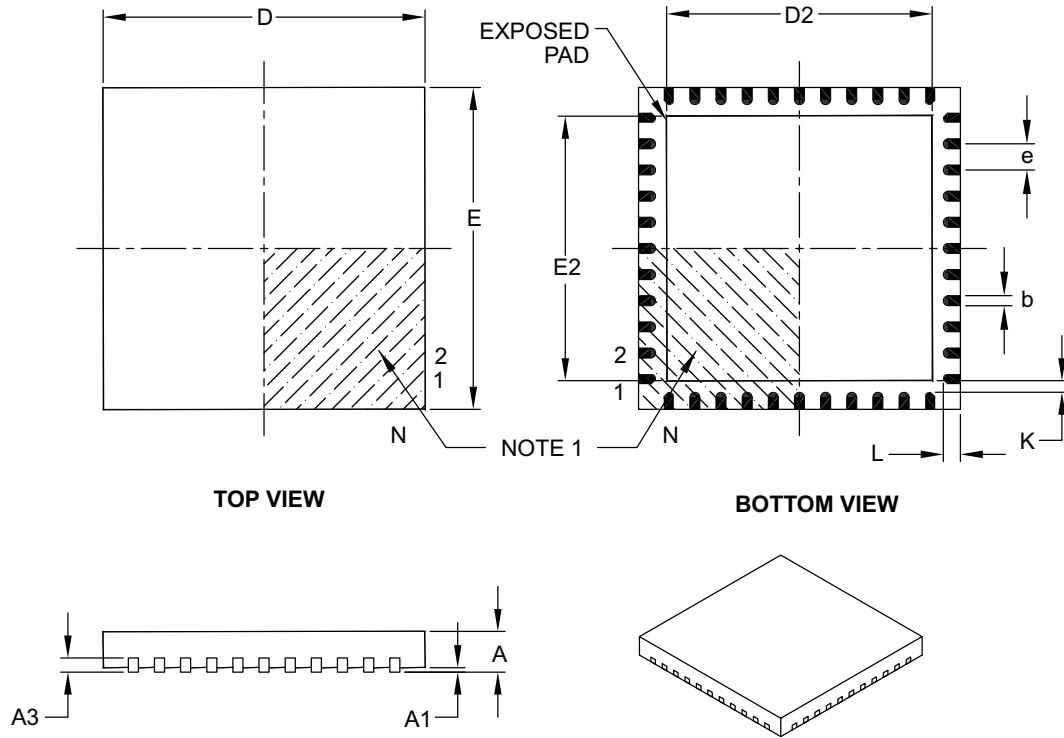
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

# PIC16F193X/LF193X

## 44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	44		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	8.00 BSC		
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D	8.00 BSC		
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

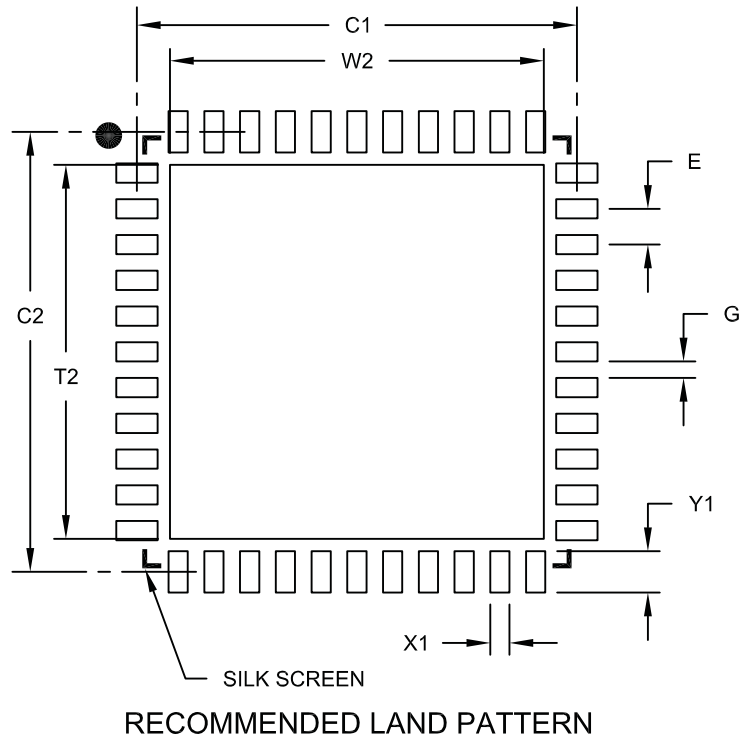
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

# PIC16F193X/LF193X

## 44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

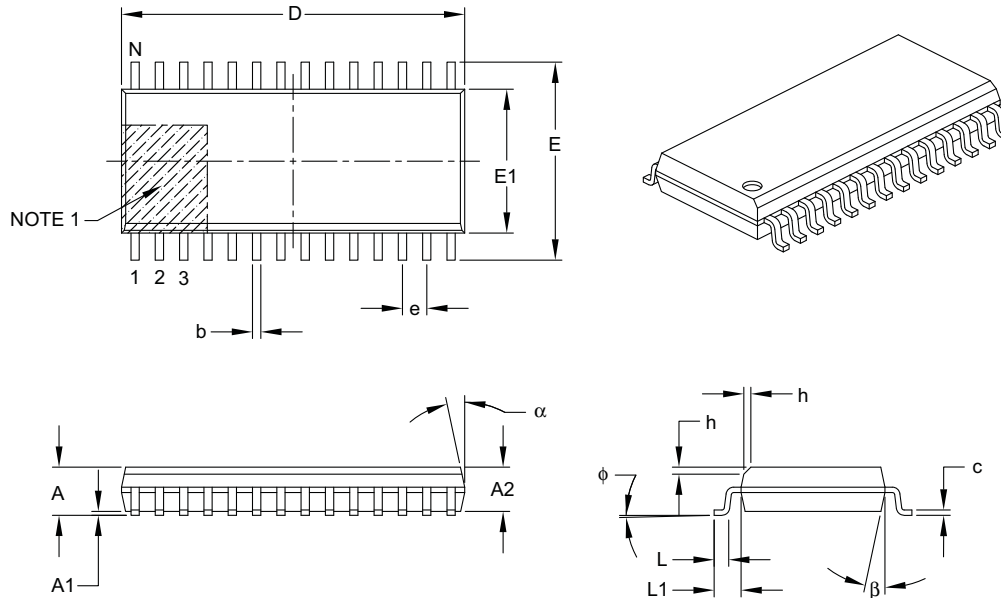
Microchip Technology Drawing No. C04-2103A



