TSC80251G1D

Extended 8–bit Microcontroller with Serial Communication Interfaces

Design Guide – October 1998

Design Guide Information

TEMIC reserves the right to make changes in the products or specifications contained in this document in order to improve design or performance and to supply the best possible products. TEMIC also assumes no responsibility for the use of any circuits described herein, conveys no license under any patents or other rights, and makes no representations that the circuits are free from patent infringement. Applications for any integrated circuits contained in this publication are for illustration purposes only and TEMIC makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification. Reproduction of any portion hereof without the prior written consent of TEMIC is prohibited.

On line information

World Wide Web: http://www.temic–semi.de

Factory Technical Support

Email Products: c251@temic.fr
Email Tools: x51_tools@temic.fr

Publisher

MHS S.A.
La Chantrerie – Route de Gachet,
BP 70602
44306 NANTES Cedex 03
France
Phone: 33 2 40 18 18 18
Fax: +33 2 40 18 19 60

Purchase of TEMIC I^{2}C components conveys a license under the Philips I^{2}C Patent Rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.

Section 1

Introduction

Table of Contents

1. Guide to this Manual

1.1. Introduction

This manual describes the TSC80251G1D embedded microcontroller, which is a member of the TEMIC TSC80251 microcontroller family. This manual is intended for use by both software and hardware designers familiar with the principles of microcontrollers.

1.2. Manual Contents

This manual contains 2 sections and 2 appendices, the paragraphs hereafter summarize the content of these sections and appendices.

1.2.1. “Introduction” Section

Chapter 1 (this chapter), “Guide to this manual”
Provides an overview of the manual. It summarizes the contents of the remaining chapters. The remainder of this chapter describes notational conventions and terminology used throughout the manual and provides references to related documentation.

Chapter 2, “Product Overview”
Summarizes the features of the TEMIC TSC80251G1D microcontroller devices.

1.2.2. “Design Information” Section

Chapter 1, “Address Spaces”
Describes the memory address space and the special function register (SFR) space of the TSC80251G1D microcontroller. It also provides a map of the SFR space showing the location of the SFRs and their reset values.

Chapter 2, “Device Configuration”
Describes microcontroller features that are configured at device reset, including the external memory interface (the number of external address bits, the number of wait states, memory regions for asserting RD#, WR#, and PSEN#, page mode), binary/source opcodes, interrupt mode, and the mapping of a portion of on–chip code memory to data memory. It describes the configuration bytes and how to program them for the desired configuration. It also describes how internal memory space maps into external memory.

Chapter 3, “External Memory Interface”
Describes the external memory signals and bus cycles. It provides waveform diagrams for the bus cycles, bus cycles with wait states, and the configuration byte bus cycles.

Chapter 4, “Input/Output Ports”
Describes the four 8–bit I/O ports (ports 0 to 3) and discusses their configuration for general–purpose I/O, external memory accesses (ports 0, 2), and alternative special functions.

Chapter 5, “Timer/Counters”
Describes the three on–chip timer/counters and discusses their application.

Chapter 6, “Serial I/O Port”
Describes the full–duplex serial I/O port and explains how to program it to communicate with external peripherals. This chapter also discusses baud rate generation (through timer 1, timer 2 and specific baud rate generator), framing error detection, multiprocessor communications, and automatic address recognition.

Chapter 7, “Event and Waveform Controller”
Describes the PCA on–chip peripheral and explains how to configure it for general–purpose applications (timers and counters) and special applications (programmable WDT and pulse–width modulator).

Chapter 8, "SSLC/Inter–Integrated Circuit Interface (I^2C)"
Describes the synchronous serial link controller when configured in I^2C mode.

Chapter 9, "SSLC/Synchronous Peripheral Interface (µWire/SPI)"
Describes the synchronous serial link controller when configured in µWire/SPI mode.

Chapter 10, "WatchDog Timer"
Describes the hardware watchdog timer (WDT). This chapter also provides instructions for using the WDT and describes the operation of the WDT during the idle and power–down modes.

Chapter 11, "Power Monitoring and Management"
Describes the TSC80251G1D power monitoring and management circuitry which provides a power–on reset, a power–fail reset, a power off flag, a clock prescaler, an idle mode, and a power–down mode.

Chapter 12, "Interrupt System"
Describes the TSC80251G1D interrupt circuitry which provides a TRAP instruction interrupt, a non maskable external interrupt, nine maskable interrupts: two external interrupts, three timer interrupts, an EWC–PCA interrupt, a serial port interrupt, a synchronous serial interface interrupt, and a keyboard interrupt. This chapter also discusses the interrupt priority scheme, the external interrupt inputs, the NMI input, and the keyboard interface.

1.2.3. Appendices

Appendix A, "Signal Descriptions"
Provides device pinouts and describes the function(s) of each pin. Descriptions are listed alphabetically by signal name.

Appendix B, "Registers"
Accumulates, for convenient reference, copies of the register definition figures that appear throughout the manual.

1.3. Notational Conventions and Terminology

The following notations and terminology are used in this manual.

The pound symbol (#) appended to a signal name means that the signal is active low.

XXXX Uppercase X (no italics) represents an unknown value or a "don't care" state or condition. The value may be either binary or hexadecimal, depending on the context. For example, 2XAFh (hex) indicates that bits 11:8 are unknown; 10XX in binary context indicates that the two LSBs are unknown.

Assert and Deassert The terms assert and deassert refer to the act of making a signal active (enabled) and inactive (disabled), respectively. The active polarity (high/low) is defined by the signal name. Active–low signals are designated by a pound symbol (#) suffix; active–high signals have no suffix. To assert RD# is to drive it low; to assert ALE is to drive it high; to deassert RD# is to drive it high; to deassert ALE is to drive it low.

Logic 0 (Low) An input voltage level equal to or less than the maximum value of V_{IL} or an output voltage level equal to or less than the maximum value of V_{OL}. See datasheet for values.

Logic 1 (High) An input voltage level equal to or greater than the minimum value of V_{IH} or an output voltage level equal to or greater than the minimum value of V_{OH}. See datasheet for values.

Numbers Hexadecimal numbers are represented by a string of hexadecimal digits followed by the character h. Binary numbers are represented by a string of binary digits followed by the character b. Decimal numbers are represented by their customary notations. That is, 255 is a decimal number, FFh is an hexadecimal number and 1111 1111b is a binary number.

Register Bits Bit locations are indexed by 7:0 for byte registers where bit 0 is the least–significant bit and 7 most significant bit. An individual bit is represented by the register name, followed by a period and the bit number. For example, PCON.4 is bit 4 of the power control register. In some discussions, bit names are used. For example, the name of PCON.4 is POF, the power–off flag.

Register Names Register names are shown in upper case. For example, PCON is the power control register. If a register name contains a lowercase character, it represents more than one register. For example, CCAPMx represents the five registers: CCAPM0 through CCAPM4.

Reserved Bits Some registers contain reserved bits. These bits are not used in this device, but they may be used in future implementations. In most cases value read from a reserved bit is indeterminate. Do not write a “1” to a reserved bit after reading it to “0”.

Set and Clear The terms set and clear refer to the value of a bit or the act of giving it a value. If a bit is set, its value is “1”; setting a bit gives it a “1” value. If a bit is clear, its value is ”0”; clearing a bit gives it a “0” value.

Signal Names Signal names are shown in upper case. When several signals share a common name, an individual signal is represented by the signal name followed by a number. Port pins are represented by the port abbreviation, a period, and the pin number (e.g., P0.0, P0.1). A pound symbol (#) appended to a signal name identifies an active–low signal.

Units of Measure The following abbreviations are used to represent units of measure:

Kbit	kilobits
Kbyte	kilobytes
Mbyte	megabytes
MHz	megahertz
ms	milliseconds
ns	nanoseconds

1.4. Related Documents and tools

1.4.1. Datasheet

The “TSC80251G1D Datasheet” contains quick reference to the product, it also includes all the DC and AC parameters, the packages information, and the ordering information.

1.4.2. Programmer’s Guide

The “TSC80251 Extended 8–bit Microcontrollers Programmer‘s Guide” contains all information for the programmer (architecture, Memory Mapping and instruction set).

1.4.3. Starter Kit

TEMIC proposes a starter kit for the TSC80251G1D to be evaluated by the designer.
The starter kit content is:

- C–Compiler (limited to 2 Kbytes of code)
- Assembler
- Linker
- TSC80251G1 Simulator
- TSC80C251G1 Evaluation Board with ROM–Monitor

1.4.4. Development Tools

Up to date information concerning development tools for TEMIC TSC80251 devices is available on the TEMIC web site.

1.4.5. World Wide Web

TEMIC offers a variety of technical information through the Word Wide Web:

http://www.temic–semi.de

1.4.6. Application support

TEMIC offers through E–mail a technical hotline dedicated to the TSC80251 microcontroller family: C251@temic.fr.

TEMIC also offers through E–mail a technical hotline dedicated to the development tools: x51_tools@temic.fr.

2. Product Overview

The TSC80251G1D products are derivatives of the TEMIC Microcontroller family based on the extended 8–bit C251 Architecture. This family of products is tailored to 8–bit microcontroller applications requiring an increased instruction throughput, a reduced operating frequency or a larger addressable memory space. The architecture can provide a significant code size reduction when compiling C programs while fully preserving the legacy of C51 assembly routines.

The TSC80251G1D derivatives are pin–out and software compatible with standard 80C51/Fx/Rx with extended on–chip data memory (1 Kbyte RAM) and up to 256 Kbytes of external code and data. Additionally, the TSC83251G1D provides on–chip code memory (16 Kbytes ROM).

They provide transparent enhancements to Intel's 8xC251Sx family with an additional Synchronous Serial Link Controller (SSLC supporting I^{2}C, µWire and SPI protocols), a Keyboard interrupt interface and Power Monitoring and Management features.

More information on the TSC80251 architecture is provided in the "TSC80251 Extended 8–bit Microcontroller Programmer's Guide".

Pinouts and signals description are provided in Appendix A, "Signals Description".

2.1. Typical Applications

- ISDN terminals
- High–Speed modems
- PABX (SOHO)
- Networking
- Line cards
- Computer peripherals
- Printers
- Plotters
- Scanners
- Banking machines
- Barcode readers
- Smart cards readers
- High–end digital monitors
- High–end joysticks

2.2. Features

- Pin–Out and software compatibility with standard 80C51 products and 80C51FA/FB/RA/RB
- Plug–in replacement of Intel's 80C251Sx
- C251 core: Intel's MCS®251 D–step compliant
    - 40–byte Register File
    - Registers Accessible as Bytes, Words or Dwords
    - Single–state Pipeline with 16–bit Internal Code Fetch
- Enriched C51 Instruction Set
    - 16–bit and 32–bit ALU
    - Compare and Conditional Jump Instructions
    - Expanded Set of Move Instructions
- Linear Addressing
- 1 Kbyte of on–chip RAM
- External memory space (Code/Data) programmable from 64 Kbytes to 256 Kbytes
- TSC83251G1D: 16 Kbytes of on–chip masked ROM
  (Engineering and fast production with TSC87251G1A OTP/EPROM version)
- TSC80251G1D: ROMless version
- Four 8–bit parallel I/O Ports (Ports 0, 1, 2 and 3 of the standard 80C51)
- Serial I/O Port: full duplex UART (80C51 compatible) with independent Baud Rate Generator
- SSLC: Synchronous Serial Link Controller
    - I^{2}C multi–master protocol
    - µWire and SPI master and slave protocols
- Three 16–bit Timers/Counters (Timers 0, 1 and 2 of the standard 80C51)
- EWC: Event and Waveform Controller
    - Compatible with Intel's Programmable Counter Array (PCA)
    - Common 16–bit Timer/Counter reference with four possible clock sources (Fosc/4, Fosc/12, Timer 1 and external input)
    - Five modules, each with four programmable modes:
        - 16–bit software Timer/Counter
        - 16–bit Timer/Counter Capture Input and software pulse measurement
        - High–speed output and 16–bit software Pulse Width Modulation (PWM)
        - 8–bit hardware PWM without overhead
    - 16–bit Watchdog Timer/Counter capability
- Secure 14–bit Hardware Watchdog Timer
- Power Monitoring and Management
    - Power–Fail reset
    - Power–On reset (integrated on the chip)
    - Power–Off flag (cold and warm resets)
    - Software programmable system clock
    - Idle and Power–Down modes
- Keyboard interrupt interface on Port 1
- Non Maskable Interrupt input (NMI)
- Real–time Wait states inputs (WAIT#/AWAIT#)
- ONCE mode and full speed Real–Time In–Circuit Emulation support (Third Party Vendors)

2.3. Block Diagram

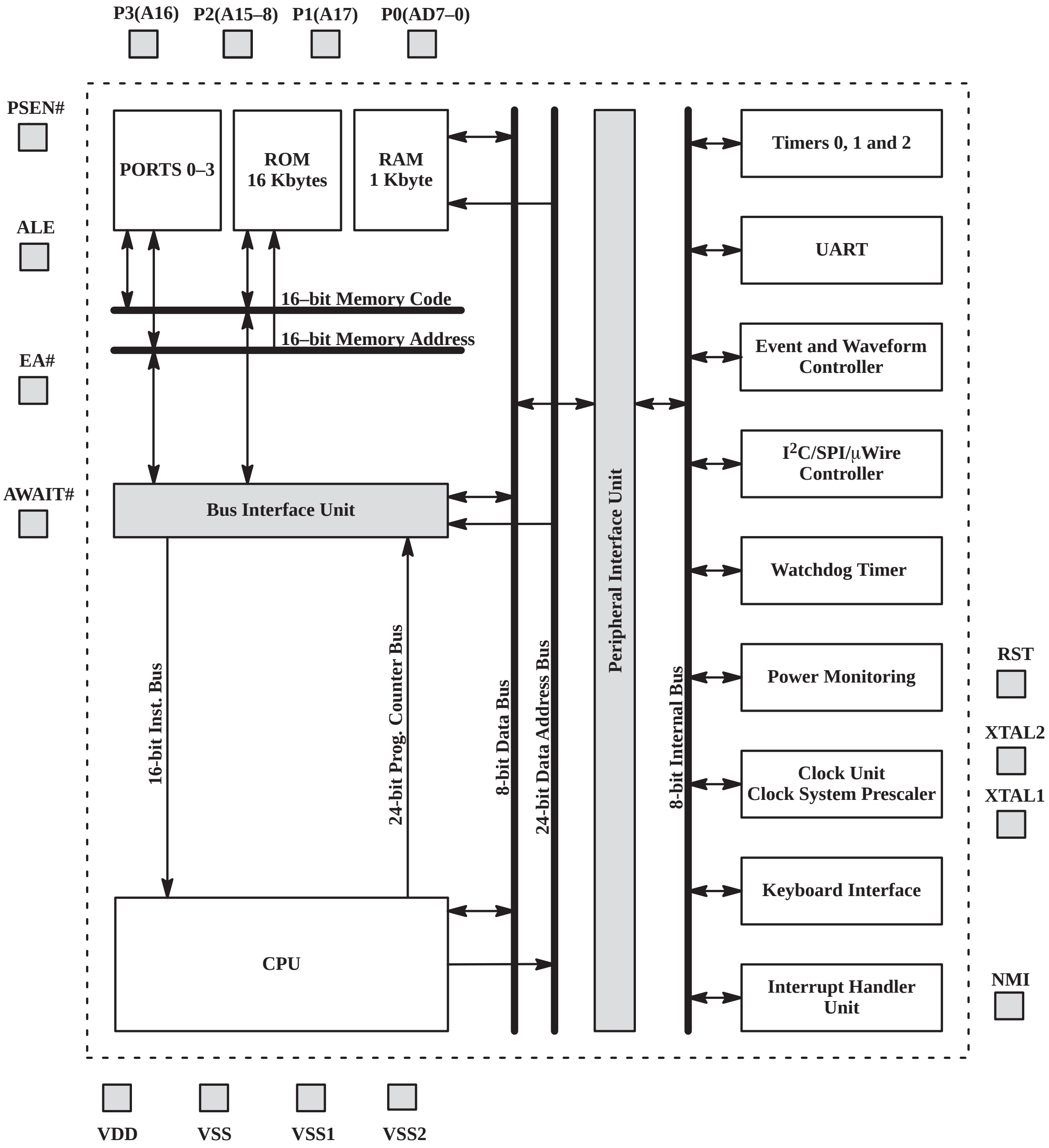


Figure 2.1 TSC80251G1D Block Diagram

Section 2

Design Information

Table of Contents

1. Address Spaces

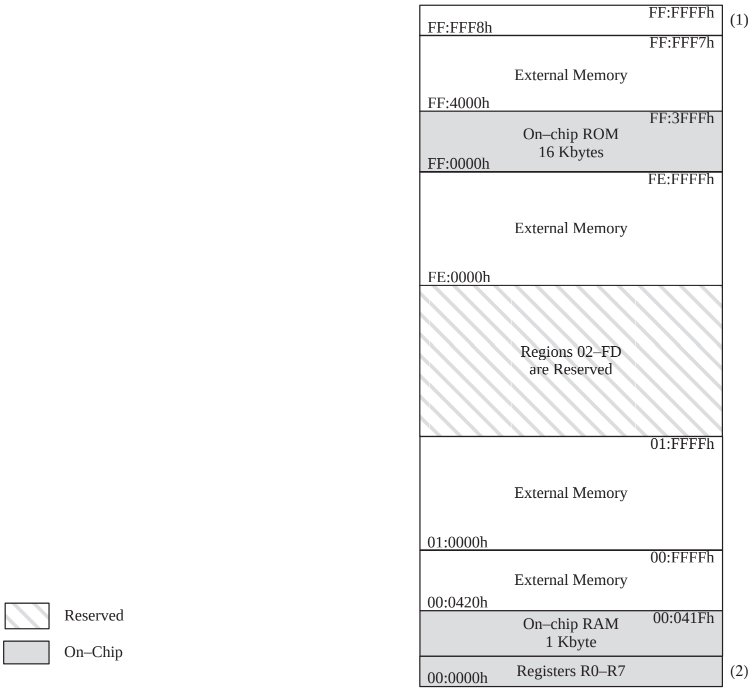
1.1. Introduction

The TSC80251G1D derivatives have three address spaces: a memory space, a special function register (SFR) space and register file. This chapter describes the memory and SFR address spaces as they apply to TSC80251G1D. Register file information and address spaces comparison of the C251 Architecture versus the C51 Architecture are described in the TSC80251 Programmer's Guide.

1.2. TSC80251G1D Memory Space

The usable memory of the TSC80251G1D consists of four 64 Kbytes regions: 00:, 01:, FE:, and FF:,. Code can execute from all four regions; code execution begins at FF:0000h. Regions 02:–FD: are reserved. Reading a location in the reserved area returns an unspecified value. Software must avoid writing to the reserved area to not spuriously write to the other existing regions.

All four regions of the memory space are available at the same time. The maximum number of external address lines is 18, which limits external memory to a maximum of four regions (256 Kbytes).



Notes:
1. *Eight–byte configuration array (FF:FFF8h – FF:FFFFh).*
2. *Four banks of registers R0–R7 (32 BYTES, 00:0000h – 00:001Fh).*

Figure 1.1 Memory Mapping in the TSC80251G1D Address Space

Figure 1.1 shows how on–chip RAM, on–chip ROM, and external memory are mapped in the TSC80251G1D address space. The first 32 bytes of on–chip RAM store banks 0–3 of the register file.

1.2.1. On–chip General–purpose Data RAM

On–chip RAM (1 Kbyte) is provided for general data storage (Figure 1.1). Instructions cannot execute from on–chip data RAM. The data is accessible by direct, indirect and displacement addressing. Locations 00:0020h–00:007Fh are also bit addressable.

1.2.2. On–chip Code Memory

The TSC80251G1D derivatives are available with 16 Kbytes of on–chip ROM (TSC83251G1D) or as well as without on–chip code memory (TSC80251G1D) The on–chip ROM is intended primarily for code storage, although its contents can also be read as data with the indirect and displacement addressing modes. Following a chip reset, program execution begins at FF:0000h.

A code fetch within the address range of the on–chip ROM accesses the on–chip ROM only if EA#= 1. For EA#= 0, a code fetch in this address range accesses external memory. The value of EA# is latched when the chip leaves the reset state. Code is fetched faster from on–chip code memory than from external memory. Table 1.1 lists the minimum times to fetch two bytes of code from on–chip memory and external memory.