# PIC16F193X/LF193X

## 28-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1	1.40 REF		
Foot Angle Top	φ	0°	–	8°
Lead Thickness	c	0.18	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

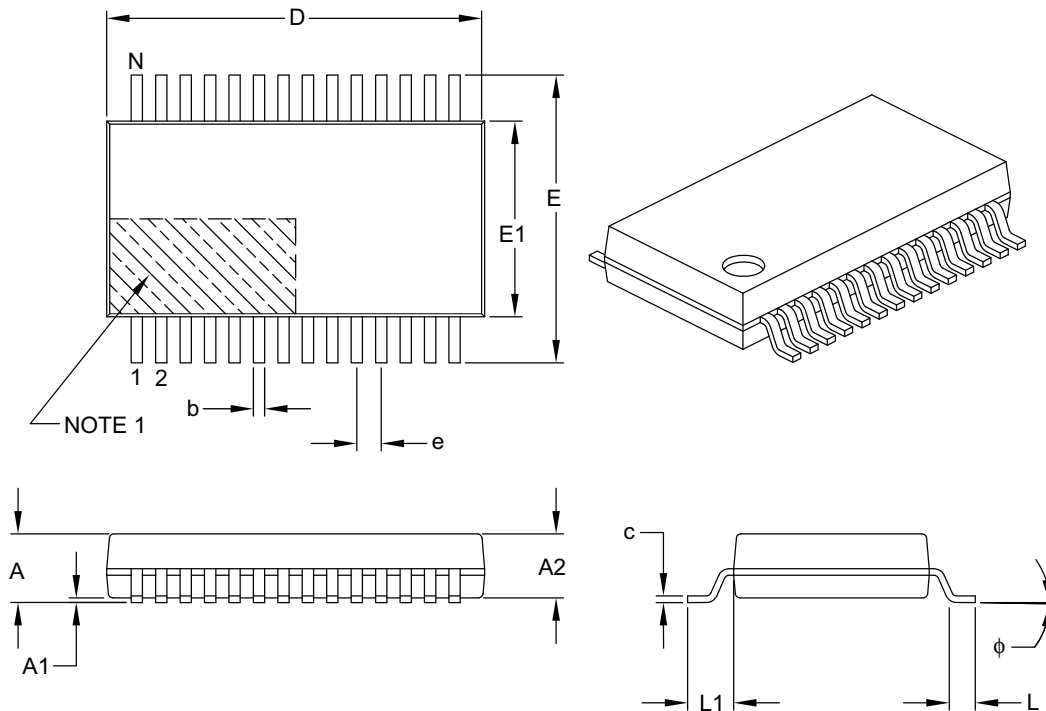
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B

# PIC16F193X/LF193X

## 28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	$\phi$	0°	4°	8°
Lead Width	b	0.22	–	0.38

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

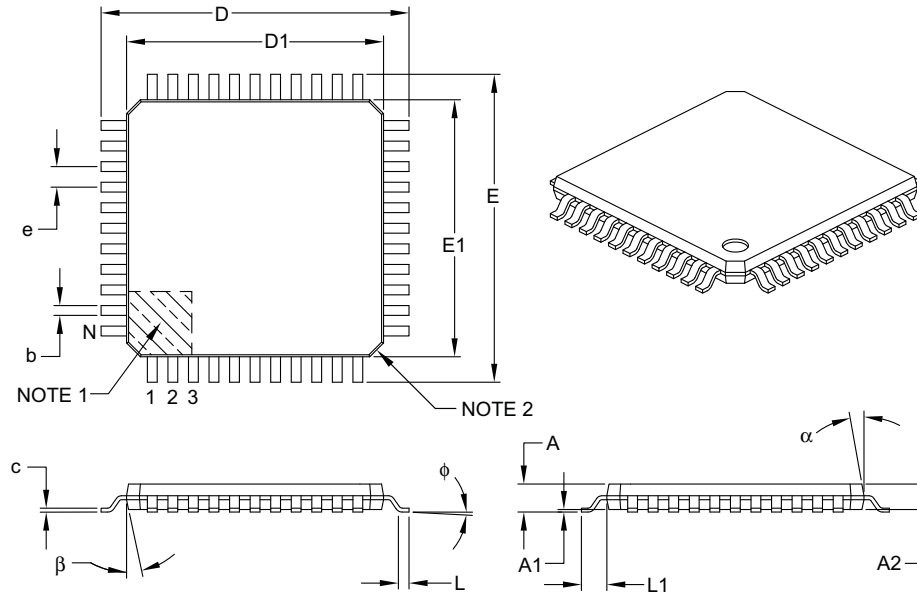
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC16F193X/LF193X

## 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	e	0.80 BSC		
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	$\phi$	0°	3.5°	7°
Overall Width	E	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	$\alpha$	11°	12°	13°
Mold Draft Angle Bottom	$\beta$	11°	12°	13°

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

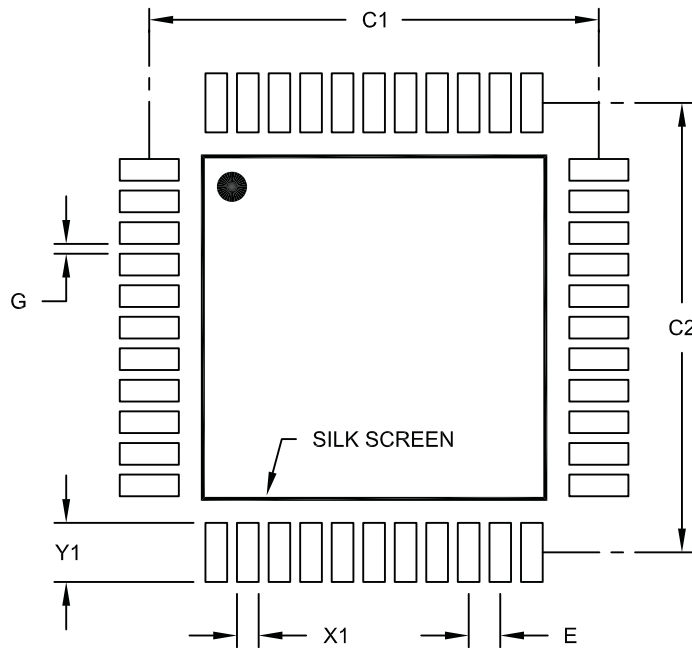
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

# PIC16F193X/LF193X

## 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packageing>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076A

# PIC16F193X/LF193X

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

Original release (12/2008)

## APPENDIX B: MIGRATING FROM OTHER PIC<sup>®</sup> DEVICES

This discusses some of the issues in migrating from other PIC<sup>®</sup> devices to the PIC16F193X/LF193X family of devices.

### B.1 PIC16F917 to PIC16F193X/LF193X

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F917	PIC16F1937
Max. Operating Speed	20 MHz	32 MHz
Max. Program Memory (Words)	8K	8K
Max. SRAM (Bytes)	368	512
A/D Resolution	10-bit	10-bit
Timers (8/16-bit)	2/1	4/1
Oscillator Modes	4	8
Brown-out Reset	Y	Y
Internal Pull-ups	RB<7:0>	RB<7:0>
Interrupt-on-change	RB<7:4>	RB<7:0>
Comparator	2	2
AUSART/EUSART	1/0	0/1
Extended WDT	Y	Y
Software Control Option of WDT/BOR	N	Y
INTOSC Frequencies	30 kHz - 8 MHz	500 kHz - 32 MHz
Clock Switching	Y	Y
Capacitive Sensing	N	Y
CCP/ECCP	2/0	2/3
Enhanced PIC16 CPU	N	Y
MSSP/SSP	0/1	1/0
LCD	Y	Y

# PIC16F193X/LF193X

---

NOTES:

## INDEX

### A

A/D	
Specifications.....	377, 378
Absolute Maximum Ratings .....	355
AC Characteristics	
Industrial and Extended .....	370
Load Conditions .....	369
ACKSTAT .....	310
ACKSTAT Status Flag .....	310
ADC .....	131
Acquisition Requirements .....	140
Associated registers.....	142
Block Diagram.....	131
Calculating Acquisition Time.....	140
Channel Selection.....	132
Configuration.....	132
Configuring Interrupt .....	136
Conversion Clock.....	132
Conversion Procedure .....	136
Internal Sampling Switch (Rss) Impedance.....	140
Interrupts.....	134
Operation .....	135
Operation During Sleep .....	135
Port Configuration .....	132
Reference Voltage (VREF).....	132
Source Impedance.....	140
Special Event Trigger.....	135
Starting an A/D Conversion .....	134
ADCON0 Register.....	36, 137
ADCON1 Register.....	36, 138
ADDFSR .....	341
ADDWFC .....	341
ADRESH Register.....	36
ADRESH Register (ADFM = 0) .....	138
ADRESH Register (ADFM = 1) .....	139
ADRESL Register (ADFM = 0).....	139
ADRESL Register (ADFM = 1).....	139
Alternate Pin Function.....	84
Analog-to-Digital Converter. See ADC	
ANSELA Register .....	86
ANSELB Register .....	91
ANSELD Register .....	97
ANSELE Register .....	101
APFCON Register.....	84
Assembler	
MPASM Assembler.....	352

### B

BAUDCON Register.....	224
BF .....	310, 312
BF Status Flag .....	310, 312
Block Diagram	
Capacitive Sensing .....	177
Block Diagrams	
(CCP) Capture Mode Operation .....	187
ADC .....	131
ADC Transfer Function .....	141
Analog Input Model.....	141, 147
CCP PWM.....	191
Clock Source.....	107
Comparator .....	144
Compare .....	189
Crystal Operation .....	110

Digital-to-Analog Converter (DAC) .....	152
EUSART Receive .....	214
EUSART Transmit.....	213
External RC Mode .....	111
Fail-Safe Clock Monitor (FSCM).....	119
Generic I/O Port.....	83
Interrupt Logic.....	69
LCD Bias Voltage Generation .....	249
LCD Clock Generation.....	248
MCLR Circuit .....	59
On-Chip Reset Circuit.....	57
Peripheral Interrupt Logic .....	70
PIC16F193X/LF193X .....	13
PWM (Enhanced) .....	195
Resonator Operation .....	110
Timer0 .....	157
Timer1 .....	161
Timer1 Gate.....	166, 167, 168
Timer2/4/6 .....	173
Voltage Reference.....	155
Voltage Reference Output Buffer Example .....	155
BORCON Register.....	63
BRA .....	342
Break Character (12-bit) Transmit and Receive .....	233
Brown-out Reset (BOR).....	62
Specifications .....	375
Timing and Characteristics .....	374

### C

C Compilers	
MPLAB C18.....	352
MPLAB C30.....	352
CALL.....	343
CALLW .....	343
Capacitive Sensing .....	177
Associated registers w/ Capacitive Sensing.....	181
Specifications .....	387
Capture Module. See Enhanced Capture/Compare/ PWM(ECCP)	
Capture/Compare/PWM .....	183
Capture/Compare/PWM (CCP) .....	185
Associated Registers w/ Capture .....	188
Associated Registers w/ Compare .....	190
Associated Registers w/ PWM .....	211
Capture Mode.....	187
CCPx Pin Configuration.....	187
Clock Selection .....	185
Compare Mode.....	189
CCPx Pin Configuration.....	189
Software Interrupt Mode .....	187, 189
Special Event Trigger .....	189
Timer1 Mode Selection.....	187, 189
Prescaler .....	187
PWM Mode.....	191
Duty Cycle .....	192
Effects of Reset .....	194
Example PWM Frequencies and Resolutions, 20 MHZ.....	193
Example PWM Frequencies and Resolutions, 32 MHZ.....	193
Example PWM Frequencies and Resolutions, 8 MHz .....	193
Operation in Sleep Mode.....	194
Resolution .....	193