Table 1.1 Minimum Times to Fetch Two Bytes of Code

Type of Code Memory	State Times
On–chip Code Memory	1
External Memory (page mode)	2
External Memory (non–page mode)	4

Note:
If your program executes exclusively from on–chip ROM (not from external memory), beware of executing code from the upper eight bytes of the on–chip ROM (FF:3FF8h–FF:3FFFh). Because of its pipeline capability, the TSC80251G1D may attempt to prefetch code from external memory (at an address above FF:3FFFh) and thereby disrupt I/O ports 0 and 2. Fetching code constants from these eight bytes does not affect ports 0 and 2.
If your program executes from both on–chip ROM and external memory, your code can be placed in the upper eight bytes of the on–chip ROM. As the TSC80251G1D fetches bytes above the top address in the on–chip ROM, the code fetches automatically become external bus cycles. In other words, the rollover from on–chip ROM to external code memory is transparent to the user.

1.2.2.1. Accessing On–chip Code Memory in Region 00:

The TSC83251G1D can be configured so that the upper half of the 16 Kbytes on–chip code memory can also be read as near data at locations in the top of region 00:. That is, locations FF:2000h–FF:3FFFh can also be accessed at locations 00:E000h–00:FFFFh. This is useful for accessing code constants stored in ROM and leads to faster code execution to retrieve these constants. Note, however, that all of the following three conditions must hold for this mapping to be effective:

- The device is configured with EMAP#= 0 in the UCONFIG1 register
- EA#= 1
- The access to this area of region 00: is a data read, not a code fetch.

If one or more of these conditions do not hold, accesses to the locations in region 00: are referred to external memory.

1.2.3. External Memory

Regions 01:, FE:, and portions of regions 00: and FF: of the memory space are implemented as external memory. For discussions of external memory see paragraph 2.4. “Configuring the External Memory Interface” and chapter 3. “External Memory Interface”.

1.3. Special Function Registers

SFRs are placed in a reserved on–chip memory region S: which is not represented in the address space mapping (Figure 1.1). The relative addresses within S: of these SFRs are provided together with their reset values in Table 1.11. They are upward compatible with the SFRs of the standard 80C51 and the Intel’s 80C251Sx family. In this table, the C251 core registers are in italics and are described in the TSC80251 Programmer’s Guide. All the SFRs are accessible by direct and bit addressing.

The Special Function Registers (SFRs) of the TSC80251G1D derivatives fall into the categories detailed in Table 1.2 to Table 1.10.

Table 1.2 C251 Core SFRs

Mnemonic	Name	Mnemonic	Name
ACC[1]	Accumulator	SPH[1]	Stack Pointer High – MSB of SPX
B[1]	B Register	DPL[1]	Data Pointer Low byte – LSB of DPTR
PSW	Program Status Word	DPH[1]	Data Pointer High byte – MSB of DPTR
PSW1	Program Status Word 1	DPXL[1]	Data Pointer Extended Low byte of DPX – Region number
SP[1]	Stack Pointer – LSB of SPX		

Note:
1. *These SFRs can also be accessed by their corresponding registers in the register file.*

Table 1.3 I/O Port SFRs

Mnemonic	Name	Mnemonic	Name
P 0	Port 0	P 2	Port 2
P 1	Port 1	P 3	Port 3

Table 1.4 Timers SFRs

Mnemonic	Name	Mnemonic	Name
TL0	Timer/Counter 0 Low Byte	TMOD	Timer/Counter 0 and 1 Modes
TH0	Timer/Counter 0 High Byte	T2CON	Timer/Counter 2 Control
TL1	Timer/Counter 1 Low Byte	T2MOD	Timer/Counter 2 Mode
TH1	Timer/Counter 1 High Byte	RCAP2L	Timer/Counter 2 Reload/Capture Low Byte
TL2	Timer/Counter 2 Low Byte	RCAP2H	Timer/Counter 2 Reload/Capture High Byte
TH2	Timer/Counter 2 High Byte	WDTRST	WatchDog Timer Reset
TCON	Timer/Counter 0 and 1 Control		

Table 1.5 Serial I/O Port SFRs

Mnemonic	Name	Mnemonic	Name
SCON	Serial Control	SADDR	Slave Address
SBUF	Serial Data Buffer	BRL	Baud Rate Reload
SADEN	Slave Address Mask	BDRCON	Baud Rate Control

Table 1.6 SSLC SFRs

Mnemonic	Name	Mnemonic	Name
SSCON	Synchronous Serial control	SSADR	Synchronous Serial Address
SSDAT	Synchronous Serial Data	SSBR	Synchronous Serial Bit Rate
SSCS	Synchronous Serial Control and Status		

Table 1.7 Event Waveform Control SFRs

Mnemonic	Name	Mnemonic	Name
CCON	EWC–PCA Timer/Counter Control	CCAP1L	EWC–PCA Compare Capture Module 1 Low Register
CMOD	EWC–PCA Timer/Counter Mode	CCAP2L	EWC–PCA Compare Capture Module 2 Low Register
CL	EWC–PCA Timer/Counter Low Register	CCAP3L	EWC–PCA Compare Capture Module 3 Low Register
CH	EWC–PCA Timer/Counter High Register	CCAP4L	EWC–PCA Compare Capture Module 4 Low Register
CCAPM0	EWC–PCA Timer/Counter Mode 0	CCAP0H	EWC–PCA Compare Capture Module 0 High Register
CCAPM1	EWC–PCA Timer/Counter Mode 1	CCAP1H	EWC–PCA Compare Capture Module 1 High Register
CCAPM2	EWC–PCA Timer/Counter Mode 2	CCAP2H	EWC–PCA Compare Capture Module 2 High Register
CCAPM3	EWC–PCA Timer/Counter Mode 3	CCAP3H	EWC–PCA Compare Capture Module 3 High Register
CCAPM4	EWC–PCA Timer/Counter Mode 4	CCAP4H	EWC–PCA Compare Capture Module 4 High Register
CCAP0L	EWC–PCA Compare Capture Module 0 Low Register		

Table 1.8 System Management SFRs

Mnemonic	Name	Mnemonic	Name
PCON	Power Control	CKRL	Clock Reload
POWM	Power Management	WCON	Synchronous Real–Time Wait State Control
PFILT	Power Filter		

Table 1.9 Interrupt SFRs

Mnemonic	Name	Mnemonic	Name
IE0	Interrupt Enable Control 0	IPL0	Interrupt Priority Control Low 0
IE1	Interrupt Priority Control 1	IPH1	Interrupt Priority Control High 1
IPH0	Interrupt Priority Control High 0	IPL1	Interrupt Priority Control Low 1

Table 1.10 Keyboard Interface SFRs

Mnemonic	Name	Mnemonic	Name
P1IE	Port 1 Interrupt Input Enable	P1LS	Port 1 Level Selection
P1F	Port 1 Flag		

Table 1.11 SFR Addresses and Reset Values

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h		CH 0000 0000	CCAP0H 0000 0000	CCAP1H 0000 0000	CCAP2H 0000 0000	CCAP3H 0000 0000	CCAP4H 0000 0000		FFh
F0h	B[1] 0000 0000								F7h
E8h		CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000	CCAP2L 0000 0000	CCAP3L 0000 0000	CCAP4L 0000 0000		EFh
E0h	ACC[1] 0000 0000								E7h
D8h	CCON 00X0 0000	CMOD 00XX X000	CCAPM0 X000 0000	CCAPM1 X000 0000	CCAPM2 X000 0000	CCAPM3 X000 0000	CCAPM4 X000 0000		DFh
D0h	PSW[1] 0000 0000	PSW1[1] 0000 0000							D7h
C8h	T2CON 0000 0000	T2MOD XXXX XX00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000			CFh
C0h									C7h
B8h	IPL0 X000 0000	SADEN 0000 0000					SPH[1] 0000 0000		BFh
B0h	P3 1111 1111	IE1 XX0X XXX0	IPL1 XX0X XXX0	IPH1 XX0X XXX0				IPH0 X000 0000	B7h
A8h	IE0 0000 0000	SADDR 0000 0000							AFh
A0h	P2 1111 1111						WDTRST 1111 1111	WCON XXXX XX00	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	P1LS 0000 0000	P1IE 0000 0000	P1F 0000 0000		9Fh
90h	P1 1111 1111		SSBR 0000 0000	SSCON (2)	SSCS (3)	SSDAT 0000 0000	SSADR 0000 0000		97h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	CKRL 0000 1000	POWM 0XXX 0XXX	8Fh
80h	P0 1111 1111	SP 0000 0111	DPL[1] 0000 0000	DPH[1] 0000 0000	DPXL[1] 0000 0001		PFILT XXXX XXXX	PCON 0000 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

□ reserved

Notes:

1. *These registers are described in the TSC80251 Programmer's Guide (C251 core registers).*
2. *In I^{2}C and SPI modes, SSCON is splitted in two separate registers. SSCON reset value is 0000 0000 in I^{2}C mode and 0000 0100 in SPI mode.*
2. *In read and write modes, SSCS is splitted in two separate registers. SSCS reset value is 1111 1000 in read mode and 0000 0000 in write mode.*

2. Device Configuration

The TSC80251G1D derivatives provide user design flexibility by configuring certain operating features at device reset. These features fall into the following categories:

- external memory interface (page mode, address bits, pre–programmed wait states and the address range for RD#, WR#, and PSEN#)
- source mode/binary mode opcodes
- selection of bytes stored on the stack by an interrupt
- mapping of the upper portion of on–chip code memory to region 00:

You can specify a 16–bit, 17–bit, or 18–bit external address bus (64 to 256 Kbytes external address space). Wait state configurations provide pre–programmed 0, 1, 2, or 3 wait states.

This chapter provides a detailed discussion of the TSC80251G1D device configuration. It describes the configuration bytes and provides information to help user in selecting a suitable configuration for his application. It discusses the choices involved in configuring the external memory interface and shows how the internal memory maps into the external memory.

2.1. Configuration Overview

The configuration of the microcontroller is established after the reset sequence based on information stored in configuration bytes. The TSC80251G1D derivatives store configuration information in two configuration bytes.

2.2. Device Configuration

The TSC80251G1D derivatives reserve the top eight bytes of the memory address map (FF:FFF8h-FF:FFFFh) for an eight–byte configuration array. The two lowest bytes of the configuration array are assigned to the user configuration bytes UCONFIG0 (FF:FFF8H) and UCONFIG1 (FF:FFF9H). Bit definitions of UCONFIG0 and UCONFIG1 are provided in Figure 2.7 and Figure 2.8. The upper 6 bytes of the configuration array are reserved for future use.

2.2.1. ROMless Devices

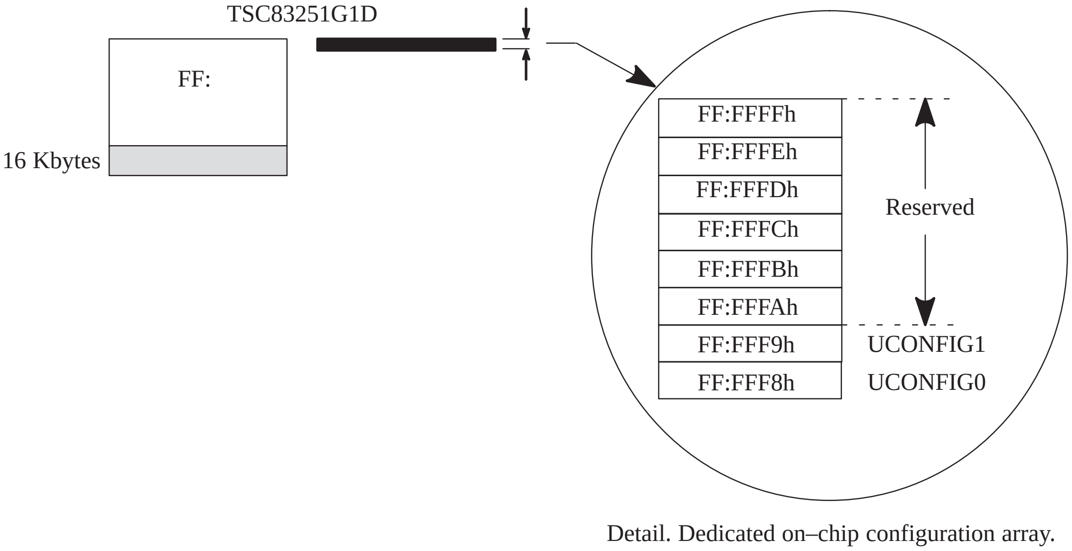
For devices without on–chip code memory (TSC80251G1D), configuration information is fetched from external memory system using internal addresses FF:FFF8h and FF:FFF9h. User configuration bytes UCONFIG0 and UCONFIG1 should be stored in an eight–byte configuration array located at the highest addresses implemented in external code memory (see Table 2.1 and Figure 2.2). Paragraph 3.6. “Configuration Byte Bus Cycle” discusses on how the configuration bytes are retrieved from external memory.

2.2.2. ROM Devices

For devices with on–chip code memory (TSC83251G1D), configuration information is fetched from a dedicated on–chip non–volatile memory at addresses FF:FFF8h and FF:FFF9h (see Figure 2.1). User configuration bytes UCONFIG0 and UCONFIG1 are programmed at the factory using customer’s configuration data supplied with the code file.

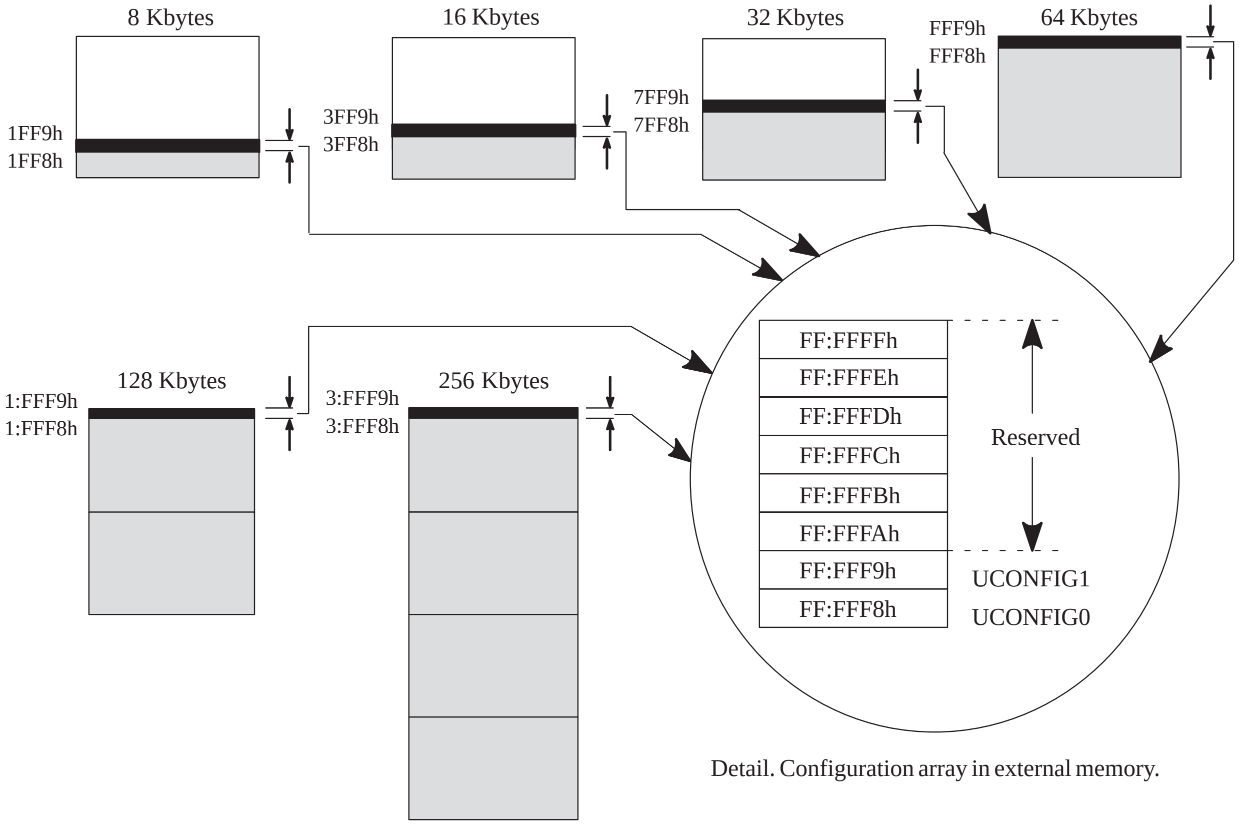
Note:
ROM devices used with EA# pin = 0 operate as ROMless devices.

Caution:
The eight highest addresses in the memory address space (FF:FFF8h-FF:FFFFh) are reserved for the configuration array. Do not read or write these locations. These addresses are also used to access the configuration array in external memory, so the same restrictions apply to the eight highest addresses implemented in external memory. Instructions that might inadvertently cause these addresses to be accessed due to call returns or prefetches should not be located at addresses immediately below the configuration array. Use an EJMP instruction, five or more addresses below the configuration array, to continue execution in other areas of memory.



Note:
For EA#= 1, the TSC83251G1D obtains configuration information from a dedicated on–chip non–volatile memory at addresses FF:FFF8h and FF:FFF9h.

Figure 2.1 Configuration Array (On–chip)



Note:
For EA#= 0, the TSC80251G1D derivatives obtain configuration information from configuration bytes in external memory using internal addresses FF:FFF8h and FF:FFF9h. In external memory, the eight–byte configuration array is located at the highest addresses implemented.

Figure 2.2 Configuration Array (External)

Table 2.1 External Addresses for Configuration Array

Size of External Address Bus (Bits)	Address of Configuration Array on External Bus (2)	Address of Configuration Bytes on External Bus (1)
16	FFF8h–FFFFh	UCONFIG1: FFF9h UCONFIG0: FFF8h
17	1FFF8h–1FFFFh	UCONFIG1: 1FFF9h UCONFIG0: 1FFF8h
18	3FFF8h–3FFFFh	UCONFIG1: 3FFF9h UCONFIG0: 3FFF8h

Notes:

1. *When EA#= 0, the reset routine retrieves UCONFIG0 and UCONFIG1 from external memory using internal addresses FF:FFF8h and FF:FFF9h, which appear on the microcontroller external address bus (A17, A16, A15:0).*
2. *The upper six bytes of the configuration array are reserved for future use.*

2.3. The Configuration Bits

This paragraph provides a brief description of the configuration bits contained in the configuration bytes (Figure 2.7 and Figure 2.8). UCONFIG0 and UCONFIG1 have five wait state bits: WSA1:0#, WSB1:0#, and WSB.

- SRC. Selects source mode or binary mode opcode configuration.
- INTR. Selects the bytes pushed onto the stack by interrupts.
- EMAP#. Maps on–chip code memory (16–Kbyte devices only) to memory region 00:.

The following bits configure the external memory interface.

- PAGE#. Selects page/non–page mode and specifies the data port.
- RD1:0. Selects the number of external address bus pins and the address range for RD#, WR,and PSEN#.
- XALE#. Extends the ALE pulse.
- WSA1:0#. Selects 0, 1, 2, or 3 pre–programmed wait states for all regions except 01:.
- WSB1:0#. Selects 0 – 3 pre–programmed wait states for memory region 01:.
- EMAP#. Affects the external memory interface in that, when asserted, addresses in therange 00:E000H-00:FFFH access on–chip memory.

2.4. Configuring the External Memory Interface

This paragraph describes the configuration options that affect the external memory interface. The configuration bits described here determine the following interface features:

- page mode or non–page mode (PAGE#)
- the number of external address pins (16, 17, or 18) (RD1:0)
- the memory regions assigned to the read signals RD# and PSEN# (RD1:0)
- the external wait states (WSA1:0#, WSB1:0#, XALE#)
- mapping a portion of on–chip code memory to data memory (EMAP#)

2.4.1. Page Mode and Non–Page Mode (PAGE#)

The PAGE# bit (UCONFIG0.1) determines whether code fetches use page mode or non–page mode and whether data is transmitted on P2 or P0. See paragraph 3.2.3. “Page Mode Bus Cycles”, for a description of the bus structure and page mode operation.

- Non–Page Mode: PAGE#= 1. The bus structure is the same as for the MCS 51 architecture with data D7:0 multiplexed with A7:0 on P0. External code fetches require two state times (4TOSC).
- Page Mode: PAGE#= 0. The bus structure differs from the bus structure in MCS 51 controllers. Data D7:0 is multiplexed with A15:8 on P2. Under certain conditions, external code fetches require only one state time (2TOSC).

Caution:

When TSC83251G1D is not used in romless mode (EA#= 1), then Port 2 is used as I/O port. In this configuration, PAGE# bit must be set to logic 1 to select non–page mode (see paragraph 3.7. “Port 0 and Port 2 Status”.

2.4.2. Configuration Bits RD1:0

The RD1:0 configuration bits (UCONFIG0.3:2) determine the number of external address signals and the address ranges for asserting the read signals PSEN#/RD# and the write signal WR#. These selections offer different ways of addressing external memory. Figure 2.3 to Figure 2.6 show how internal memory maps into external memory for the four values of RD1:0.