# PIC16F193X/LF193X

Setup for Operation.....	194	DACCON1 (Digital-to-Analog Converter Control 1)	
System Clock Frequency Changes.....	194	Register .....	153
PWM Period.....	192	Data EEPROM Memory.....	321
Setup for PWM Operation.....	194	Associated Registers.....	332
CCP1CON Register.....	40, 41	Code Protection.....	332
CCPR1H Register.....	40, 41	Reading .....	325
CCPR1L Register.....	40, 41	Writing .....	325
CCPTMRS0 Register.....	185	Data Memory .....	24
CCPTMRS1 Register.....	186	DC and AC Characteristics.....	389
CCPxAS Register.....	204	DC Characteristics	
CCPxCON (ECCPx) Register.....	184	Extended and Industrial.....	365
Clock Accuracy with Asynchronous Operation .....	222	Industrial and Extended.....	358
Clock Sources		Development Support.....	351
External Modes .....	109	Device Configuration .....	125
EC.....	109	Code Protection.....	129
HS.....	110	Configuration Word.....	125
LP.....	110	User ID .....	129
OST.....	109	Device Overview.....	13
RC.....	111	Digital-to-Analog Converter (DAC) .....	151
XT.....	110	Associated Registers.....	154
Internal Modes .....	111	Effects of a Reset .....	151
Frequency Selection .....	115	Operation During Sleep.....	151
HFINTOSC.....	112	Specifications .....	380
Internal Oscillator Clock Switch Timing.....	115		
INTOSC .....	111	<b>E</b>	
INTOSCIO.....	111	ECCP/CCP. See Enhanced Capture/Compare/PWM	
LFINTOSC.....	112	EEADR Registers .....	321
MFINTOSC.....	112	EEADRH Registers.....	321
Clock Switching.....	117	EEADRL Register.....	322
CMOUT Register.....	149	EEADRL Registers .....	321
CMxCON0 Register.....	148	EECON1 Register.....	321, 323
CMxCON1 Register.....	149	EECON2 Register.....	321, 324
Code Examples		EEDATH Register.....	322
A/D Conversion.....	136	EEDATL Register .....	322
Changing Between Capture Prescalers.....	187	EEPROM Data Memory	
Initializing PORTA.....	85	Avoiding Spurious Write .....	332
Initializing PORTB.....	89	Write Verify .....	332
Initializing PORTC.....	93	Effects of Reset	
Initializing PORTD.....	96	PWM mode.....	194
Initializing PORTE.....	101	Electrical Specifications.....	355
Write Verify .....	332	Enhanced Capture/Compare/PWM	
Writing to Flash Program Memory .....	330	Timer Resources .....	184
Comparator		Enhanced Capture/Compare/PWM (ECCP).....	184
Associated Registers .....	150	Enhanced PWM Mode.....	195
Operation .....	143	Auto-Restart .....	205
Comparator Module .....	143	Auto-shutdown.....	203
Cx Output State Versus Input Conditions .....	145	Direction Change in Full-Bridge Output Mode..	201
Comparator Specifications.....	380	Full-Bridge Application.....	199
Comparator Voltage Reference (CVREF)		Full-Bridge Mode .....	199
Associated Registers .....	156	Half-Bridge Application .....	198
Comparators		Half-Bridge Application Examples .....	206
C2OUT as T1 Gate.....	163	Half-Bridge Mode.....	198
Compare Module. See Enhanced Capture/ Compare/PWM (ECCP)		Output Relationships (Active-High and Active-Low).....	196
CONFIG1 Register.....	126	Output Relationships Diagram.....	197
CONFIG2 Register.....	128	Programmable Dead Band Delay.....	206
Core Registers .....	49	Shoot-through Current.....	206
CPSCON0 Register .....	180	Start-up Considerations.....	203
CPSCON1 Register .....	181	Specifications .....	377
Customer Change Notification Service .....	413	Enhanced Mid-range CPU.....	14
Customer Notification Service.....	413	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) .....	213
Customer Support.....	413	Errata .....	11
		EUSART .....	213
<b>D</b>		Associated Registers	
DACCON0 (Digital-to-Analog Converter Control 0) Register..	153	Baud Rate Generator .....	226