A key to the memory interface is the relationship between internal memory a dresses and external memory addresses. While the TSC80251G1D has 24 internal address bits, the number of external address lines is less than 24 (i.e., 16, 17, or 18 depending on the values of RD1:0). This means that reads/writes to different internal memory addresses can access the same location in external memory.

For example, if the TSC80251G1D is configured for 17 external address lines, a write to location 01:6000h and a write to location FF:6000h accesses the same 17–bit external address (1:6000h) because A16= 1 for both internal addresses. In other words, regions 01: and FF: map into the same 64 Kbytes region in external memory.

In some situations, however, a multiple mapping from internal memory to external memory does not preclude using more than one region. For example, for a device with on–chip ROM configured for 17 address bits and with EA#= 1, an access to FF:0000h-FF:3FFFh (16 Kbytes) accesses the on–chip ROM, while an access to 01:0000h-01:3FFFh is to external memory. In this case, you could execute code from these lo–cations in region FF: and store data in the corresponding locations in region 01: without conflict.

2.4.2.1. RD1:0= 00 (18 External Address Bits)

The selection RD1:0= 00 provides 18 external address bits: A15:0 (ports P0 and P2), A16 (from P3.7/RD#/A16), and A17 (from P1.7/CEX4/A17/WCLK). Bits A16 and A17 can select four 64–Kbyte regions of external memory for a total of 256 Kbytes (Figure 2.3). This is the largest possible external memory space.

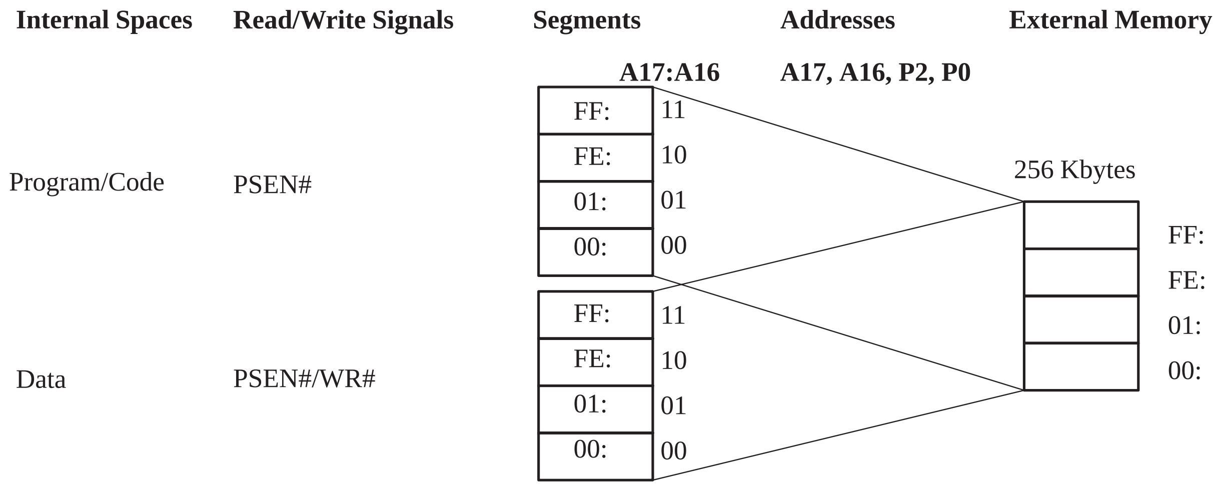


Figure 2.3 Internal/External Memory Segments (RD1:0= 00)

2.4.2.2. RD1:0= 01 (17 External Address Bits)

The selection RD1:0= 01 provides 17 external address bits: A15:0 (ports P0 and P2) and A16 (from P3.7/RD#/A16). Bit A16 can select two 64–Kbyte regions of external memory for a total of 128 Kbytes (Figure 2.4). Regions 00: and FE: (each having A16= 0) map into the same 64–Kbyte region in external memory. This duplication also occurs for regions 01: and FF:

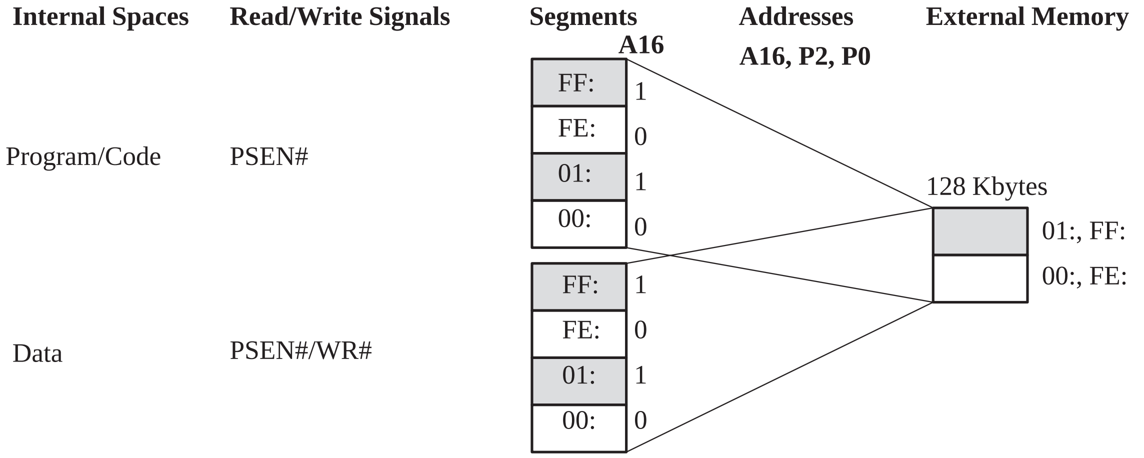


Figure 2.4 Internal/External Memory Segments (RD1:0= 01)

2.4.2.3. RD1:0= 10 (16 External Address Bits)

For RD1:0= 10, the 16 external address bits (A15:0 on ports P0 and P2) provide a single 64–Kbyte region in external memory (Figure 2.5). This selection provides the smallest external memory space; however, pin P3.7/RD#/A16 is available for general I/O and pin P1.7 is available for general I/O, PCA I/O, SSLC I/O, and synchronous real–time wait clock output. This selection is useful when the availability of these pins is required and/or a small amount of external memory is sufficient.

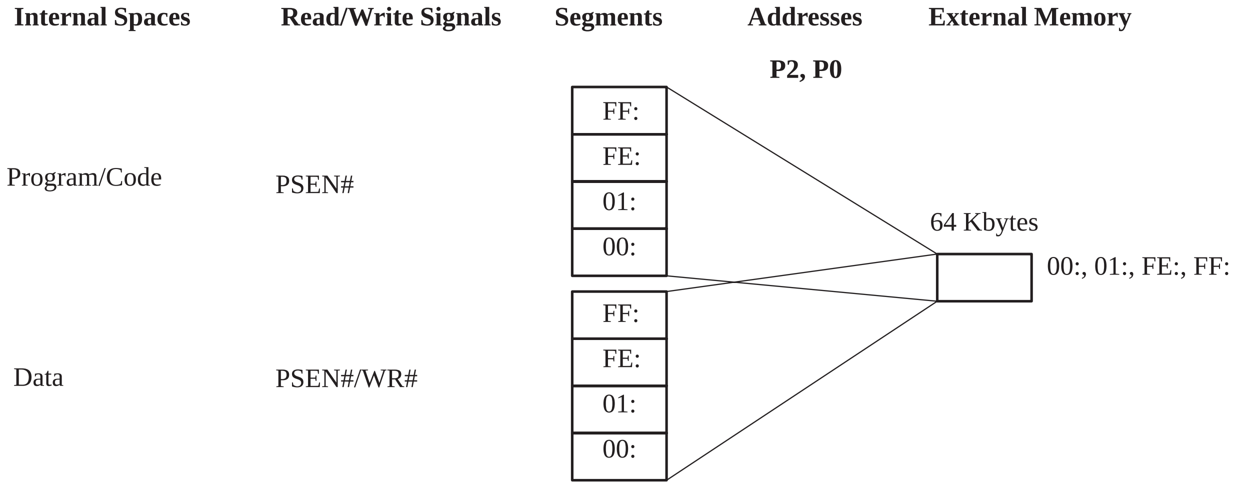


Figure 2.5 Internal/External Memory Segments (RD1:0= 10)

2.4.2.4. RD1:0= 11 (Compatible with MCS® 51 Microcontrollers)

The selection RD1:0= 11 provides only 16 external address bits (A15:0 on ports P0 and P2). However, PSEN# is the read signal for regions FE:-FF:, while RD# is the read signal for regions 00:-01: (Figure 2.6). The two read signals effectively expand the external memory space to two 64–Kbyte regions. WR# is asserted only for writes to regions 00:-01:. This selection provides compatibility with MCS 51 microcontrollers, which have separate external memory spaces for code and data.

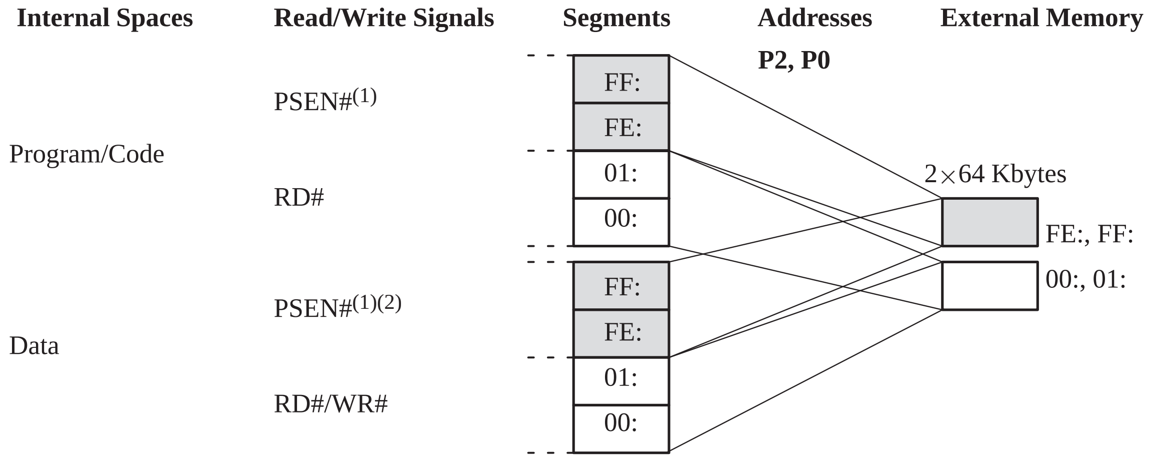


Figure 2.6 Internal/External Memory Segments (RD1:0= 11)

Notes:

1. *PSEN# is asserted instead of RD# when reading data in regions FE: and FF:.*
2. *Writing in region FE: and region FF: corresponds to writing in region 00: and region 01: respectively.*

2.4.3. Wait State Configuration Bits

You can add wait states to external bus cycles by extending the RD#/WR#/PSEN# pulse and/or extending the ALE pulse. Each additional wait state extends the pulse by $2T_{OSC}$. A separate wait state specification for external accesses via region 01: permits a slow external device to be addressed in region 01: without slowing accesses to other external devices. Table 2.2 summarizes the wait state selections for RD#,WR#,PSEN#. For waveform diagrams showing wait states see “External Bus Cycles with Configurable Wait States”.

2.4.3.1. Configuration Bits WSA1:0#, WSB1:#

The WSA1:0# wait state bits (UCONFIG0.6:5) permit RD#, WR#, and PSEN# to be extended by 1, 2, or 3 wait states for accesses to external memory via all regions except region 01:. The WSB1:0# wait state bits (UCONFIG1.2:1) permit RD#, WR#, and PSEN# to be extended by 1, 2, or 3 wait states for accesses to external memory via region 01:.

2.4.3.2. Configuration Bit WSB

Use the WSB bit only for A–stepping compatibility. The WSB wait state bit (UCONFIG1.3) permits RD#, WR#, and PSEN# to be extended by one wait state for accesses to external memory via region 01:.

2.4.3.3. Configuration Bit XALE#

Clearing XALE# (UCONFIG0.4) extends the time ALE is asserted from T_{OSC} to $3T_{OSC}$. THis accommodates an address latch that is too slow for the normal ALE signal. Paragraph 3.4.2. “Extending ALE”, shows an external bus cycle with ALE extended.

Table 2.2 RD#, WR#, PSEN# External Wait States

Regions	Configuration bits		Number of Wait States
	WSA1#	WSA0#	
00: FE: FF:	0	0	3 WS
	0	1	2 WS
	1	0	1 WS
	1	1	0 WS
	WSB1#	WSB0#	
01:	0	0	3 WS
	0	1	2 WS
	1	0	1 WS
	1	1	0 WS

2.4.4. Opcode Configurations (SRC)

The SRC configuration bit (UCONFIG0.0) selects the source mode or binary mode opcode arrangement.

Refer to TSC80251 Programmer’s Guide for information on source mode and binary mode.

2.4.5. Mapping On–chip Code Memory to Data Memory (EMAP#)

The EMAP# bit (UCONFIG1.0) provides the option of accessing the upper half of on–chip code memory as data memory. This allows code constants to be accessed as data in region 00: using direct addressing. See paragraph 1.2.2.1. “Accessing On–chip Code Memory in Region 00:”, for the exact conditions required for this mapping to be effective.

EMAP#= 0. For TSC80251G1D, the upper 8 Kbytes of on–chip code memory (FF:2000h-FF:3FFFh) are mapped to locations 00:E000h-00:FFFFh.

EMAP#= 1. Mapping of on–chip code memory to region 00: does not occur. Addresses in the range 00:E000h-00:FFFFh access external RAM.

2.4.6. Interrupt Mode (INTR)

The INTR bit (UCONFIG1.4) determines what bytes are stored on the stack when an interrupt occurs and how the RETI (Return from Interrupt) instruction restores operation.

For INTR= 0, an interrupt pushes the two lower bytes of the PC onto the stack in the following order: PC.7:0, PC.15:8. The RETI instruction pops these two bytes in the reverse order and uses them as the 16–bit return address in region FF:.

For INTR= 1, an interrupt pushes the three PC bytes and the PSW1 register onto the stack in the following order: PSW1, PC.23:16, PC.7:0, PC.15:8. The RETI instruction pops these four bytes and then returns to the specified 24–bit address, which can be anywhere in the 16 Mbytes address space.

2.5. Registers

UCONFIG0
Configuration Byte 0

7	6	5	4	3	2	1	0
–	WSA1#	WSA0#	XALE#	RD1	RD0	PAGE#	SRC

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to UCONFIG0.
6	WSA1#	**Wait State A bits** Select the number of wait states for RD#, WR# and PSEN# signals for external memory accesses (all regions except 01:). WSA1# WSA0# Number of wait states 0 0 3 0 1 2 1 0 1 1 1 0
5	WSA0#	
4	XALE#	**Extend ALE bit** Clear to extend the time of the ALE pulse from T_{OSC} to 3×T_{OSC}. Set to keep the time of the ALE pulse to T_{OSC}.
3	RD1	**Memory Signal Select bits** Codes specify a 18–bit, 17–bit or 16–bit external address bus and address ranges for RD#, WR# and PSEN# signals (see Table 2.3).
2	RD0	
1	PAGE#	**Page Mode Select bit**[1] Clear for page mode with A15:8/D7:0 on Port 2 and A7:0 on Port 0. Set for non–page mode[2] with A15:8 on Port 2 and A7:0/D7:0 on Port 0.
0	SRC	**Source Mode/Binary Mode Select bit** Clear for binary mode. Set for source mode.

Notes:
1. *UCONFIG0 is fetched twice so it can be properly read both in Page or Non–Page modes. If P2.1 is cleared during the first data phase, a page mode configuration is used, otherwise the subsequent fetches are performed in Non–Page mode (see paragraph 3.6. "Configuration Byte Bus Cycle".*
2. *This selection provides compatibility with the standard 80C51 hardware which is multiplexing the address LSB and the data on Port 0.*

Figure 2.7 Configuration Byte 0

The following Table describes in detail the address range selected with RDx bits.

Table 2.3 Memory Signal Selections (RD1:0)

RD1	RD0	P1.7	P3.7/RD#	PSEN#	WR#	External Memory
0	0	A17	A16	Read signal for all external memory locations	Write signal for all external memory locations	256 Kbytes
0	1	I/O pin	A16	Read signal for all external memory locations	Write signal for all external memory locations	128 Kbytes
1	0	I/O pin	I/O pin	Read signal for all external memory locations	Write signal for all external memory locations	64 Kbytes
1	1	I/O pin	Read signal for regions 00: and 01: (data memory)	Read signal for regions FE: and FF: (code memory)	Write signal for regions 00: and 01: (data memory)	64 Kbytes[1]

Note:
1. *This selection provides compatibility with C51 microcontrollers which have separate external memory space for data and code.*

UCONFIG1

Configuration Byte 1

7	6	5	4	3	2	1	0
–	–	–	INTR	WSB	WSB1#	WSB0#	EMAP#

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to UCONFIG1.
6	–	**Reserved** Set this bit when writing to UCONFIG1.
5	–	**Reserved** Set this bit when writing to UCONFIG1.
4	INTR	**Interrupt Mode bit**[1] Clear so that the interrupts push two bytes onto the stack (the two lower bytes of the PC register). Set so that the interrupts push four bytes onto the stack (the three bytes of the PC register and the PSW1 register).
3	WSB	**Wait State B bit**[2] Clear to generate one wait state for memory region 01:. Set for no wait states for memory region 01:.
2	WSB1#	**Wait State B bits** Select the number of wait states for RD#, WR# and PSEN# signals for external memory accesses (only region 01:). WSB1# WSB0# Number of wait states 0 0 3 0 1 2 1 0 1 1 1 0
1	WSB0#	
0	EMAP#	**On–Chip Code Memory Map bit** Clear to map the upper 8 Kbytes of on–chip code memory (FF:2000h–FF:3FFFh) to (00:E000h–00:FFFFh). Set to not map the upper 8 Kbytes of on–chip code memory (FF:2000h–FF:3FFFh). Locations (00:E000h–00:FFFFh) are implemented by external RAM.

Notes:

1. *Two or four bytes are transparently popped according to INTR when using the RETI instruction. INTR must be set if interrupts are used with code executing outside region FF:.*
2. *Use only for Step A compatibility; set this bit when WSB1:0# are used.*

Figure 2.8 Configuration Byte 1

3. External Memory Interface

3.1. Introduction

The external memory interface comprises the external bus (ports 0 and 2) as well as the bus control signals (RD#, WR#, PSEN# and ALE). Chip configuration bytes (see chapter 2. “Device Configuration”) determine several interface options: page mode or non–page mode for external code fetches; the number of external address bits (16, 17, or 18); the address ranges for RD#, WR#, and PSEN#; and the number of preprogrammed external wait states to extend RD#, WR#, PSEN# or ALE. Two kinds of real–time wait states are available: the asynchronous always enabled and the synchronous that can be enabled with special function register WCON.1:0. You can use these options to tailor the interface to your application. See also paragraph 2.4. “Configuring the External Memory Interface”.

The external memory interface operates in either page mode or non–page mode. Page mode provides increased performance by reducing the time for external code fetches. Page mode does not apply to code fetches from on–chip memory. The reset routine configures the TSC80251G1D for operation in page mode or non–page mode according to bit 1 of configuration byte UCONFIG0. Figure 3.1 shows the structure of the external address bus for page and non–page mode operation. P0 carries address A7:0 while P2 carries address A15:8. Data D7:0 is multiplexed with A7:0 on P0 in non–page mode and with A15:8 on P2 in page mode. Table 3.1 describes the external memory interface signals. The address and data signals (AD7:0 on port 0 and A15:8 on port 2) are defined for non–page mode.

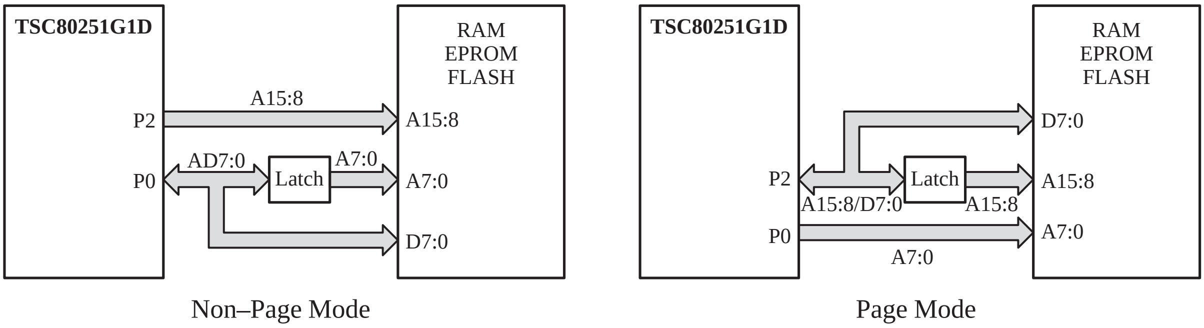


Figure 3.1 Bus Structure in Non–Page Mode and Page Mode

Table 3.1 External Memory Interface Signals

Signal Name	Type	Description	Alternative Function
A17	O	**18th Address Bit** Output to memory as 18th external address bit (A17) in extended bus applications, depending on the values of bits RD0 and RD1 in UCONFIG0 byte.	P1.7
A16	O	**17th Address Bit** Output to memory as 17th external address bit (A16) in extended bus applications, depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see also RD#).	P3.7
A15:8[1]	O	**Address Lines** Upper address lines for the external bus (non–page mode).	P2.7:0
AD7:0[1]	I/O	**Address/Data Lines** Multiplexed lower address lines and data for the external memory (non–page mode).	P0.7:0
ALE	O	**Address Latch Enable** ALE signals the start of an external bus cycle and indicates that valid address information are available onlines A16/A17 and A7:0.	–
AWAIT#	I	**Real–time Asynchronous Wait States Input** When this pin is active (low level), the memory cycle is stretched until it becomes high.	–
EA#	I	**External Access Enable** EA# directs program memory accesses to on–chip or off–chip code memory. For EA#= 0, all program memory accesses are off-chip. For EA#= 1, an access is on-chip ROM if the address is within the range of the on–chip ROM; otherwise the access is off-chip. The value of EA# is latched at reset. For devices without ROM on-chip, EA# must be strapped to ground.	–
PSEN#	O	**Program Store Enable/Read signal output** PSEN# is asserted for a memory address range that depends on bits RD0 and RD1 in UCONFIG0 byte (see also RD#). RD1 RD0 Addresses Range for Assertion 0 0 All addresses 0 1 All addresses 1 0 All addresses 1 1 All addresses $\geq$ 80:0000h.	–
RD#	O	**Read or 17th Address Bit (A16)** Read signal output to external data memory or 17th external address bit (A16), depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see also PSEN#). RD1 RD0 Function 0 0 The pin functions as A16 only 0 1 The pin functions as A16 only 1 0 The pin functions as P3.7 only 1 1 RD# asserted for reads at all addresses $\leq$ 7F:FFFFh.	P3.7
WAIT#	I	**Real–time Synchronous Wait States Input** The real–time synchronous WAIT# input is enabled by setting RTWE bit in WCON (S:A7h). During bus cycles, the external memory system can signal 'system ready' to the microcontroller in real time by controlling the WAIT# input signal on the port 1.6 input.	P1.6
WCLK	O	**Wait Clock Output** The real–time WCLK output is enabled by setting RTWCE bit in WCON (S:A7h). When enabled, the WCLK output produces a square wave signal with a period of one half the oscillator frequency on the port P1.7 output.	P1.7
WR#	O	**Write** Write signal output to external memory. WR# is asserted for writes to all valid memory locations	P3.6

Note:

[1] *If the chip is configured for page–mode operation, port 0 carries the lower address bits (A7:0), and port 2 carries the upper address bits (A15:8) and the data (D7:0).*

3.2. External Bus Cycles

This paragraph describes the bus cycles the TSC80251G1D executes to fetch code, read data, and write data in external memory. Both page mode and non–page mode are described and illustrated. For simplicity, the accompanying figures depict the bus cycle waveforms in idealized form and do not provide precise timing information. This section does not cover wait states (see paragraph 3.4. “External Bus Cycles with Configurable Wait States”) or configuration byte bus cycles (see paragraph 3.6. “Configuration Byte Bus Cycles”). For bus cycle timing parameters refer to the TSC80251G1D datasheet.

An “inactive external bus” exists when the TSC80251G1D is not executing external bus cycles. This occurs under any of the three following conditions:

- Bus Idle (The chip is in normal operating mode but no external bus cycles are executing)
- The chip is in idle mode
- The chip is in powerdown mode

3.2.1. Bus Cycle Definitions

Table 3.2 lists the types of external bus cycles. It also shows the activity on the bus for non–page mode and page mode bus cycles with no wait states. There are three types of non–page mode bus cycles: code read, data read, and data write. There are four types of page mode bus cycles: code fetch (page miss), code read (page hit), data read, and data write. The data read and data write cycles are the same for page mode and non–page mode (except the multiplexing of D7:0 on ports 0 and 2).

Table 3.2 Bus Cycle Definitions (No Wait States)

Mode	Bus Cycle	Bus Activity[1]		
		State 1	State 2	State 3
Non–Page Mode	Code Read	ALE	RD#/PSEN#, code in	
	Data Read[2]	ALE	RD#/PSEN#	data in
	Data Write[2]	ALE	WR#	WR# high, data out
Page Mode	Code Read, Page Miss	ALE	RD#/PSEN#, code in	
	Code Read, Page Hit[3]	PSEN#, code in		
	Data Read[2]	ALE	RD#/PSEN#	data in
	Data Write[2]	ALE	WR#	WR# high, data out

Notes:
1. *Signal timing implied by this table is approximate (idealized).*
2. *Data read (page mode)= data read (non–page mode) and data write (page mode)= data write (non–page mode) except that in page mode data appears on P2 (multiplexed with A15:0), whereas in non–page mode data appears on P0 (multiplexed with A7:0).*
3. *The initial code read page hit bus cycle can execute only following a code read page miss cycle.*

3.2.2. Non–Page Mode Bus Cycles

In non–page mode, the external bus structure is the same as for C51 microcontrollers. The upper address bits (A15:8) are on port 2, and the lower address bits (A7:0) are multiplexed with the data (D7:0) on port 0. External code read bus cycles execute in approximately two state times (see Table 3.2 and Figure 3.2). External data read bus cycles (see Figure 3.3) and external write bus cycles (see Figure 3.4) execute in approximately three state times. For the write cycle (see Figure 3.4), a third state is appended to provide recovery time for the bus. Note that the write signal WR# is asserted for all memory regions, except for the case of RD1:0= 11, where WR# is asserted for regions 00:-01: but **not** for regions FE:-FF:.

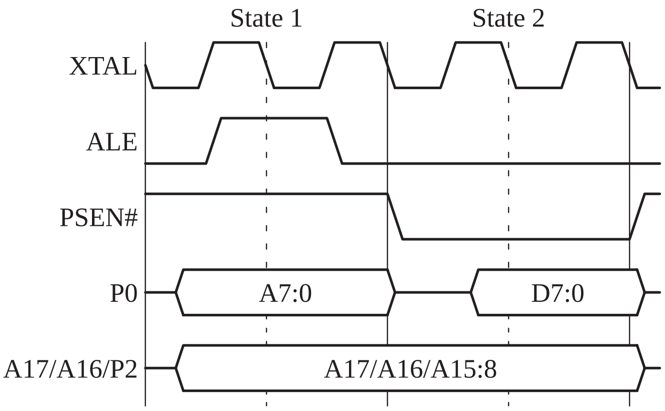


Figure 3.2 External Code Fetch (Non–Page Mode)

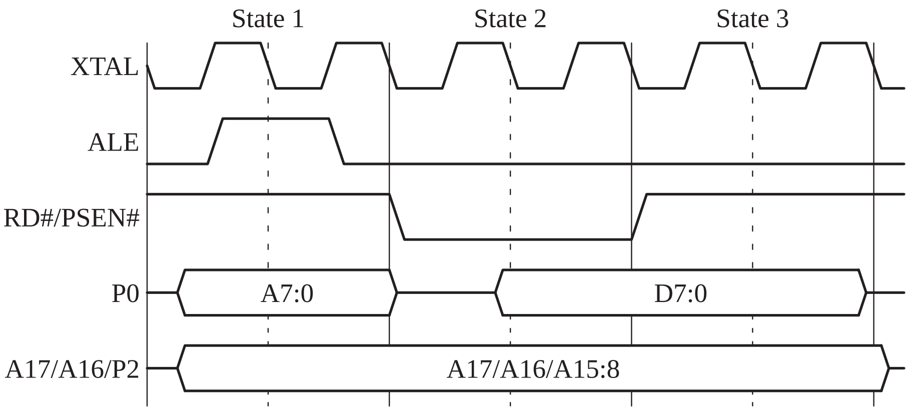


Figure 3.3 External Data Read (Non–Page Mode)

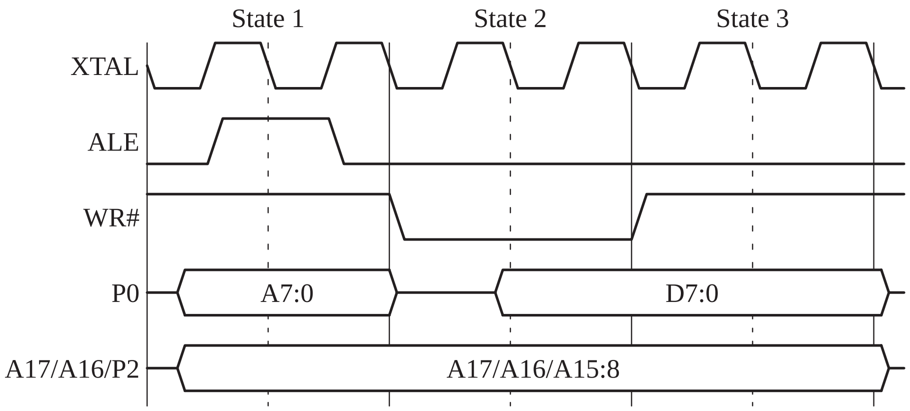


Figure 3.4 External Data Write (Non–Page Mode)

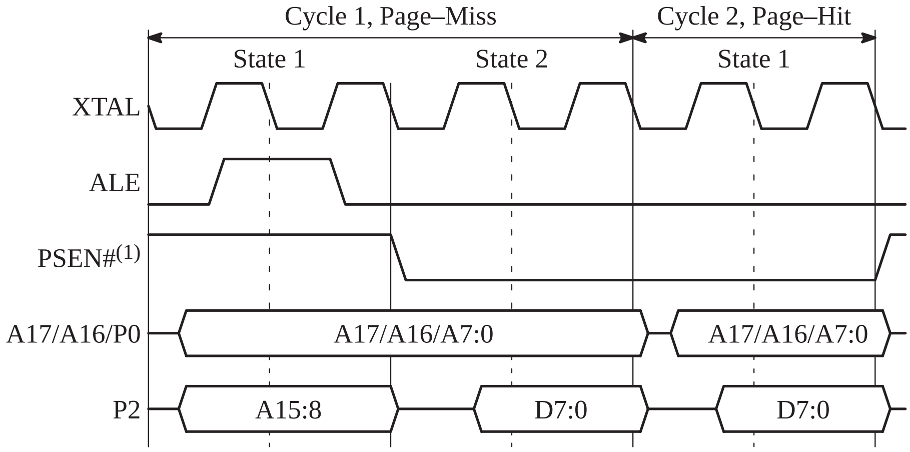
3.2.3. Page Mode Bus Cycles

Page mode increases performance by reducing the time for external code fetches. Under certain conditions the controller fetches an instruction from external memory in one state time instead of two (Table 3.2). Page mode does not affect internal code fetches. The first code fetch to a 256–byte "page" of memory always uses a two–state bus cycle. Subsequent successive code fetches to the same page (*page hits*) require only a one–state bus cycle. When a subsequent fetch is to a different page (a *page miss*) it again requires a two–state bus cycle. The following external code fetches are always page–miss cycles:

- the first external code fetch after a page rollover
- the first external code fetch after an external data bus cycle
- the first external code fetch after powerdown or idle mode
- the first external code fetch after a branch, return, interrupt, etc

In page mode, the TSC80251G1D bus structure differs from the bus structure in C51 microcontrollers (see Figure 3.1). The upper address bits A15:8 are multiplexed with the data D7:0 on port 2, and the lower address bits (A7:0) are on port 0.

Figure 3.5 shows the two types of external bus cycles for code fetches in page mode. The page–miss cycle is the same as a code fetch cycle in non–page mode (except D7:0) is multiplexed with A15:8 on P2.). For the page–hit, the upper eight address bits are the same as for the preceding cycle. Therefore, ALE is not asserted, and the values of A15:8 are retained in the address latches. In a single state, the new values of A7:0 are placed on port 0, and memory places the instruction byte on port 2. Notice that a page hit reduces the available address access time by one state. Therefore, faster memories may be required to support page mode.



Note:
1. During a sequence of page hits, PSEN# remains low until the end of the last page–hit cycle.

Figure 3.5 External Code Fetch (Page Mode)

Figure 3.6 and Figure 3.7 show the bus cycles for data reads and data writes in page mode. These cycles are identical to those for non–page mode, except for the different signals on ports 0 and 2.

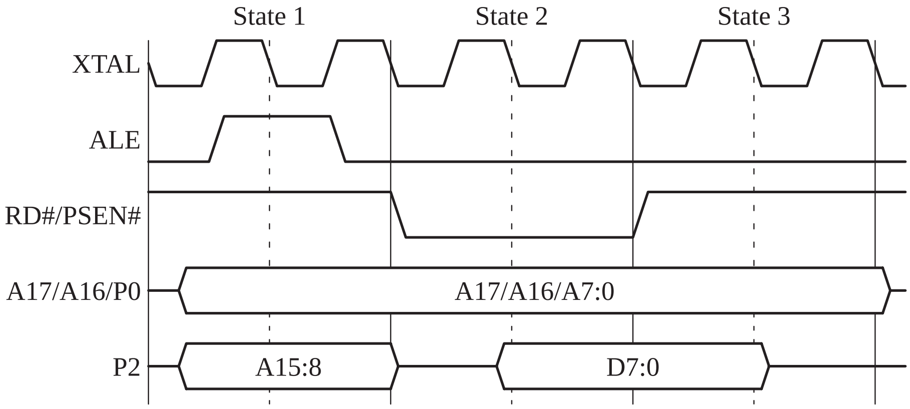


Figure 3.6 External Data Read (Page Mode)

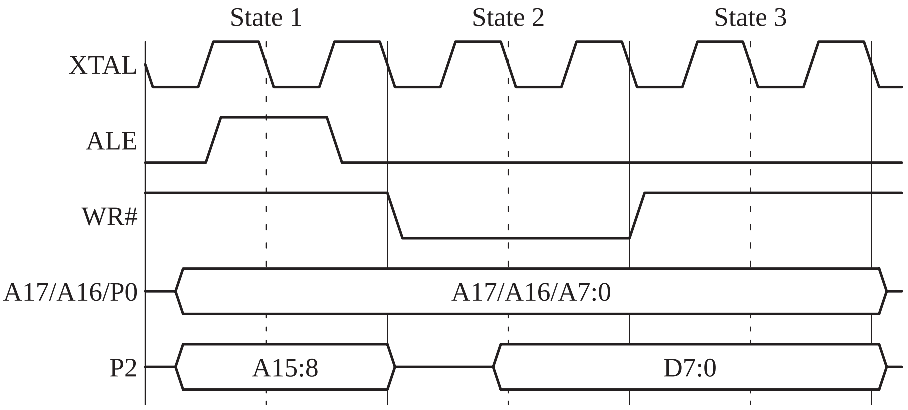


Figure 3.7 External Data Write (Page Mode)

3.3. Wait States

The TSC80251G1D provides four types of wait state solutions to external memory problems: real–time, RD#/WR#/PSEN#, and ALE wait states. The TSC80251G1D supports also two types of real–time wait state operations for dynamic bus control: real–time asynchronous and synchronous wait state. See paragraph 3.5. “External Bus Cycles with Real–time Wait States.” In addition, the TSC80251G1D can be configured at reset to add wait states to external bus cycles by extending the ALE or RD#/WR#/PSEN# pulses. See paragraph 2.4.3. “Wait State Configuration Bits”. You can configure the chip to use multiple types of wait states. Accesses to on–chip code and data memory always use zero wait states. The following paragraphs demonstrate wait state usage.

3.4. External Bus Cycles with Configurable Wait States

Three types of wait state solutions are available; real time, RD#/WR#/PSEN#, and ALE wait states. The TSC80251G1D can be configured to add wait states to the external bus cycles extending the ALE pulse or by adding 0, 1, 2, or 3 wait states to the RD#/WR#/PSEN# pulses.

The XALE# configuration bit specifies 0 or 1 wait state for ALE. The WSA1:0# and WSB1:0# configuration bits specify the number of wait states for RD/WR/PSEN. See paragraph 2.4.3. “Wait State Configuration Bits.”

3.4.1. Extending RD#/WR#/PSEN#

Figure 3.8 shows the non–page mode code fetch bus cycle with one RD#/PSEN# wait state. The wait state extends the bus cycle to three states. Figure 3.9 shows the non–page mode data write bus cycle with one WR# wait state. The wait state extends the bus cycle to four states. The waveforms in Figure 3.9 also apply to the non–page mode data read external bus cycle if RD#/PSEN# is substituted for WR#.

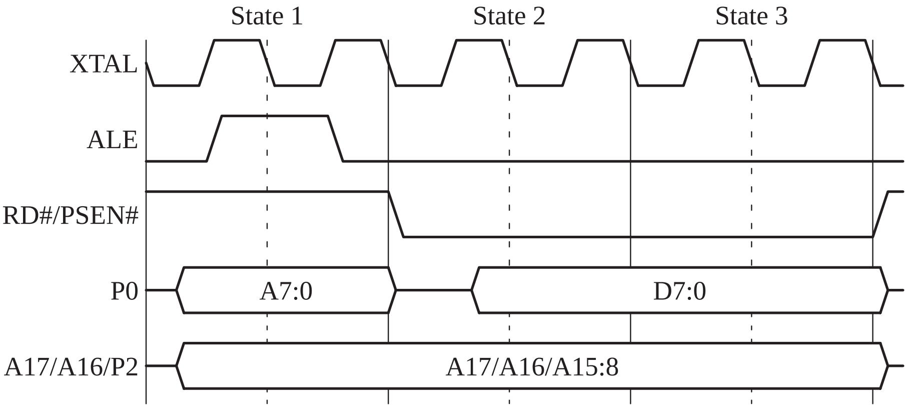


Figure 3.8 External Code Fetch (Non–Page Mode, One RD#PSEN# Wait State)

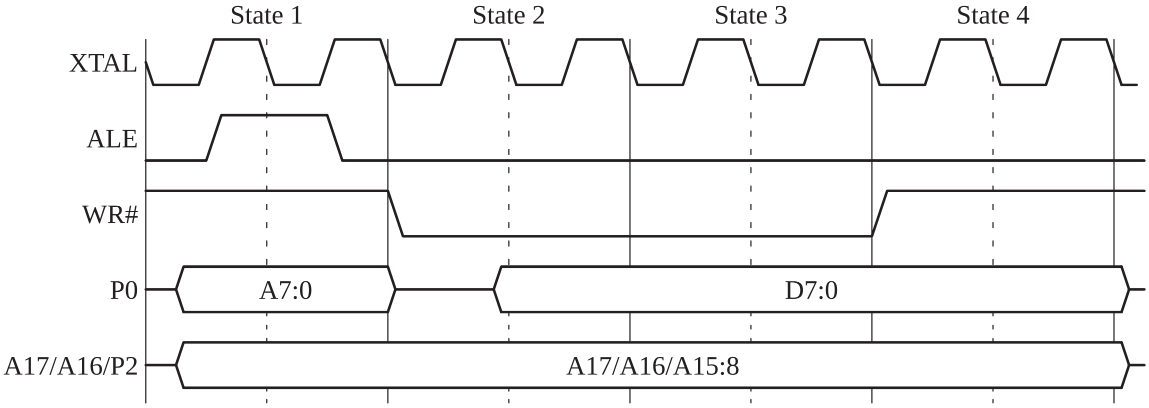


Figure 3.9 External Data Write (Non–Page Mode, One WR# Wait State)

3.4.2. Extending ALE

Figure 3.10 shows the non–page mode code fetch external bus cycle with ALE extended. The wait state extends the bus cycle from two states to three. For read and write external bus cycles, the extended ALE extends the bus cycle from three states to four.