# PIC16F193X/LF193X

Asynchronous Mode .....	215	Sleep Operation.....	315
12-bit Break Transmit and Receive .....	233	Stop Condition Timing .....	314
Associated Registers		INDF Register35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48	
Receive.....	221	Indirect Addressing, INDF and FSR Registers .....	53
Transmit.....	217	Instruction Format.....	338
Auto-Wake-up on Break .....	231	Instruction Set.....	337
Baud Rate Generator (BRG) .....	225	ADDLW.....	341
Clock Accuracy .....	222	ADDWF .....	341
Receiver.....	218	ADDWFC .....	341
Setting up 9-bit Mode with Address Detect.....	220	ANDLW.....	341
Transmitter.....	215	ANDWF .....	341
Baud Rate Generator (BRG)		BRA .....	342
Auto Baud Rate Detect .....	230	CALL.....	343
Baud Rate Error, Calculating.....	225	CALLW .....	343
Baud Rates, Asynchronous Modes .....	227	LSLF .....	345
Formulas.....	226	LSRF .....	345
High Baud Rate Select (BRGH Bit) .....	225	MOVF .....	345
Synchronous Master Mode .....	234, 238	MOVIW .....	346
Associated Registers		MOVLB .....	346
Receive.....	237	MOVWI .....	347
Transmit.....	235	OPTION.....	347
Reception.....	236	RESET .....	347
Transmission .....	234	SUBWFB .....	349
Synchronous Slave Mode		TRIS .....	350
Associated Registers		BCF .....	342
Receive.....	239	BSF .....	342
Transmit.....	238	BTFSC .....	342
Reception.....	239	BTFSS .....	342
Transmission .....	238	CALL.....	343
Extended Instruction Set		CLRF .....	343
ADDFSR .....	341	CLRW .....	343
		CLRWDT .....	343
<b>F</b>		COMF .....	343
Fail-Safe Clock Monitor.....	119	DECF .....	343
Fail-Safe Condition Clearing .....	119	DECFSZ .....	344
Fail-Safe Detection .....	119	GOTO .....	344
Fail-Safe Operation.....	119	INCF .....	344
Reset or Wake-up from Sleep.....	119	INCFSZ .....	344
Firmware Instructions.....	337	IORLW.....	344
Fixed Voltage Reference (FVR)		IORWF.....	344
Specifications.....	380	MOVLW .....	346
Flash Program Memory .....	321	MOVWF .....	346
Erasing.....	328	NOP .....	347
Writing.....	328	RETFIE .....	348
FSR Register35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48		RETLW .....	348
FVRCON (Fixed Voltage Reference Control) Register.....	156	RETURN.....	348
		RLF .....	348
<b>I</b>		RRF .....	349
I <sup>2</sup> C Mode (MSSP)		SLEEP.....	349
Acknowledge Sequence Timing.....	314	SUBLW .....	349
Bus Collision		SUBWF .....	349
During a Repeated Start Condition.....	318	SWAPF .....	350
During a Stop Condition.....	319	XORLW .....	350
Effects of a Reset.....	315	XORWF .....	350
I <sup>2</sup> C Clock Rate w/BRG.....	320	INTCON Register.....	73
Master Mode		Internal Oscillator Block	
Operation .....	306	INTOSC	
Reception.....	312	Specifications .....	371
Start Condition Timing .....	308, 309	Internal Sampling Switch (Rss) Impedance .....	140
Transmission .....	310	Internet Address .....	413
Multi-Master Communication, Bus Collision		Interrupt-On-Change.....	103
and Arbitration .....	315	Associated Registers.....	105
Multi-Master Mode .....	315	Interrupts .....	69
Read/Write Bit Information (R/W Bit).....	291	ADC .....	136
Slave Mode		Associated registers w/ Interrupts .....	80
Transmission .....	296		

# PIC16F193X/LF193X

TMR1 .....	165
INTOSC Specifications .....	371
IOCBF Register .....	104
IOCBN Register .....	104
IOCBP Register .....	104

## L

LATA Register .....	85, 93
LATB Register .....	90
LATD Register .....	96
LATE Register .....	99

## LCD

Associated Registers .....	272
Bias Voltage Generation .....	249, 250
Clock Source Selection .....	248
Configuring the Module .....	271
Disabling the Module .....	271
Frame Frequency .....	254
Interrupts .....	267
LCDCON Register .....	241
LCDPS Register .....	241
Multiplex Types .....	254
Operation During Sleep .....	269
Pixel Control .....	254
Prescaler .....	248
Segment Enables .....	254
Waveform Generation .....	256
LCDCON Register .....	241, 243
LCDCST Register .....	246
LCDDATAX Registers .....	247, 252
LCDPS Register .....	241, 244
LP Bits .....	248
LCDREF Register .....	245
LCDRL Register .....	252
LCDSEn Registers .....	247
Liquid Crystal Display (LCD) Driver .....	241
Load Conditions .....	369
LSLF .....	345
LSRF .....	345

## M

Master Synchronous Serial Port. See MSSP	
MCLR .....	59
Internal .....	59
Memory Organization .....	21
Data .....	24
Program .....	21
Microchip Internet Web Site .....	413
Migrating from other PIC Microcontroller Devices .....	403
MOVIW .....	346
MOVLB .....	346
MOVWI .....	347
MPLAB ASM30 Assembler, Linker, Librarian .....	352
MPLAB ICD 2 In-Circuit Debugger .....	353
MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator .....	353
MPLAB Integrated Development Environment Software ..	351
MPLAB PM3 Device Programmer .....	353
MPLAB REAL ICE In-Circuit Emulator System .....	353
MPLINK Object Linker/MPLIB Object Librarian .....	352
MSSP .....	273
SSPBUF Register .....	283
SSPSR Register .....	283

## O

OPCODE Field Descriptions .....	337
OPTION .....	347
OPTION Register .....	51, 159
OSCCON Register .....	108
Oscillator	
Associated Registers .....	120
Oscillator Module .....	107
EC .....	107
HFINTOSC .....	107
HS .....	107
INTOSC .....	107
LFINTOSC .....	107
LP .....	107
MFINTOSC .....	107
RC .....	107
XT .....	107
Oscillator Parameters .....	371
Oscillator Specifications .....	370
Oscillator Start-up Timer (OST) Specifications .....	375
Oscillator Switching	
Fail-Safe Clock Monitor .....	119
Two-Speed Clock Start-up .....	117
OSCCON Register .....	113
OSCTUNE Register .....	114

## P

P1A/P1B/P1C/P1D. See Enhanced Capture/ Compare/PWM (ECCP) .....	195
Packaging .....	391
Marking .....	391, 392
PDIP Details .....	393
PCL and PCLATH .....	52
PCL Register 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48	
PCLATH Register 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48	
PCON Register .....	36, 65
PICSTART Plus Development Programmer .....	354
PIE1 Register .....	36, 74
PIE2 Register .....	36, 75
PIE3 Register .....	76
Pin Diagram	
PIC16F1933/1936/1938, PICLF1933/1936/1938, 28-pin PDIP/SOIC/SSOP .....	3
PIC16F1933/1936/1938, PICLF1933/1936/1938, 28-pin QFN .....	4
PICF1934/1937/1939, PICLF1934/1937/1939, 44-pin QFN .....	7
PICF1934/1937/1939, PICLF1934/1937/1939, 44-pin TQFP .....	8
PICF1934/1937/1939, PICLF1934/1937/1939, 40-pin PDIP .....	6
Pinout Descriptions	
PIC16F193X/PIC16LF193X .....	15
PIR1 Register .....	35, 77
PIR2 Register .....	35, 78
PIR3 Register .....	79
PORTA .....	85
ANSELA Register .....	86
Associated Registers .....	88
PORTA Register .....	35, 37
Specifications .....	373
PORTA Register .....	85
PORTB .....	89