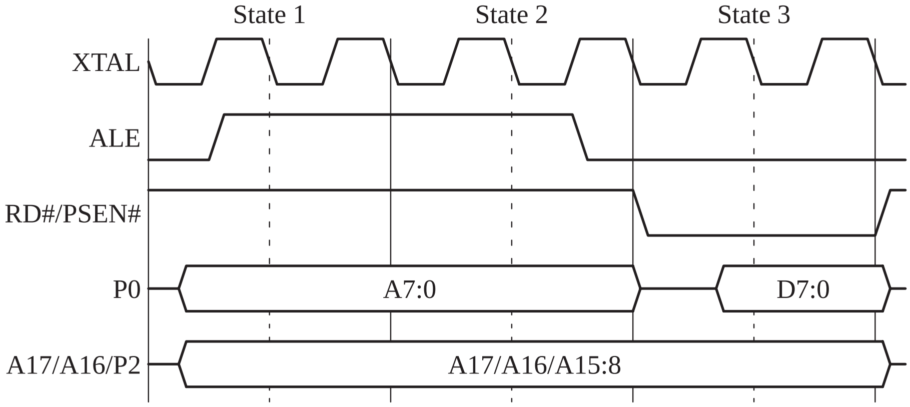


Figure 3.10 External Code Fetch (Non–Page Mode, One ALE Wait State)

3.5. External Bus Cycles with Real–Time Wait States

In addition to fixed–length wait states such as RD#/WR#/PSEN# and ALE, the TSC80251G1D offers two types of real–time wait state: the real–time asynchronous wait state and a real time synchronous wait state. The programmer can dynamically adjust the delay of the real–time wait state.

3.5.1. Real–Time Asynchronous Wait state

The real–time asynchronous AWAIT# input is always enabled. During bus cycles, the external memory system can signal “system ready” to the microcontroller in real time by controlling the AWAIT# input signal located on a separate dedicated input pin. Sampling of AWAIT# is coincident with the middle of RD#/PSEN# or WR# signals driven low during a bus cycle. A ‘not–ready’ condition is recognized by the AWAIT# signal held at V_{IL} by the external memory system.

3.5.2. Real–Time Asynchronous Wait State Bus Cycle Diagrams

Figure 3.11 shows the code fetch/data read bus cycle and Figure 3.12 shows the data write bus cycle in page mode and non–page mode. The page mode and non–page mode differs only on the port 0 and port 2 operation. In non–page mode, port 0 carries the lower address bits (A7:0) and the data (D7:0) and port 2 carries the upper address bits (A15:8). In page mode, port 0 carries the lower address bits (A7:0), and port 2 carries the upper address bits (A15:8) and the data (D7:0). All real–time asynchronous wait times are illustrated in the TSC80251G1D Datasheet.

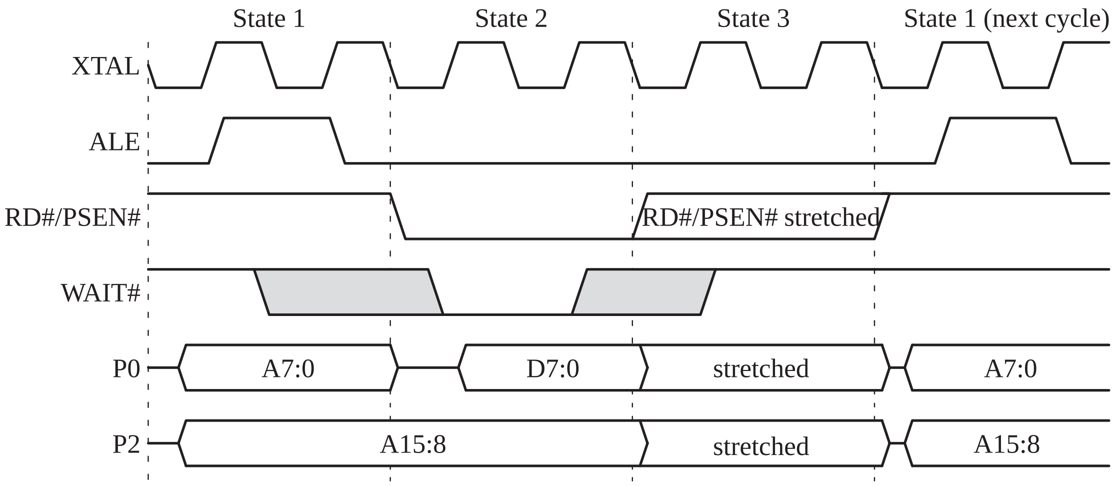


Figure 3.11 External Code Fetch/Data Read (Real–Time Asynchronous Wait State)

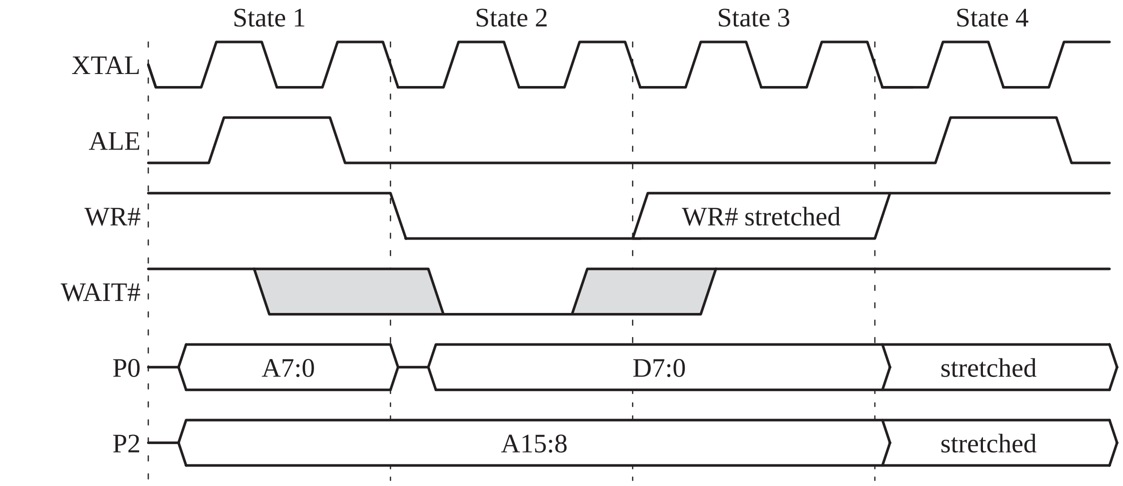


Figure 3.12 External Data Write (Real–Time Asynchronous Wait State)

3.5.3. Real–Time Synchronous Wait State

The real–time synchronous WAIT# input is enabled by writing a logical ‘1’ to the WCON.0 (RTWE) bit at S:A7h (see Figure 3.16). During bus cycles, the external memory system can signal “system ready” to the microcontroller in real time by controlling the WAIT# input signal on the port 1.6 input. Sampling of WAIT# is coincident with the activation of RD#/PSEN# or WR# signals driven low during a bus cycle. A ‘not–ready’ condition is recognized by the WAIT# signal held at V $_{IL}$ by the external memory system. Use of the SSLC or the PCA module 3 may conflict with your design. Do not use CEX3/SCL/SCK interchangeably with the WAIT# signal on the port 1.6 input.

3.5.4. Real–Time Synchronous Wait Clock

The real–time synchronous WAIT CLOCK output is driven at port 1.7 (WCLK) by writing a logical ‘1’ to the WCON.1 (RTWCE) bit at S:A7h (see Figure 3.16). When enabled, the WCLK output produces a square wave signal with a period of one–half the oscillator frequency. Use of the SSLC or the PCA module 4 may conflict with your design. Do not use CEX4/SDA/MOSI interchangeably with WCLK output. Use of address signal A17 disables WCLK, CEX4 and SDA/MOSI operation at the port 1.7 output.

3.5.5. Real–Time Synchronous Wait State Bus Cycle Diagrams

Figure 3.13 shows the code fetch/data read bus cycle in and Figure 3.14 shows the data write bus cycle in page mode and non–page mode. The page mode and non–page mode differs only on the port 0 and port 2 operation. In non–page mode, port 0 carries the lower address bits (A7:0) and the data (D7:0) and port 2 carries the upper address bits (A15:8). In page mode, port 0 carries the lower address bits (A7:0), and port 2 carries the upper address bits (A15:8) and the data (D7:0). All the real–time synchronous wait times are illustrated in the TSC80251G1D Datasheet.

Caution:
The real–time synchronous wait function has critical external timing for code fetch. For this reason, it is not advisable to use the real–time synchronous wait feature for code fetch in page mode.

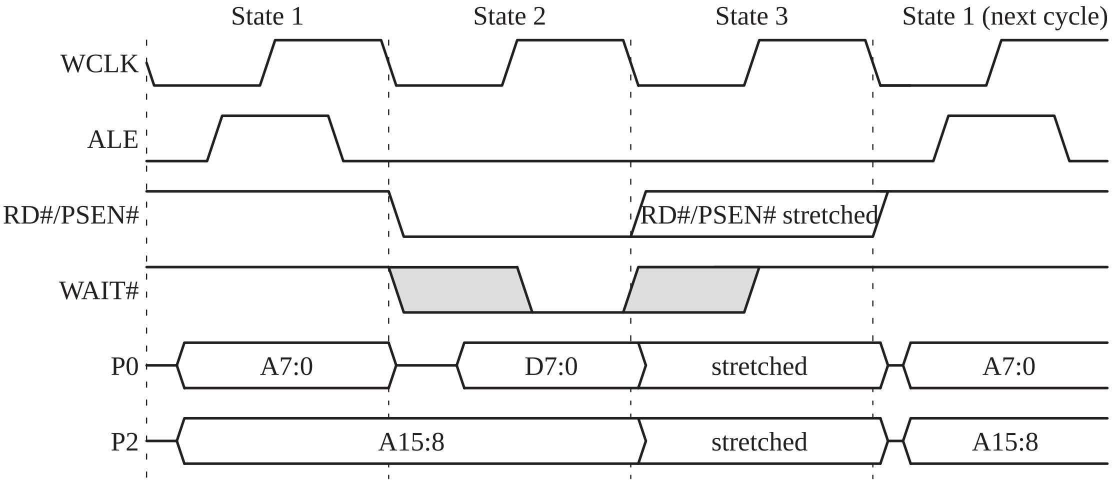


Figure 3.13 External Code Fetch/Data Read (Real Time Synchronous Wait State)

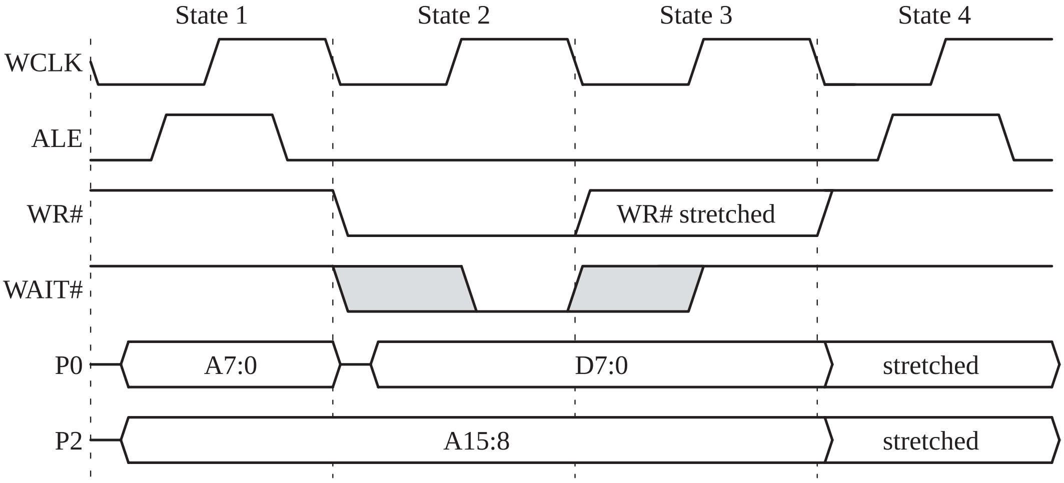


Figure 3.14 External Data Write (Real Time Synchronous Wait State)

3.6. Configuration Byte Bus Cycles

If EA#= 0, devices obtain configuration information from a configuration array in external memory. This section describes the bus cycles executed by the reset routine to fetch user configuration bytes from external memory. Configuration bytes are discussed in chapter 2. “Device Configuration”. To determine whether the external memory is set up for page mode or non–page mode operation, the TSC80251G1D accesses external memory using internal address FF:FFF8h (UCONFIG0). See states 1 to 4 in Figure 3.15. If the external memory is set up for page mode, it places UCONFIG0 on P2 as D7:0, overwriting A15:8 (FFh). If external memory is set up for non–page mode, A15:8 is not overwritten. The TSC80251G1D examines P2 bit 1. Subsequent configuration byte fetches are in page mode if P2.1= 0 and in non–page mode if P2.1= 1. The TSC80251G1D fetches UCONFIG0 again (states 5 to 8 in Figure 3.15) and then UCONFIG1 via internal address FF:FFF9h.

The configuration byte bus cycles always execute with ALE extended and one PSEN# wait state.

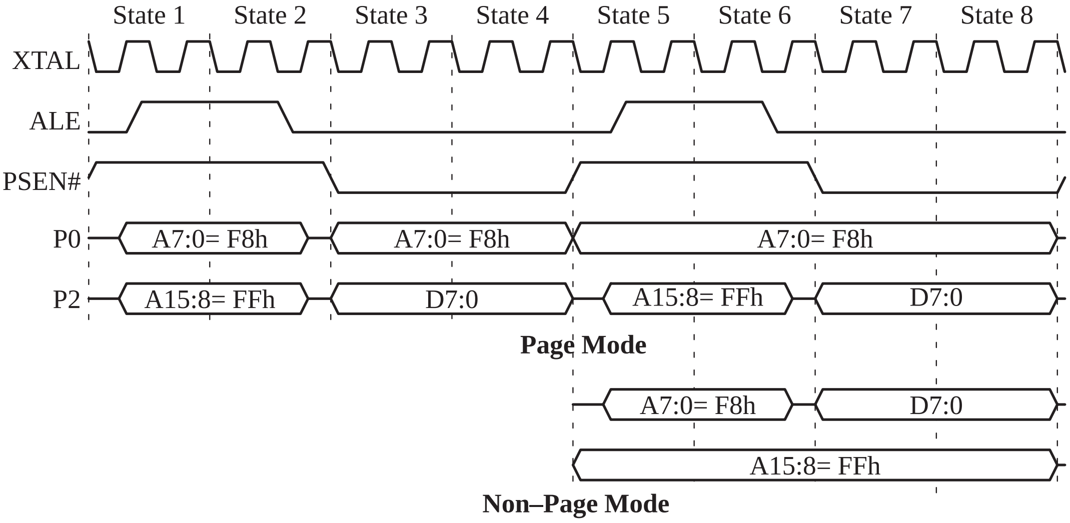


Figure 3.15 Configuration Byte Bus Cycles

3.7. Port 0 and Port 2 Status

This section summarizes the status of the port 0 and port 2 pins when these ports are used as the external bus. A more comprehensive description of the ports and their use is given in chapter 4.4. “Port 0 and Port 2” When port 0 and port 2 are used as the external memory bus, the signals on the port pins can originate from three sources:

- the TSC80251G1D CPU (address bits, data bits)
- the port SFRs: P0 and P2 (logic levels)
- an external device (data bits)

The port 0 pins (but not the port 2 pins) can also be held in a high–impedance state. Table 3.3 lists the status of the port 0 and port 2 pins when the chip is in the normal operating mode and the external bus is idle or executing a bus cycle.

Table 3.3 Port 0 and Port 2 Pin Status in Normal Operating Mode

Port	8–bit/16–bit Addressing	Non–Page Mode		Page Mode	
		Bus Cycle	Bus Idle	Bus Cycle	Bus Idle
Port 0	8 or 16	AD7:0(1)	High Impedance(3)	A7:0(1)	High Impedance(3)
Port 2	8	P2(2)	P2	P2/D7:0(2)	High Impedance(4)
	16	A15:8	P2	A15:8/D7:0	High Impedance(4)

Notes:
1. *During external memory accesses, the CPU writes FFh to the P0 register which content is lost.*
2. *The P2 register can be used to select one 256–byte page in external memory.*
3. *P0 content is output when written to.*
4. *P2 content is not output when written to.*

3.7.1. Port 0 and Port 2 Pin Status in Non–Page Mode

In non–page mode, the port pins have the same signals as those on the 80C51. For an external memory instruction using a 16–bit address, the port pins carry address and data bits during the bus cycle. However, if the instruction uses an 8–bit address (e.g., MOVX @Ri), the contents of P2 are driven onto the pins. These pin signals can be used to select 256–bit pages in external memory. During a bus cycle, the CPU always writes FFh to P0, and the former contents of P0 are lost. A bus cycle does not change the contents of P2. When the bus is idle, the port 0 pins are held at high impedance, and the contents of P2 are driven onto the port 2 pins.

3.7.2. Port 0 and Port 2 Pin Status in Page Mode

In a page–mode bus cycle, the data is multiplexed with the upper address byte on port 2. However, if the instruction uses an 8–bit address (e.g., MOVX @Ri), the contents of P2 are driven onto the pins when data is not on the pins. These logic levels can be used to select 256–bit pages in external memory. During bus idle, the port 0 and port 2 pins are held at high impedance. (For port pin status when the chip in is idle mode, powerdown mode, or reset, see chapter 11. “Power Monitoring and Management”)

3.8. Registers

WCON (S:A7h)

Real–Time Synchronous Wait State Control Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	RTWCE	RTWE

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	RTWCE	**Real–Time Synchronous WAIT CLOCK enable** Clear to disable synchronous WAIT CLOCK. Set to enable the synchronous WAIT CLOCK on port 1.7 (WCLK). The square wave output signal is one–half the oscillator frequency.
0	RTWE	**Real–Time Synchronous WAIT# enable** Clear to disable real–time synchronous wait state. Set to enable real–time synchronous wait state input on port 1.6 (WAIT#).

Reset Value= 00XX X000b

Figure 3.16 Real–Time Synchronous Wait State Control Register (WCON)

4. Input/Output Ports

4.1. Introduction

The TSC80251G1D uses input/output (I/O) Ports to exchange data with external devices. In addition to performing general–purpose I/O, some Ports are capable of external memory operations; others allow for alternate functions. All four TSC80251G1D I/O Ports are bidirectional. Each Port contains a latch, an output driver and an input buffer. Port 0 and Port 2 output drivers and input buffers facilitate external memory operations. Port 0 drives the lower address byte onto the parallel address bus and Port 2 drives the upper address byte onto the bus. In non–page mode, the data is multiplexed with the lower address byte on Port 0. In page mode, the data is multiplexed with the upper address byte on Port 2. All Port 1 and Port 3 pins serve for both general–purpose I/O and alternate functions (see Table 4.1 to Table 4.4).

Table 4.1 Port 0 Pin Descriptions

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P0.0	I/O	AD0 A0	Address/Data line 0 (Non–page mode) Address line 0 (Page mode)	I/O O
P0.1	I/O	AD1 A1	Address/Data line 1 (Non–page mode) Address line 1 (Page mode)	I/O O
P0.2	I/O	AD2 A2	Address/Data line 2 (Non–page mode) Address line 2 (Page mode)	I/O O
P0.3	I/O	AD3 A3	Address/Data line 3 (Non–page mode) Address line 3 (Page mode)	I/O O
P0.4	I/O	AD4 A4	Address/Data line 4 (Non–page mode) Address line 4 (Page mode)	I/O O
P0.5	I/O	AD5 A5	Address/Data line 5 (Non–page mode) Address line 5 (Page mode)	I/O O
P0.6	I/O	AD6 A6	Address/Data line 6 (Non–page mode) Address line 6 (Page mode)	I/O O
P0.7	I/O	AD7 A7	Address/Data line 7 (Non–page mode) Address line 7 (Page mode)	I/O O

Table 4.2 Port 1 Pin Descriptions

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P1.0	I/O	T2 P1.0	Timer 2 external clock input/output Keyboard input 0	I/O I
P1.1	I/O	T2EX P1.1	Timer 2 external input Keyboard input 1	I I
P1.2	I/O	ECI P1.2	EWC external clock input Keyboard input 2	I I
P1.3	I/O	CEX0 P1.3	EWC module 0 Capture input/PWM output Keyboard input 3	I/O I
P1.4	I/O	CEX1 P1.4	EWC module 1 Capture input/PWM output Keyboard input 4	I/O I
P1.5	I/O	CEX2 MISO P1.5	EWC module 2 Capture input/PWM output µWire/SPI master input slave output Keyboard input 5	I/O I/O I

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P1.6	I/O	CEX3 SCL SCK P1.6 WAIT#	EWC module 3 Capture input/PWM output I^{2}C serial clock µWire/SPI serial clock Keyboard input 6 Real–time Synchronous Wait state input	I/O O O I I
P1.7	I/O	A17 CEX4 SDA MOSI P1.7 WCLK	Address line 17 EWC module 4 Capture input/PWM output I^{2}C serial data µWire/SPI master output slave input Keyboard input 7 Real–time synchronous Wait Clock output	I/O I/O I/O I/O I O

Table 4.3 Port 2 Pin Descriptions

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P2.0	I/O	A8 A8/D0	Address line 8 (Non–page mode) Address line 8/Data line 0 (Page mode)	O I/O
P2.1	I/O	A9 A9/D1	Address line 9 (Non–page mode) Address line 9/Data line 1 (Page mode)	O I/O
P2.2	I/O	A10 A10/D2	Address line 10 (Non–page mode) Address line 10/Data line 2 (Page mode)	O I/O
P2.3	I/O	A11 A11/D3	Address line 11 (Non–page mode) Address line 11/Data line 3 (Page mode)	O I/O
P2.4	I/O	A12 A12/D4	Address line 12 (Non–page mode) Address line 12/Data line 4 (Page mode)	O I/O
P2.5	I/O	A13 A13/D5	Address line 13 (Non–page mode) Address line 13/Data line 5 (Page mode)	O I/O
P2.6	I/O	A14 A14/D6	Address line 14 (Non–page mode) Address line 14/Data line 6 (Page mode)	O I/O
P2.7	I/O	A15 A15/D7	Address line 15 (Non–page mode) Address line 15/Data line 7 (Page mode)	O I/O

Table 4.4 Port 3 Pin Descriptions

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P3.0	I/O	RXD	Serial Port Receive Data input	I
P3.1	I/O	TXD	Serial Port Transmit Data output	O
P3.2	I/O	INT0#	External Interrupt 0	I
P3.3	I/O	INT1#	External Interrupt 1	I
P3.4	I/O	T0	Timer 0 input	I
P3.5	I/O	T1	Timer 1 input	I
P3.6	I/O	WR#	Write signal to external memory	O
P3.7	I/O	RD# A16	Read signal to external memory Address line 16	O O

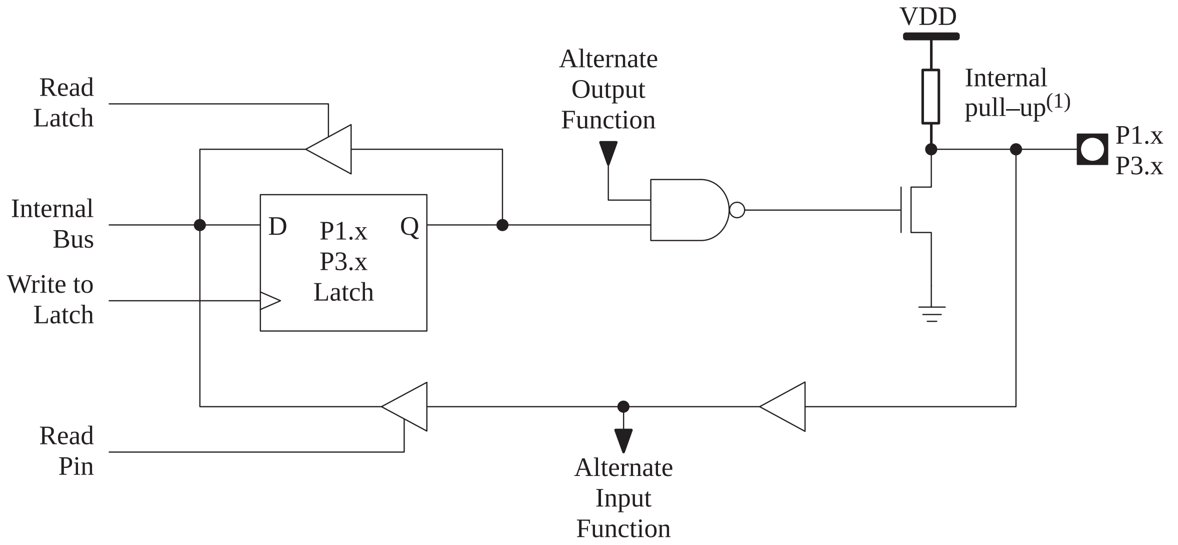
Notes:
EWC= Event Waveform Controller
I^{2}C= Inter–Integrated Circuit
PWM= Pulse Width Modulation
SPI= Serial Peripheral Interface

4.2. I/O Configurations

Each Port SFR operates via type–D latches, as illustrated in Figure 4.1 for Ports 1 and 3. A CPU “write to latch” signal initiates transfer of internal bus data into the type–D latch. A CPU “read latch” signal transfers the latched Q output onto the internal bus. Similarly, a “read pin” signal transfers the logical level of the Port pin. Some Port data instructions activate the “read latch” signal while others activate the “read pin” signal. Latch instructions are referred to as Read–Modify–Write instructions (see paragraph 4.5. “Read–Modify–Write Instructions”). Each I/O line may be independently programmed as input or output.

4.3. Port 1 and Port 3

Figure 4.1 shows the structure of Ports 1 and 3, which have internal pull–ups. An external source can pull the pin low. Each Port pin can be configured either for general–purpose I/O or for its alternate input or output function (see Table 4.2 and Table 4.4).
To use a pin for general–purpose output, set or clear the corresponding bit in the Px register (x= 1 or 3). To use a pin for general–purpose input, set the bit in the Px register. This turns off the output FET drive.
To configure a pin for its alternate function, set the bit in the Px register. When the latch is set, the “alternate output function” signal controls the output level (see Figure 4.1). The operation of Ports 1 and 3 is discussed further in “Quasi–Bidirectional Port Operation” paragraph.



Note:
1. The internal pull–up is disabled on P1.6/SCL and P1.7/SDA when I^{2}C interface is enabled (open–drain structure).

Figure 4.1 Port 1 and Port 3 Structure

4.4. Port 0 and Port 2

Ports 0 and 2 are used for general–purpose I/O or as the external address/data bus. Port 0, shown in Figure 4.2, differs from the other Ports in not having internal pull–ups. Figure 4.3 shows the structure of Port 2. An external source can pull a Port 2 pin low.
To use a pin for general–purpose output, set or clear the corresponding bit in the Px register (x= 0 or 2). To use a pin for general–purpose input set the bit in the Px register to turn off the output driver FET.

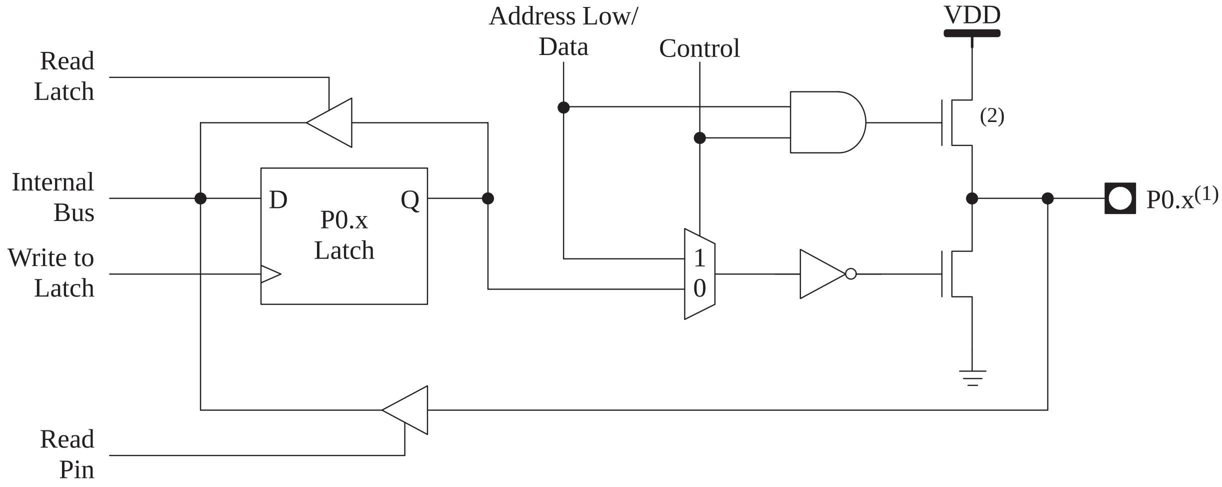
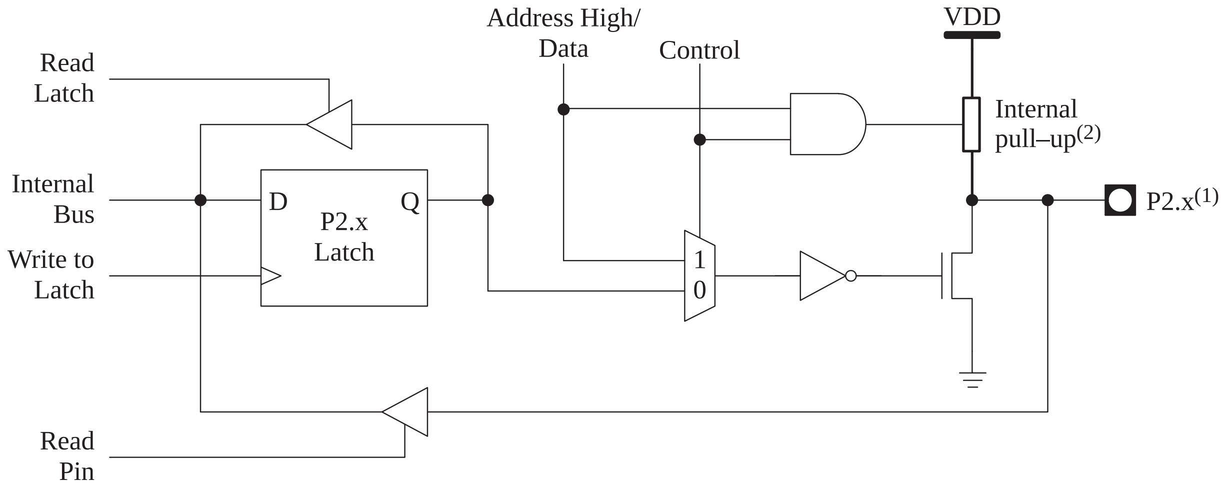


Figure 4.2 Port 0 Structure

Notes:
1. *Port 0 is precluded from use as general purpose I/O Ports when used as address/data bus drivers.*
2. *Port 0 internal strong pull–ups assist the logic–one output for memory bus cycles only. Except for these bus cycles, the pull–up FET is off, Port 0 outputs are open–drain.*



Notes:
1. *Port 2 is precluded from use as general purpose I/O Ports when used as address/data bus drivers.*
2. *Port 2 internal strong pull–ups FET (p1 in Figure 4.4) assist the logic–one output for memory bus cycles.*

Figure 4.3 Port 2 Structure

When Port 0 and Port 2 are used for an external memory cycle, an internal control signal switches the output–driver input from the latch output to the internal address/data line. Paragraph 3.2. “External Bus Cycles” describes the TSC80251G1D bus cycles, and paragraph 3.7. “Port 0 and Port 2 Status” summarizes the status of the port 0 and port 2 pins when these ports are used as the external bus.

4.5. Read–Modify–Write Instructions

Some instructions read the latch data rather than the pin data. The latch based instructions read the data, modify the data and then rewrite the latch. These are called “Read–Modify–Write” instructions. Below is a complete list of these special instructions (see Table 4.5). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Table 4.5 Read–Modify–Write Instructions

Instruction	Description	Example
ANL	logical AND	ANL P1,A
ORL	logical OR	ORL P2,A
XRL	logical EX–OR	XRL P3,A
JBC	jump if bit= 1 and clear bit	JBC P1.1, LABEL
CPL	complement bit	CPL P3.0
INC	increment	INC P2
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

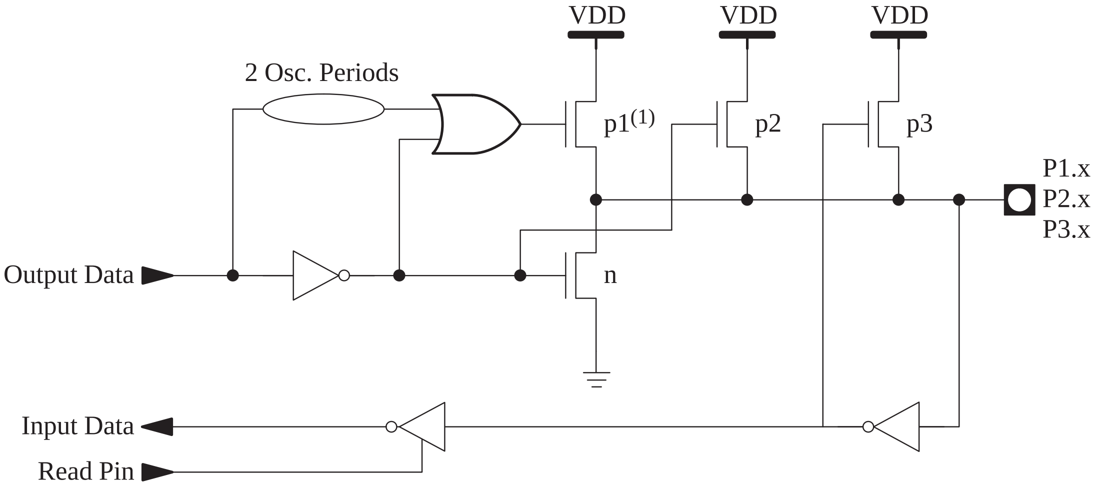
It is not obvious the last three instructions in this list are Read–Modify–Write instructions. These instructions read the Port (all 8 bits), modify the specifically addressed bit and write the new byte back to the latch. These Read–Modify–Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor can not rise above the transistor’s base–emitter junction voltage (a value lower than V_{IL}). With a logic one written to the bit, attempts by the CPU to read the Port at the pin are misinterpreted as logic zero. A read of the latch rather than the pin returns the correct logic–one value.

4.6. Quasi–Bidirectional Port Operation

Port 1, Port 2 and Port 3 have fixed internal pull–ups and are referred to as “quasi–bidirectional” Ports. When configured as an input, the pin impedance appears as logic one and sources current in response to an external logic zero condition. Port 0 is a “true bidirectional” pin. The pins float when configured as input. Resets write logical one to all Port latches. If logical zero is subsequently written to a Port latch, it can be returned to input conditions by a logical one written to the latch.

Note:
Port latch values change near the end of Read–Modify–Write instruction cycles. Output buffers (and therefore the pin state) update early in the instruction after the Read–Modify–Write instruction cycle.

Logical zero–to–one transitions in Port 1, Port 2 and Port 3 use an additional pull–up (p1) to aid this logic transition (see Figure 4.4). This increases switch speed. This extra pull–up sources 100 times normal internal circuit current during 2 oscillator clock periods. The internal pull–ups are field–effect transistors rather than linear resistors. Pull–ups consist of three p–channel FET (pFET) devices. A pFET is on when the gate senses logical zero and off when the gate senses logical one. pFET #1 is turned on for two oscillator periods immediately after a zero–to–one transition in the Port latch. A logical one at the Port pin turns on pFET #3 (a weak pull–up) through the inverter. This inverter and pFET pair form a latch to drive logical one. pFET #2 is a very weak pull–up switched on whenever the associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3.



Note:

1. Port 2 p1 assists the logic–one output for memory bus cycles.

Figure 4.4 Internal Pull–Up Configurations

4.7. Port Loading

The current–sink capability of Port 1, Port 2 and Port 3 is reported in the "AC Characteristics" section of the TSC80251G1D datasheet. These Port pins can be driven by open–collector and open–drain devices. Logic zero–to–one transitions occur slowly as limited current pulls the pin to a logic–one condition (see Figure 4.4). A logic zero input turns off pFET #3. This leaves only pFET #2 weakly in support of the transition. The current–sink capability of Port 0 is reported in the "AC Characteristics" section of the TSC80251G1D datasheet. However, the Port 0 pins require external pull–ups to drive external gate inputs. External circuits must be designed to limit current requirements to these conditions.

5. Timers/Counters

5.1. Introduction

The TSC80251G1D contains three general–purpose, 16–bit Timers/Counters. Although they are identified as Timer 0, Timer 1, and Timer 2, you can independently configure each to operate in a variety of modes as a Timer or as an event Counter. When operating as a Timer, a Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, a Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request.

The Timer registers and associated control and capture registers are implemented as addressable Special Function Registers (SFRs). Four of the SFRs provide programmable control of the Timers as follows:

- Timer/Counter mode control register (TMOD) and Timer/Counter control register (TCON) control Timer 0 and Timer 1.
- Timer/Counter 2 mode control register (T2MOD) and Timer/Counter 2 control register (T2CON) control Timer 2.

Table 5.1 describes the external signals referred to in this chapter.

Table 5.1 External Signals

Mnemonic	Type	Description	Multiplexed With
INT0#	I	**External Interrupt 0** This input sets the IE0 interrupt flag in TCON register. IT0 selects the triggering method: IT0= 1 selects edge–triggered (high–to–low); IT0= 0 selects level–triggered (active low). INT0# also serves as external run control for Timer 0, when selected by GATE0 bit in TCON register.	P3.2
INT1#	I	**External Interrupt 1** This input sets the IE1 interrupt flag in TCON register. IT1 selects the triggering method: IT1= 1 selects edge–triggered (high–to–low); IT1= 0 selects level–triggered (active low). INT1# also serves as external run control for Timer 1, when selected by GATE1 bit in TCON register.	P3.3
T0	I	**Timer 0 External Clock Input** When Timer 0 operates as a Counter, a falling edge on the T0 pin increments the count.	P3.4
T1	I	**Timer 1 External Clock Input** When Timer 1 operates as a Counter, a falling edge on the T1 pin increments the count.	P3.5
T2	I/O	**Timer 2 Clock Input/Output** This signal is the external clock input for the Timer 2 capture mode and it is the Timer 2 clock output for the clock–out mode.	P1.0
T2EX	I	**Timer 2 External Input** In Timer 2 capture mode, a falling edge initiates a capture of the Timer 2 registers. In Auto–reload mode, a falling edge causes the Timer 2 to be reloaded. In the up–down Counter mode, this signal determines the count direction: high=up, low=down.	P1.1

The various operating modes of each Timer/Counter are described below

5.2. Timer/Counter Operations

For instance, a basic operation is Timer registers THx and TLx (x= 0, 1 or 2) connected in cascade to form a 16–bit Timer. Setting the run control bit (TRx) in TCON or T2CON register (see Figure 5.9 or Figure 5.15) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the Timer overflow flag (TFx) in TCON or T2CON register. Setting the TRx does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the Timer/Counter is unpredictable.

The C/Tx# control bit selects Timer operation or Counter operation by selecting the divided–down system clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the Timer/Counter is unpredictable.
For Timer operation (C/Tx#= 0), the Timer register counts the divided–down system clock. The Timer register is incremented once every peripheral cycle, i.e. once every six states. Since six states equals 12 oscillator periods (clock cycles), the Timer clock rate is F_{OSC} /12.
Exceptions are the Timer 2 Baud Rate and Clock–Out modes, where the Timer register is incremented by the system clock divided by two.
For Counter operation (C/Tx#= 1), the Timer register counts the negative transitions on the Tx external input pin. The external input is sampled during every S5P2 state. Programmer's Guide describes the notation for the states in a peripheral cycle. When the sample is high in one cycle and low in the next one, the Counter is incremented. The new count value appears in the register during the next S3P1 state after the transition was detected. Since it takes 12 states (24 oscillator periods) to recognize a negative transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle.

5.3. Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 5.1 to Figure 5.4 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (see Figure 5.10) and bits 0, 1, 4 and 5 of TCON register (see Figure 5.9). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) and mode of operation (M10 and M00). TCON register provides Timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).
For normal Timer operation (GATE0= 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control Timer operation.
Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag generating an interrupt request.
It is important to stop Timer/Counter before changing mode.

5.3.1. Mode 0 (13–bit Timer)

Mode 0 configures Timer 0 as an 13–bit Timer which is set up as an 8–bit Timer (TH0 register) with a modulo 32 prescaler implemented with the lower five bits of TL0 register (see Figure 5.1). The upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments TH0 register.

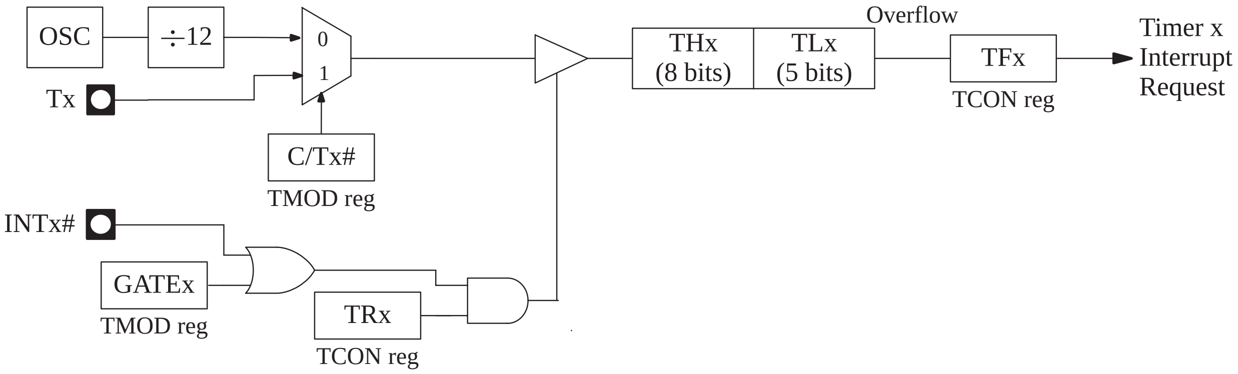


Figure 5.1 Timer/Counter x (x= 0 or 1) in Mode 0

5.3.2. Mode 1 (16–bit Timer)

Mode 1 configures Timer 0 as a 16–bit Timer with TH0 and TL0 registers connected in cascade (see Figure 5.2). The selected input increments TL0 register.