# PIC16F193X/LF193X

Additional Pin Functions		ADCON0 (ADC Control 0) .....	137
Weak Pull-up .....	89	ADCON1 (ADC Control 1) .....	138
ANSELB Register .....	91	ADRESH (ADC Result High) with ADFM = 0) .....	138
Associated Registers .....	92	ADRESH (ADC Result High) with ADFM = 1) .....	139
Interrupt-on-Change .....	89	ADRESL (ADC Result Low) with ADFM = 0) .....	139
P1B/P1C/P1D. See Enhanced Capture/ Compare/PWM+ (ECCP+) .....	89	ADRESL (ADC Result Low) with ADFM = 1) .....	139
Pin Descriptions and Diagrams .....	92	ANSELA (PORTA Analog Select) .....	86
PORTB Register .....	35, 37	ANSELB (PORTB Analog Select) .....	91
PORTB Register .....	90	ANSELD (PORTD Analog Select) .....	97
PORTC .....	93	ANSELE (PORTE Analog Select) .....	101
Associated Registers .....	95	APFCON (Alternate Pin Function Control) .....	84
P1A. See Enhanced Capture/Compare/ PWM+ (ECCP+) .....	93	BAUDCON (Baud Rate Control) .....	224
Pin Descriptions and Diagrams .....	95	BORCON (Brown-out Reset Control) .....	63
PORTC Register .....	35, 37	CCPTMRS0 (CCP Timers Control 0) .....	185
Specifications .....	373	CCPTMRS1 (CCP Timers Control 1) .....	186
PORTC Register .....	93	CCPxAS (CCPx Auto-Shutdown Control) .....	204
PORTD .....	96	CCPxCON (ECCPx Control) .....	184
Additional Pin Functions		CMOUT (Comparator Output) .....	149
ANSELD Register .....	97	CMxCON0 (Cx Control) .....	148
Associated Registers .....	98	CMxCON1 (Cx Control 1) .....	149
P1B/P1C/P1D. See Enhanced Capture/ Compare/PWM+ (ECCP+) .....	96	Configuration Word 1 .....	126
Pin Descriptions and Diagrams .....	98	Configuration Word 2 .....	128
PORTD Register .....	35, 37	CPSCON0 (Capacitive Sensing Control Register 0) .....	180
PORTD Register .....	96	CPSCON1 (Capacitive Sensing Control Register 1) .....	181
PORTE .....	99	DACCON0 .....	153
ANSELE Register .....	101	DACCON1 .....	153
Associated Registers .....	102	EEADRL (EEPROM Address) .....	322
Pin Descriptions and Diagrams .....	102	EECON1 (EEPROM Control 1) .....	323
PORTE Register .....	35, 37	EECON2 (EEPROM Control 2) .....	324
PORTE Register .....	99	EEDATH (EEPROM Data) .....	322
Power-Down Mode (Sleep) .....	333	EEDATL (EEPROM Data) .....	322
Associated Registers .....	334	FVRCON .....	156
Power-on Reset .....	59	INTCON (Interrupt Control) .....	73
Power-up Time-out Sequence .....	64	IOCBF (Interrupt-on-Change Flag) .....	104
Power-up Timer (PWRT) .....	59	IOCBN (Interrupt-on-Change Negative Edge) .....	104
Specifications .....	375	IOCBP (Interrupt-on-Change Positive Edge) .....	104
PR2 Register .....	35, 43	LATA (Data Latch PORTA) .....	85
Precision Internal Oscillator Parameters .....	371	LATB (Data Latch PORTB) .....	90
Program Memory .....	21	LATC (Data Latch PORTC) .....	93
Map and Stack (PIC16F1933/LF1933, PIC16F1934/LF1934) .....	22	LATD (Data Latch PORTD) .....	96
Map and Stack (PIC16F1936/LF1936, PIC16F1937/LF1937) .....	22	LATE (Data Latch PORTE) .....	99
Map and Stack (PIC16F1938/LF1938, PIC16F1939/LF1939) .....	23	LCDCON (LCD Control) .....	243
Programming, Device Instructions .....	337	LCDCST (LCD Contrast Control) .....	246
PSTRxCON Register .....	208	LCDDATAx (LCD Data) .....	247, 252
Pulse Steering .....	208	LCDPS (LCD Phase) .....	244
PWM (ECCP Module)		LCDREF (LCD Reference Voltage Control) .....	245
Pulse Steering .....	208	LCDRL (LCD Reference Voltage Control) .....	252
Steering Synchronization .....	210	LCDSEn (LCD Segment Enable) .....	247
PWM Mode. See Enhanced Capture/Compare/PWM .....	195	OPTION_REG (OPTION) .....	51, 159
PWMxCON Register .....	207	OSCCON (Oscillator Control) .....	108
<b>R</b>		OSCSTAT (Oscillator Status) .....	113
RCREG .....	220	OSCTUNE (Oscillator Tuning) .....	114
RCREG Register .....	38	PCON (Power Control Register) .....	65
RCSTA Register .....	38, 223	PCON (Power Control) .....	65
Reader Response .....	414	PIE1 (Peripheral Interrupt Enable 1) .....	74
Read-Modify-Write Operations .....	337	PIE2 (Peripheral Interrupt Enable 2) .....	75
Register		PIE3 (Peripheral Interrupt Enable 3) .....	76
RCREG Register .....	230	PIR1 (Peripheral Interrupt Register 1) .....	77
Registers		PIR2 (Peripheral Interrupt Request 2) .....	78
		PIR3 (Peripheral Interrupt Request 3) .....	79
		PORTA .....	85
		PORTB .....	90
		PORTC .....	93
		PORTD .....	96
		PORTE .....	99