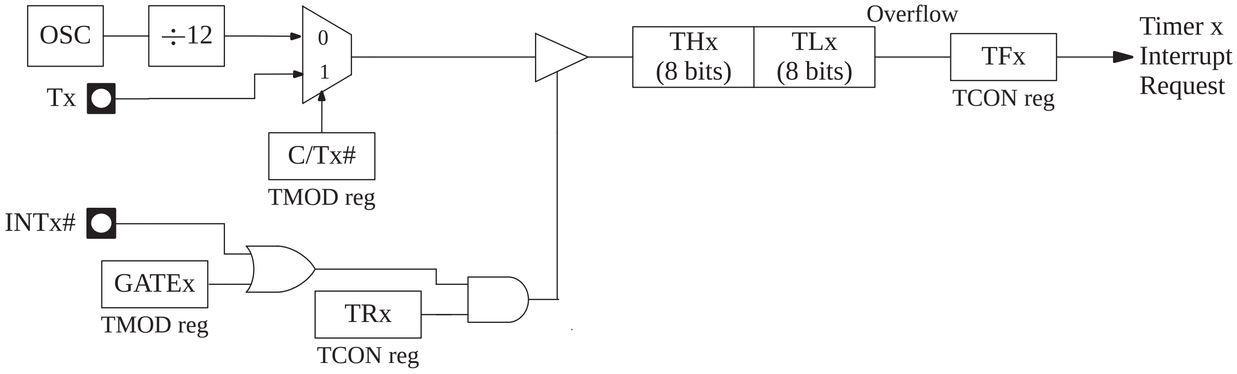


Figure 5.2 Timer/Counter x (x= 0 or 1) in Mode 1

5.3.3. Mode 2 (8–bit Timer with Auto–Reload)

Mode 2 configures Timer 0 as an 8–bit Timer (TL0 register) that automatically reloads from TH0 register (see Figure 5.3). TL0 overflow sets TF0 flag in TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to TH0 register.

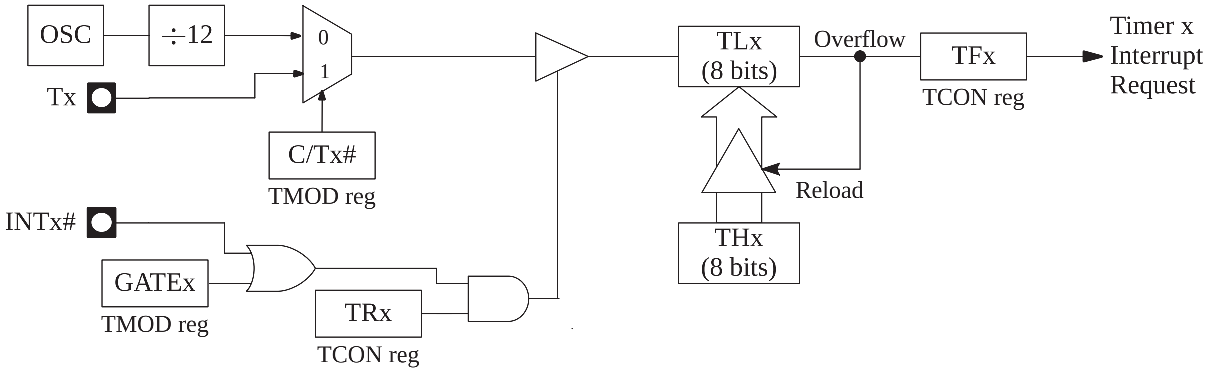


Figure 5.3 Timer/Counter x (x= 0 or 1) in Mode 2

5.3.4. Mode 3 (Two 8–bit Timers)

Mode 3 configures Timer 0 such that registers TL0 and TH0 operate as separate 8–bit Timers (see Figure 5.4). This mode is provided for applications requiring an additional 8–bit Timer or Counter. TL0 uses the Timer 0 control bits C/T0# and GATE0 in TMOD register, and TR0 and TF0 in TCON register in the normal manner. TH0 is locked into a Timer function (counting F_{OSC} /12) and takes over use of the Timer 1 interrupt (TF1) and run control (TR1) bits. Thus, operation of Timer 1 is restricted when Timer 0 is in mode 3.

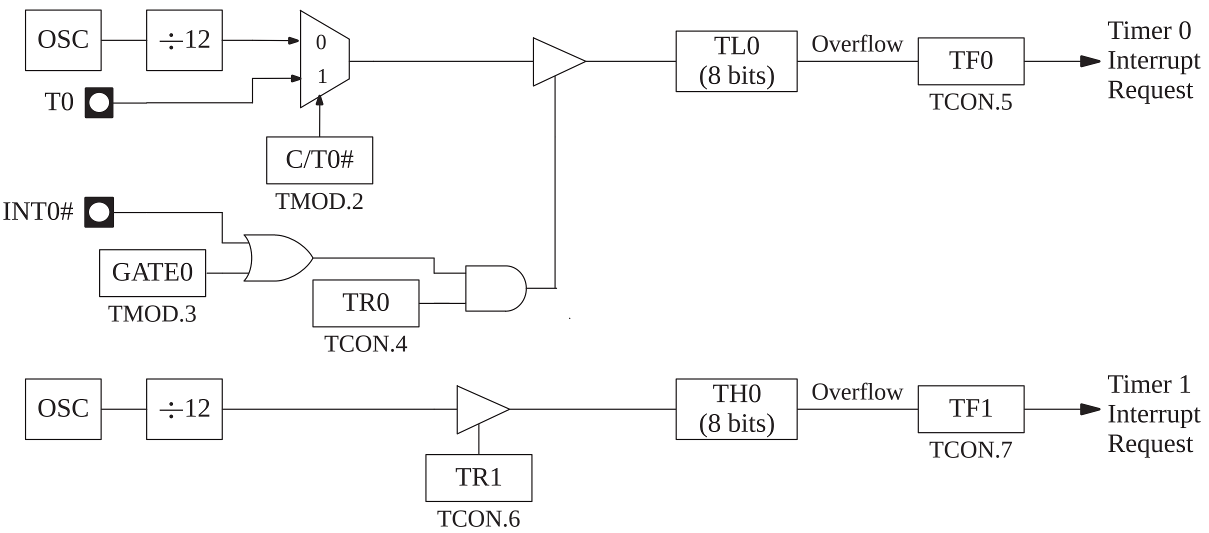


Figure 5.4 Timer/Counter 0 in Mode 3: Two 8-bit Counters

5.4. Timer 1

Timer 1 is identical to Timer 0 excepted for Mode 3 which is a hold–count mode. Following comments help to understand the differences:

- Timer 1 functions as either a Timer or event Counter in three modes of operation. Figure 5.1 to Figure 5.3 show the logical configuration for modes 0, 1, and 2. Timer 1's mode 3 is a hold–count mode.
- Timer 1 is controlled by the four high–order bits of TMOD register (see Figure 5.10) and bits 2, 3, 6 and 7 of TCON register (see Figure 5.9). TMOD register selects the method of Timer gating (GATE1), Timer or Counter operation (C/T1#) and mode of operation (M11 and M01). TCON register provides Timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and interrupt type control bit (IT1).
- Timer 1 can serve as the Baud Rate Generator for the Serial Port. Mode 2 is best suited for this purpose.
- For normal Timer operation (GATE1= 0), setting TR1 allows TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin INT1# to control Timer operation.
- Timer 1 overflow (count rolls over from all 1s to all 0s) sets the TF1 flag generating an interrupt request.
- When Timer 0 is in mode 3, it uses Timer 1's overflow flag (TF1) and run control bit (TR1). For this situation, use Timer 1 only for applications that do not require an interrupt (such as a Baud Rate Generator for the Serial Port) and switch Timer 1 in and out of mode 3 to turn it off and on.
- It is important to stop Timer/Counter before changing mode.

5.4.1. Mode 0 (13–bit Timer)

Mode 0 configures Timer 1 as a 13–bit Timer, which is set up as an 8–bit Timer (TH1 register) with a modulo–32 prescaler implemented with the lower 5 bits of the TL1 register (see Figure 5.1). The upper 3 bits of TL1 register are ignored. Prescaler overflow increments TH1 register.

5.4.2. Mode 1 (16–bit Timer)

Mode 1 configures Timer 1 as a 16–bit Timer with TH1 and TL1 registers connected in cascade (see Figure 5.2). The selected input increments TL1 register.

5.4.3. Mode 2 (8–bit Timer with Auto–Reload)

Mode 2 configures Timer 1 as an 8–bit Timer (TL1 register) with automatic reload from TH1 register on overflow (see Figure 5.3). TL1 overflow sets TF1 flag in TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

5.4.4. Mode 3 (Halt)

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e. when Timer 0 is in mode 3.

5.5. Timer 2

Timer 2 is a 16–bit Timer/Counter. The count is maintained by two eight–bit Timer registers, TH2 and TL2, connected in cascade. Timer 2 is controlled by T2MOD register (see Figure 5.16) and T2CON register (see Figure 5.15).
Timer 2 provides the following operating modes: capture mode, auto–reload mode, Baud Rate Generator mode, and programmable clock–out mode. Select the operating mode with T2MOD and T2CON register bits as shown in Table 5.2 Auto–reload is the default mode. Setting RCLK and/or TCLK selects the Baud Rate Generator mode.

Timer 2 operation is similar to Timer 0 and Timer 1. C/T2# selects F_{OSC}/12 (Timer operation) or external pin T2 (Counter operation) as the Timer register input. Setting TR2 allows TL2 to be incremented by the selected input.

The operating modes are described in the following paragraphs. Block diagrams in Figure 5.5 to Figure 5.8 show the Timer 2 configuration for each mode.
It is important to stop Timer/Counter before changing mode.

Table 5.2 Timer 2 Modes of Operation

Mode	RCLK (in T2CON)	TCLK (in T2CON)	T2OE (in T2MOD)	CP/RL2# (in T2CON)
Auto-reload	0	0	0	0
Capture	0	0	0	1
Programmable Clock–Out	0	0	1	0
Reserved	0	0	1	1
Baud Rate Generator	0	1	0	X
BRG and Clock–Out	0	1	1	X
Baud Rate Generator	1	X	0	X
BRG and Clock–Out	1	X	1	X

5.5.1. Auto–Reload Mode

The auto–reload mode configures Timer 2 as a 16–bit Timer or event Counter with automatic reload. The Timer operates an as an up Counter or as an up/down Counter, as determined by the down Counter enable bit DCEN in T2MOD register. At device reset, DCEN is cleared, so in the auto–reload mode, Timer 2 defaults to operation as an up Counter.

5.5.1.1. Up Counter Operation

When DCEN= 0, Timer 2 operates as an up Counter (see Figure 5.5). The external enable bit EXEN2 in T2CON register provides two options. If EXEN2= 0, Timer 2 counts up to FFFFh and sets TF2 overflow flag. The overflow condition loads the 16–bit value of the reload/capture registers (RCAP2H, RCAP2L) into the Timer registers (TH2, TL2). The values in RCAP2H and RCAP2L are preset by software. In this case, T2EX is not used.
If EXEN2= 1, the Timer registers are reloaded by either a Timer overflow or a high–to–low transition at external input T2EX. This transition also sets EXF2 bit in T2CON register. Either TF2 or EXF2 bit can generate an interrupt request.

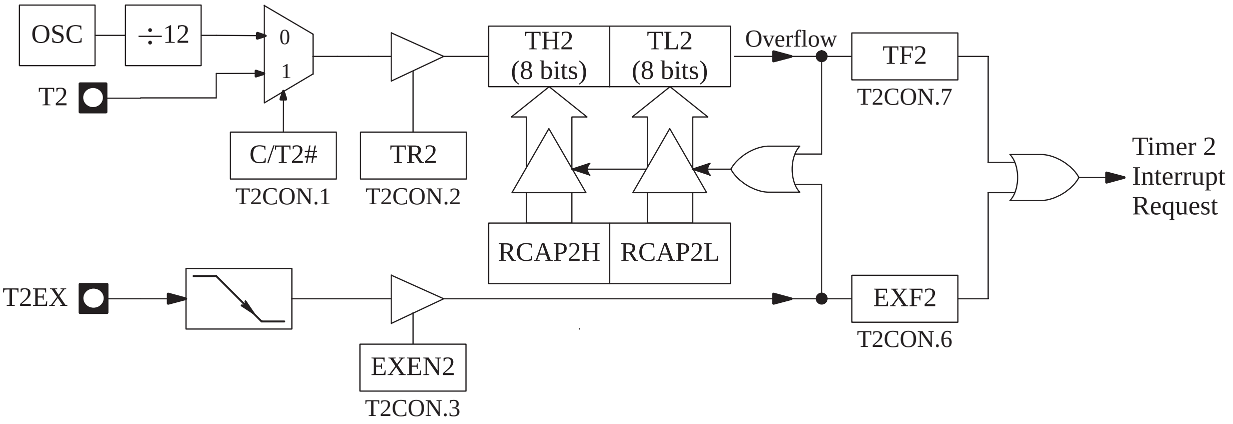


Figure 5.5 Timer 2: Auto Reload Mode Up Counter (DCEN= 0)

5.5.1.2. Up/Down Counter Operation

When DCEN= 1, Timer 2 operates as an up/down Counter (see Figure 5.6). External pin T2EX controls the direction of the count. When T2EX is high, Timer 2 counts up. The Timer overflow occurs at FFFFh which sets the TF2 overflow flag and generates an interrupt request. The overflow also causes the 16–bit value in RCAP2H and RCAP2L to be loaded into the Timer registers (TH2, TL2).

When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the Timer registers (TH2, TL2) equals the value stored in RCAP2H and RCAP2L. The underflow sets TF2 bit and reloads FFFFh into the Timer registers.

The EXF2 bit toggles when Timer 2 overflows or underflows according to the direction of the count. When Timer 2 operates as an up/down Counter, EXF2 does not generate an interrupt. This bit can be used to provide 17–bit resolution.

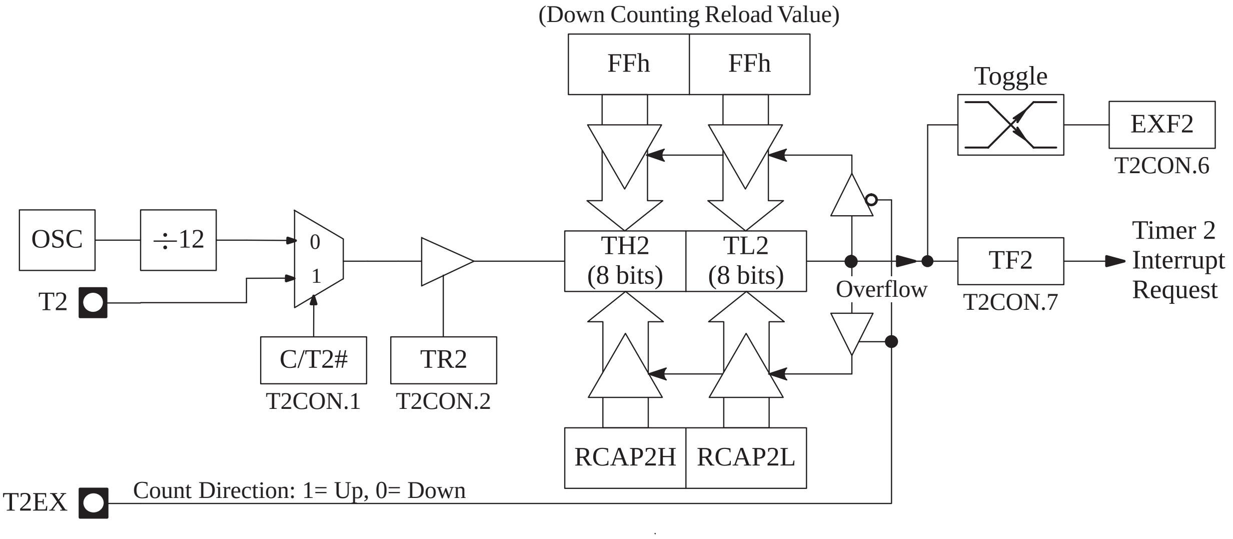


Figure 5.6 Timer 2: Auto Reload Mode Up/Down Counter (DCEN= 1)

5.5.2. Capture Mode

In the capture mode, Timer 2 functions as a 16–bit Timer or Counter (see Figure 5.7). An overflow condition sets TF2 bit, which you can use to request an interrupt. Setting the external enable bit EXEN2 allows RCAP2H and RCAP2L registers to capture the current value in Timer registers TH2 and TL2 in response to a 1–to–0 transition at external input T2EX. The transition at T2EX also sets bit EXF2 in T2CON register. EXF2 bit, like TF2, can generate an interrupt.

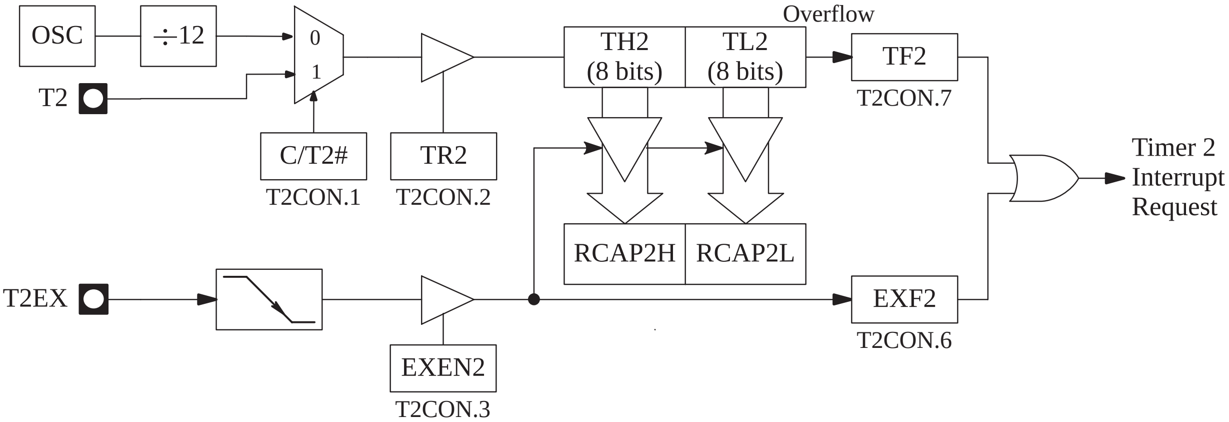


Figure 5.7 Timer 2: Capture Mode

5.5.3. Baud Rate Generator Mode

This mode configures Timer 2 as a Baud Rate Generator for use with the Serial Port. Select this mode by setting the RCLK and/or TCLK bits in T2CON register. see paragraph 6.7.3.3. for more information on this mode.

5.5.4. Clock–Out Mode

In the clock–out mode, Timer 2 operates as a 50%–duty–cycle, programmable clock generator (see Figure 5.8). The input clock increments TL2 at frequency $F_{OSC}/2$. The Timer repeatedly counts to overflow from a preloaded value. At overflow, the contents of RCAP2H and RCAP2L registers are loaded into TH2 and TL2. In this mode, Timer 2 overflows do not generate interrupts. The formula gives the clock–out frequency as a function of the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$\text{Clock\_Out Frequency} = \frac{F_{OSC}}{4 \times (65536 - \text{RCAP2H}/\text{RCAP2L})}$$

For a 16 MHz system clock, Timer 2 has a programmable frequency range of 61 Hz to 4 MHz. The generated clock signal is brought out to T2 pin.

Timer 2 is programmed for the clock–out mode as follows:

- Set T2OE bit in T2MOD. This gates the Timer register overflow to the ÷2 Counter.
- Clear C/T2# bit in T2CON register to select $F_{OSC}/2$ as the Timer input signal. This also enables the clock output (T2 pin).
- Determine the 16–bit reload value from the formula and enter it in the RCAP2H/RCAP2L registers.
- Enter a 16–bit initial value in Timer register TH2/TL2. It can be the same as the reload value or a different one depending on the application.
- To start the Timer, set TR2 run control bit in T2CON register.