# PIC16F193X/LF193X

PSTRxCON (Pulse Steering Control) .....	208	SSPOV .....	312
PWMxCON (Enhanced PWM Control) .....	207	SSPOV Status Flag .....	312
RCSTA (Receive Status and Control).....	223	SSPSTAT Register .....	39, 276
Special Function, Summary .....	35	R/W Bit .....	291
SRCON0 (SR Latch Control 0) .....	122	Stack .....	52
SRCON1 (SR Latch Control 1) .....	123	Accessing .....	53
SSPADD (MSSP Address and Baud Rate, I <sup>2</sup> C Mode).....	280	Reset .....	53
SSPCON1 (MSSP Control 1).....	277	Stack Overflow/Underflow .....	64
SSPCON2 (SSP Control 2).....	278	STATUS Register .....	50
SSPCON3 (SSP Control 3).....	279	SUBWFB .....	349
SSPMSK (SSP Mask) .....	280	<b>T</b>	
SSPSTAT (SSP Status).....	276	T1CON Register .....	35, 169
STATUS .....	50	T1GCON Register .....	170
T1CON (Timer1 Control).....	169	T2CON Register .....	35, 43
T1GCON (Timer1 Gate Control).....	170	Thermal Considerations.....	368
TRISA (Tri-State PORTA) .....	86	Timer0.....	157
TRISB (Tri-State PORTB) .....	91	Associated Registers .....	159
TRISC (Tri-State PORTC) .....	94	Operation .....	157
TRISD (Tri-State PORTD) .....	97	Specifications .....	376
TRISE (Tri-State PORTE).....	101	Timer1.....	161
TxCON .....	175	Associated registers .....	171
TXSTA (Transmit Status and Control) .....	222	Asynchronous Counter Mode .....	163
WDTCON (Watchdog Timer Control).....	61	Reading and Writing .....	163
WPUB (Weak Pull-up PORTB) .....	90	Clock Source Selection.....	162
RESET .....	347	Interrupt .....	165
Reset .....	57	Operation .....	162
Reset Instruction .....	64	Operation During Sleep .....	165
Resets .....	57	Oscillator .....	163
Associated Registers .....	68	Prescaler .....	163
Revision History .....	403	Specifications .....	376
<b>S</b>		Timer1 Gate Selecting Source .....	163
SCK .....	281	TMR1H Register .....	161
SDI .....	281	TMR1L Register.....	161
SDO .....	281	Timer2	
Serial Clock, SCK.....	281	Associated registers .....	176
Serial Data In (SDI).....	281	Timer2/4/6.....	173
Serial Data Out (SDO) .....	281	Associated registers .....	176
Shoot-through Current .....	206	Timers	
Slave Select (SS).....	281	Timer1	
Software Simulator (MPLAB SIM).....	352	T1CON .....	169
SPBRG.....	225	T1GCON.....	170
SPBRG Register .....	37, 38	Timer2/4/6	
SPBRGH .....	225	TxCON.....	175
Special Event Trigger.....	135	Timing Diagrams	
Special Function Registers (SFRs) .....	35	A/D Conversion.....	378
SPI Mode (MSSP)		A/D Conversion (Sleep Mode).....	379
Associated Registers .....	286	Acknowledge Sequence .....	314
Serial Clock.....	281	Asynchronous Reception.....	220
Serial Data In .....	281	Asynchronous Transmission.....	216
Serial Data Out .....	281	Asynchronous Transmission (Back to Back) .....	216
Slave Select .....	281	Auto Wake-up Bit (WUE) During Normal Operation .....	232
SPI Clock .....	283	Auto Wake-up Bit (WUE) During Sleep .....	232
Typical Connection .....	282	Automatic Baud Rate Calibration.....	230
SR Latch .....	121	Baud Rate Generator with Clock Arbitration.....	307
SRCON0 Register.....	122	BRG Reset Due to SDA Arbitration During Start Condition .....	317
SRCON1 Register.....	123	Brown-out Reset (BOR).....	374
SS .....	281	Brown-out Reset Situations .....	62
SSPADD Register .....	39, 280	Bus Collision During a Repeated Start Condition (Case 1).....	318
SSPBUF Register .....	39	Bus Collision During a Repeated Start Condition (Case 2).....	318
SSPCON 1 Register.....	277	Bus Collision During a Start Condition (SCL = 0) .....	317
SSPCON Register.....	39	Bus Collision During a Stop Condition (Case 1) .....	319
SSPCON2 Register.....	278		
SSPCON3 Register.....	279		
SSPMSK Register.....	280		

# PIC16F193X/LF193X

Bus Collision During a Stop Condition (Case 2) .....	319	TMR0 Register.....	35
Bus Collision During Start Condition (SDA only) .....	316	TMR1H Register.....	35
Bus Collision for Transmit and Acknowledge.....	315	TMR1L Register.....	35
CLKOUT and I/O.....	372	TMR2 Register.....	35, 43
Clock Synchronization .....	304	TRIS .....	350
Clock Timing .....	370	TRISA Register.....	36, 86
Comparator Output .....	143	TRISB .....	89
Enhanced Capture/Compare/PWM (ECCP) .....	377	TRISB Register.....	36, 91
Fail-Safe Clock Monitor (FSCM) .....	120	TRISC .....	93
First Start Bit Timing .....	308	TRISC Register.....	36, 94
Full-Bridge PWM Output .....	200	TRISD .....	96
Half-Bridge PWM Output .....	198, 206	TRISD Register.....	36, 97
I <sup>2</sup> C Bus Data .....	385	TRISE .....	99
I <sup>2</sup> C Bus Start/Stop Bits.....	384	TRISE Register.....	36, 101
I <sup>2</sup> C Master Mode (7 or 10-Bit Transmission) .....	311	Two-Speed Clock Start-up Mode.....	117
I <sup>2</sup> C Master Mode (7-Bit Reception).....	313	TXCON (Timer2/4/6) Register .....	175
I <sup>2</sup> C Stop Condition Receive or Transmit Mode .....	314	TxCON Register .....	211
INT Pin Interrupt.....	71	TXREG .....	215
Internal Oscillator Switch Timing.....	116	TXREG Register.....	38
LCD Interrupt Timing in Quarter-Duty Cycle Drive....	268	TXSTA Register.....	38, 222
LCD Sleep Entry/Exit when SLPEN = 1 or CS = 00 .	270	BRGH Bit .....	225
PWM Auto-shutdown .....	205	<b>U</b>	
Firmware Restart .....	205	USART	
PWM Direction Change .....	201	Synchronous Master Mode	
PWM Direction Change at Near 100% Duty Cycle ...	202	Requirements, Synchronous Receive .....	381
PWM Output (Active-High).....	196	Requirements, Synchronous Transmission.....	381
PWM Output (Active-Low) .....	197	Timing Diagram, Synchronous Receive .....	381
Repeat Start Condition.....	309	Timing Diagram, Synchronous Transmission...	381
Reset, WDT, OST and Power-up Timer .....	373	<b>V</b>	
Send Break Character Sequence .....	233	VREF. <i>SEE</i> ADC Reference Voltage	
SPI Master Mode (CKE = 1, SMP = 1) .....	382	<b>W</b>	
SPI Mode (Master Mode).....	283	Wake-up on Break.....	231
SPI Slave Mode (CKE = 0) .....	383	Wake-up Using Interrupts.....	334
SPI Slave Mode (CKE = 1) .....	383	Watchdog Timer (WDT).....	59
Synchronous Reception (Master Mode, SREN) .....	237	Clock Source .....	59
Synchronous Transmission.....	235	Modes.....	60
Synchronous Transmission (Through TXEN) .....	235	Period .....	59
Time-out Sequence		Specifications .....	375
Case 1 .....	66	WCOL .....	307, 310, 312, 314
Case 2 .....	67	WCOL Status Flag.....	307, 310, 312, 314
Case 3 .....	67	WDTCON Register .....	61
Timer0 and Timer1 External Clock .....	376	WPUB Register.....	90
Timer1 Incrementing Edge.....	165	WWW Address .....	413
Two Speed Start-up .....	118	WWW, On-Line Support .....	11
Type-A in 1/2 Mux, 1/2 Bias Drive .....	257		
Type-A in 1/2 Mux, 1/3 Bias Drive .....	259		
Type-A in 1/3 Mux, 1/2 Bias Drive .....	261		
Type-A in 1/3 Mux, 1/3 Bias Drive .....	263		
Type-A in 1/4 Mux, 1/3 Bias Drive .....	265		
Type-A/Type-B in Static Drive.....	256		
Type-B in 1/2 Mux, 1/2 Bias Drive .....	258		
Type-B in 1/2 Mux, 1/3 Bias Drive .....	260		
Type-B in 1/3 Mux, 1/2 Bias Drive .....	262		
Type-B in 1/3 Mux, 1/3 Bias Drive .....	264		
Type-B in 1/4 Mux, 1/3 Bias Drive .....	266		
USART Synchronous Receive (Master/Slave) .....	381		
USART Synchronous Transmission (Master/Slave) .	381		
Wake-up from Interrupt .....	334		
Timing Diagrams and Specifications			
PLL Clock.....	371		
Timing Parameter Symbology.....	369		
Timing Requirements			
I <sup>2</sup> C Bus Data .....	386		
I <sup>2</sup> C Bus Start/Stop Bits .....	385		
SPI Mode .....	384		

# PIC16F193X/LF193X

---

NOTES:

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com), click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://support.microchip.com>**

# PIC16F193X/LF193X

---

---

## 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  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_

From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC16F193X/LF193X Literature Number: DS41364A

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?

---

---



# PIC16F193X/LF193X

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
<b>Device:</b>	PIC16F1933, PIC16LF1933, PIC16F1933T, PIC16LF1933T <sup>(1)</sup> PIC16F1934, PIC16LF1934, PIC16F1934T, PIC16LF1934T <sup>(1)</sup> PIC16F1936, PIC16LF1936, PIC16F1936T, PIC16LF1936T <sup>(1)</sup> PIC16F1937, PIC16LF1937, PIC16F1937T, PIC16LF1937T <sup>(1)</sup> PIC16F1938, PIC16LF1938, PIC16F1938T, PIC16LF1938T <sup>(1)</sup> PIC16F1939, PIC16LF1939, PIC16F1939T, PIC16LF1939T <sup>(1)</sup>		
<b>Temperature Range:</b>	I = -40°C to +85°C E = -40°C to +125°C		
<b>Package:</b>	ML = Micro Lead Frame (QFN) P = Plastic DIP PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP SS = SSOP		
<b>Pattern:</b>	3-Digit Pattern Code for QTP (blank otherwise)		

**Examples:**

- PIC16LF1937 - I/P = Industrial temp., Plastic DIP package, low-voltage V<sub>DD</sub> limits.
- PIC16F1934 - I/PT = Industrial temp., TQFP package, standard V<sub>DD</sub> limits.
- PIC16F1933 - E/ML = Extended temp., QFN package, standard V<sub>DD</sub> limits.

**Note 1:** F = Standard Voltage Range  
LF = Low Voltage Range

**2:** T = In tape and reel for QFN, TQFP, SOIC and SSOP packages only.



---

---

## 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:  
<http://support.microchip.com>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

**Atlanta**  
Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

**Boston**  
Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

**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

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**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: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**China - Nanjing**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**China - Qingdao**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**China - Shanghai**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**China - Shenyang**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

**China - Wuhan**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**China - Xiamen**  
Tel: 86-592-2388138  
Fax: 86-592-2388130

**China - Xian**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**China - Zhuhai**  
Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-4182-8400  
Fax: 91-80-4182-8422

**India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**India - Pune**  
Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

**Japan - Yokohama**  
Tel: 81-45-471- 6166  
Fax: 81-45-471-6122

**Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

**Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**Taiwan - Hsin Chu**  
Tel: 886-3-572-9526  
Fax: 886-3-572-6459

**Taiwan - Kaohsiung**  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Munich**  
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

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

01/02/08