Operation is similar to Timer 2 operation as a Baud Rate Generator. It is possible to use Timer 2 as a Baud Rate Generator and a clock generator simultaneously. For this configuration, the baud rates and clock frequencies are not independent since both functions use the values in the RCAP2H and RCAP2L registers.

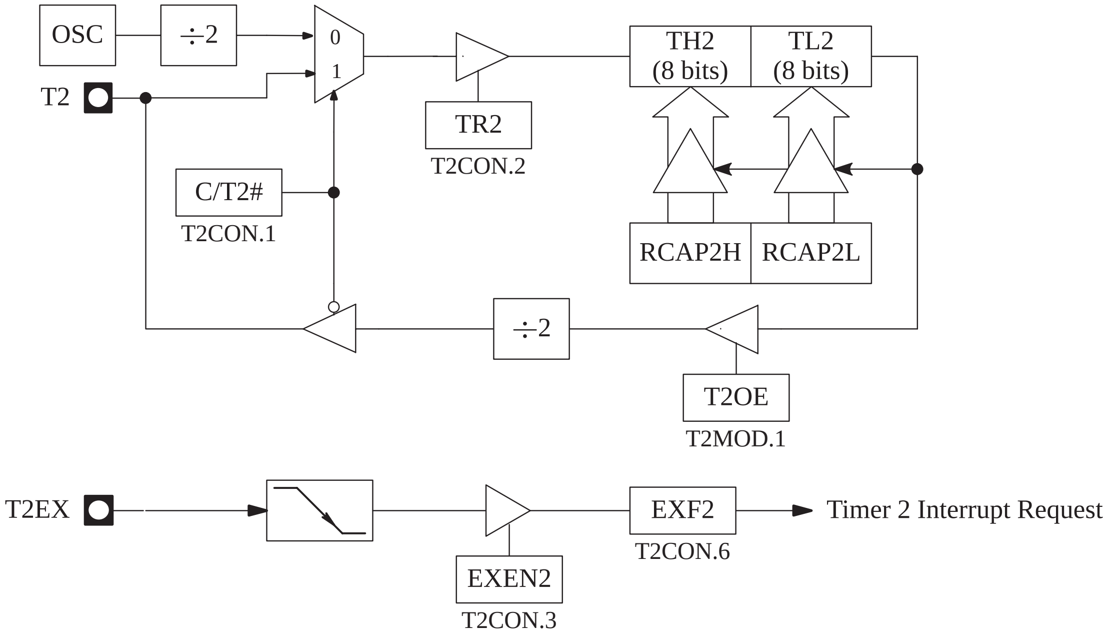


Figure 5.8 Timer 2: Clock Out Mode

5.6. Registers

TCON (S:88h)
Timer/Counter Control Register

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

Bit Number	Bit Mnemonic	Description
7	TF1	**Timer 1 Overflow flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 1 register overflows.
6	TR1	**Timer 1 Run Control bit** Clear to turn off Timer/Counter 1. Set to turn on Timer/Counter 1.
5	TF0	**Timer 0 Overflow flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 0 register overflows.
4	TR0	**Timer 0 Run Control bit** Clear to turn off Timer/Counter 0. Set to turn on Timer/Counter 0.
3	IE1	**Interrupt 1 Edge flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	**Interrupt 1 Type Control bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

Reset Value= 0000 0000b

Figure 5.9 TCON Register

TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	**C/T1#**	**M11**	**M01**	**GATE0**	**C/T0#**	**M10**	**M00**

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select bit** Clear for Timer operation: Timer 1 counts the divided–down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select bits** M11 M01 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH1) with 5–bit prescaler (TL1). 0 1 Mode 1: 16–bit Timer/Counter. 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL1). Reloaded from TH1 at overflow. 1 1 Mode 3: Timer 1 halted. Retains count.
4	M01	
3	GATE0	**Timer 0 Gating Control bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select bit** Clear for Timer operation: Timer 0 counts the divided–down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select bit** M10 M00 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH0) with 5–bit prescaler (TL0). 0 1 Mode 1: 16–bit Timer/Counter. 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3: TL0 is an 8–bit Timer/Counter. TH0 is an 8–bit Timer using Timer 1’s TR0 and TF0 bits.
0	M00	

Reset Value= 0000 0000b

Figure 5.10 TMOD Register

TH0 (S:8Ch)

Timer 0 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 0.**

Reset Value= 0000 0000b

Figure 5.11 TH0 Register

TL0 (S:8Ah)

Timer 0 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 0.**

Reset Value= 0000 0000b

Figure 5.12 TL0 Register

TH1 (S:8Dh)

Timer 1 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 1.**

Reset Value= 0000 0000b

Figure 5.13 TH1 Register

TL1 (S:8Bh)

Timer 1 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1.**

Reset Value= 0000 0000b

Figure 5.14 TL1 Register

T2CON (S:C8h)
Timer/Counter 2 Control Register

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 Overflow flag** TF2 is not set if RCLK= 1 or TCLK= 1. Set by hardware when Timer 2 overflows. Must be cleared by software
6	EXF2	**Timer 2 External flag** EXF2 does not cause an interrupt in up/down counter mode (DCEN= 1). Set by hardware if EXEN2= 1 when a negative transition on T2EX pin is detected.
5	RCLK	**Receive Clock bit** Clear to select Timer 1 as the Timer Receive Baud Rate Generator for the Serial Port in modes 1 and 3. Set to select Timer 2 as the Timer Receive Baud Rate Generator for the Serial Port in modes 1 and 3.
4	TCLK	**Transmit Clock bit** Clear to select Timer 1 as the Timer Transmit Baud Rate Generator for the Serial Port in modes 1 and 3. Set to select Timer 2 as the Timer Transmit Baud Rate Generator for the Serial Port in modes 1 and 3.
3	EXEN2	**Timer 2 External Enable bit** Clear to ignore events on T2EX pin for Timer 2. Set to cause a capture or reload when a negative transition on T2EX pin is detected unless Timer 2 is being used as the Baud Rate Generator for the Serial Port.
2	TR2	**Timer 2 Run Control bit** Clear to turn off Timer 2. Set to to turn on Timer 2.
1	C/T2#	**Timer 2 Counter/Timer Select bit** Clear for Timer operation: Timer 2 counts the divided–down system clock. Set for Counter operation: Timer 2 counts negative transitions on external pin T2.
0	CP/RL2#	**Capture/Reload bit** CP/RL2# is ignored and Timer 2 is forced to auto–reload on Timer 2 overflow if RCLK= 1 or TCLK= 1. Clear to auto–reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2= 1. Set to capture on negative transitions on T2EX pin if EXEN2= 1

Reset Value= 0000 0000b

Figure 5.15 T2CON Register

T2MOD (S:C9h)
Timer/Counter 2 Mode Control Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T2OE	DCEN

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	T2OE	**Timer 2 Output Enable bit** Clear to disable the programmable clock output to external pin T2 in the Timer 2 clock–out mode. Set to enable the programmable clock output to external pin T2 in the Timer 2 clock–out mode.
0	DCEN	**Down Count Enable bit** Clear to configure Timer 2 as an up Counter. Set to configure Timer 2 as an up/down Counter.

Reset Value= XXXX XX00b

Figure 5.16 T2MOD Register

TH2 (S:CDh)

Timer 2 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 2.**

Reset Value= 0000 0000b

Figure 5.17 TH2 Register

TL2 (S:CCh)

Timer 2 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 2.**

Reset Value= 0000 0000b

Figure 5.18 TL2 Register

RCAP2H (S:CBh)

Timer 2 Reload/Capture High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 2 Reload/Capture.**

Reset Value= 0000 0000b

Figure 5.19 RCAP2H Register

RCAP2L (S:CAh)

Timer 2 Reload/Capture Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 2 Reload/Capture.**

Reset Value= 0000 0000b

Figure 5.20 RCAP2L Register

6. Serial I/O Port

6.1. Introduction

This chapter provides instructions on programming the Serial Port and generating the Serial I/0 Baud Rates with Timer 1, Timer 2 and the internal Baud Rate Generator. The Serial Input/Output Port supports communication with modems and other external peripheral devices.

The Serial Port provides both synchronous and asynchronous communication modes. It operates as a Universal Asynchronous Receiver and Transmitter (UART) in three full–duplex modes (Modes 1, 2 and 3). Asynchronous transmission and reception can occur simultaneously and at different Baud Rates. The UART supports framing–bit error detection, overrun error detection, multiprocessor communication, and automatic address recognition. The Serial Port also operates in a single synchronous mode (Mode 0).

The synchronous mode (Mode 0) operates either at a single Baud Rate (80C51 compatibility) or at a variable Baud Rate with an independent and internal Baud Rate Generator. Mode 2 can operate at two Baud Rates. Modes 1 and 3 operate over a wide range of Baud Rates, which are generated by Timer 1, Timer 2 and internal Baud Rate Generator.

The Serial Port signals are defined in Table 6.1. Figure 6.1 shows the Serial Port block diagram.

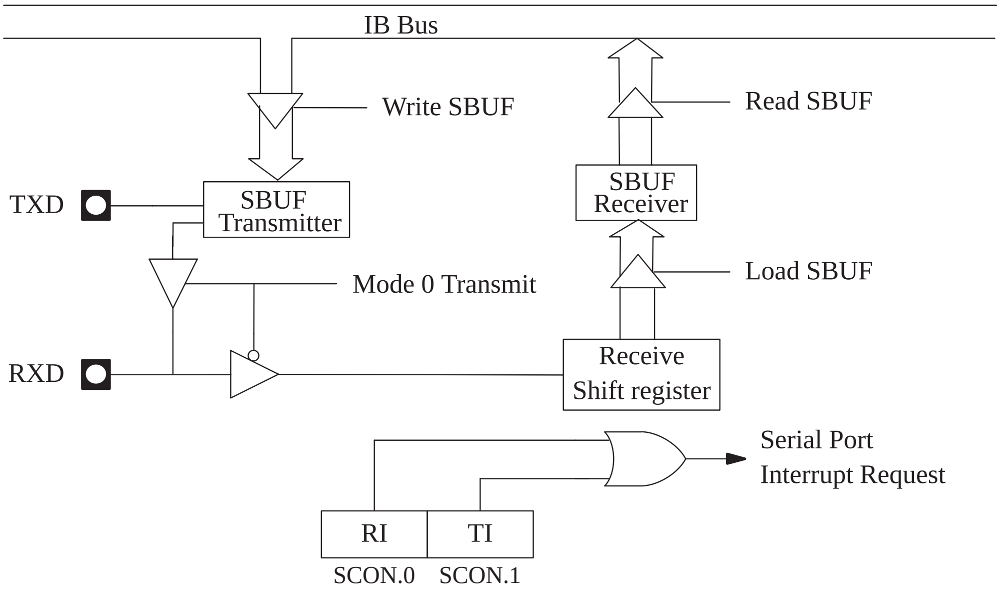


Figure 6.1 Serial Port Block Diagram

Table 6.1 Serial Port Signals

Name	Type	Description	Multiplexed with
TXD	O	**Transmit Data** In mode 0, TXD transmits the clock signal. In modes 1, 2 and 3, TXD transmits serial data.	P3.1
RXD	I/O	**Receive Data** In mode 0, RXD transmits and receives serial data. In mode 1,2 and 3, RXD receives serial data.	P3.0

For the three asynchronous modes, the UART transmits on the TXD pin and receives on the RXD pin. For the synchronous mode (Mode 0), the UART outputs a clock signal enabled by BRR bit into BDRCON (see Figure 6.11) on the TXD pin and sends and receives messages on the RXD pin (see Figure 6.1). The Baud Rate is selected using SRC, SPD bits value into BDRCON register. SBUF register, which holds received bytes and bytes to be transmitted, actually consists of two physically different registers. To send, software writes a byte to SBUF; to receive, software reads SBUF. The receive shift register allows reception of a second byte before the first byte has been read from SBUF. The UART sets interrupt bits TI and RI on transmission and reception, respectively. These two bits share a single interrupt request and interrupt vector. When RI is set, new received byte cannot overwrite the byte already stored into SBUF.

It is important to stop exchange or wait until exchange completion before changing Baud Rate whatever the modes used.

6.2. Modes of Operation

The Serial Port can operate in one synchronous and three asynchronous modes.

6.2.1. Synchronous Mode (Mode 0)

Mode 0 is a half–duplex, synchronous mode, which is commonly used to expand the I/0 capabilities of a device with shift registers. The transmit data (TXD) pin outputs a set of eight clock pulses while the receive data (RXD) pin transmits or receives a byte of data. The 8–bit data are transmitted and received least–significant bit (LSB) first. Shifts occur in the last phase (S6P2) of every peripheral cycle, which corresponds to a Baud Rate of F_{OSC}/12. Figure 6.2 shows the timing for transmission and reception in Mode 0.

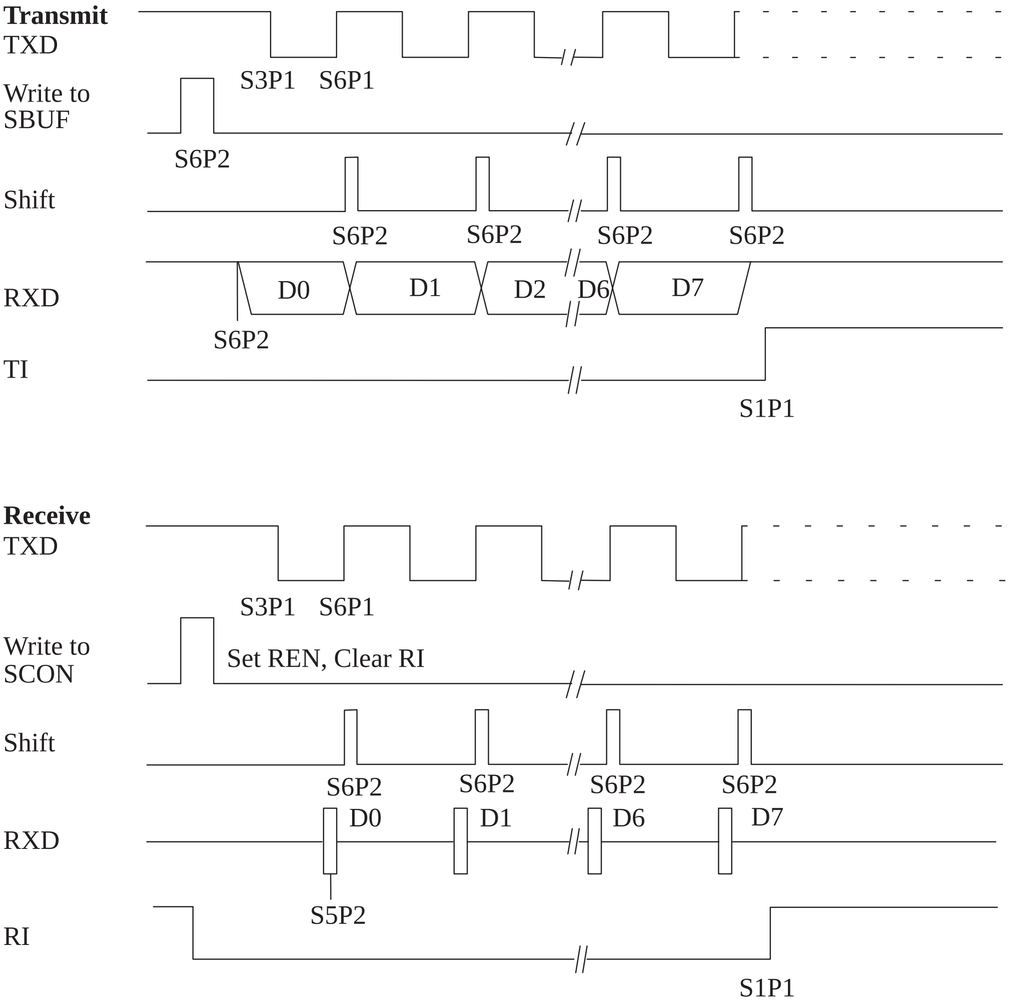


Figure 6.2 Mode 0 Timings

6.2.1.1. Transmission (Mode 0)

Follow these steps to begin a transmission:

- Write to SCON register clearing bits SM0, SM1.
- Write the byte to be transmitted to the SBUF register. This write starts the transmission.

Hardware executes the write to SBUF in the last phase (S6P2) of a peripheral cycle. At S6P2 of the following cycle, hardware shifts the LSB (D0) onto the RXD pin. At S3P1 of the next cycle, the TXD pin goes low for the first clock–signal pulse. Shifts continue every peripheral cycle. In the ninth cycle after the write to SBUF, the MSB (D7) is on the RXD pin. At the beginning of the 10th cycle, hardware drives the RXD pin high and asserts TI to indicate the end of the transmission.

6.2.1.2. Reception (Mode 0)

To start a reception in mode 0, write to the SCON register. Clear SM0, SM1 and RI bits and set the REN bit.
Hardware executes the write to SCON in the last phase (S6P2) of a peripheral cycle (see Figure 6.2). In the second peripheral cycle, the LSB (D0) is sampled on the RXD pin at S5P2. The D0 bit is then shifted into the shift register. After eight shifts at S6P2 of every peripheral cycle, the MSB (D7) is shifted into the shift register, and hardware asserts RI bit to indicate a completed reception. Software can then read the received byte from SBUF register.

6.2.2. Asynchronous Modes (Modes 1, 2 and 3)

The Serial Port has three asynchronous modes of operation:

- Mode 1
  Mode 1 is a full–duplex, asynchronous mode with independant Transmit and Receive Baud Rate selection. The data frame (see Figure 6.3) consists of 10 bits: one start, eight data bits and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. When a message is received, the stop bit is read in the RB8 bit in SCON register. The Baud Rate is generated either by overflow of Timer 1 or by overflow of Timer 2 or by overflow of the internal Baud Rate Generator (see paragraph 6.7.3.4. “Internal Baud Rate Generator”).
- Modes 2 and 3
  Modes 2 and 3 are full–duplex, asynchronous modes. Only Mode 3 allows independant Transmit and Receive Baud Rate selection.The data frame (see Figure 6.3) consists of 11–bit: one start bit, 8–bit data (transmitted and received LSB first), one programmable ninth data bit and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. On receive, the ninth bit is read from RB8 bit in SCON register. On transmit, the ninth data bit is written to TB8 bit in SCON register. (Alternatively, you can use the ninth bit as a command/data flag.)
  - In mode 2, the Baud Rate is programmable to 1/32 or 1/64 of the oscillator frequency following SMOD1 bit value located into PCON register.
  - In mode 3, the Baud Rate is generated either by overflow of Timer 1 or by overflow of Timer 2, or by overflow of internal Baud Rate Generator.

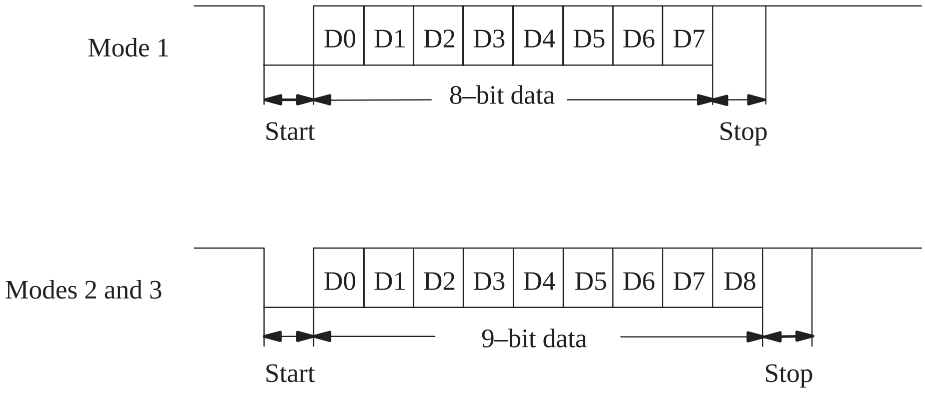


Figure 6.3 Data Frames (Modes 1, 2 and 3)

6.2.2.1. Transmission (Modes 1, 2 and 3)

Follow these steps to initiate a transmission:

- Write to SCON register. Select the mode with SM0 and SM1 bits. For modes 2 and 3, also write the ninth bit to TB8 bit.
- Write the byte to be transmitted to SBUF register. This write starts the transmission.

6.2.2.2. Reception (Modes 1, 2 and 3)

To prepare for a reception, set REN bit in SCON register. The actual reception is then initiated by a detected high–to–low transition on the RXD pin.

6.3. Framing Bit Error Detection (Modes 1, 2 and 3)

Framing bit error detection is provided for the three asynchronous modes. To enable the framing bit error detection feature, set SMOD0 bit in PCON register. When this feature is enabled, the receiver checks each incoming data frame for a valid stop bit. An invalid stop bit may result from noise on the serial lines or from simultaneous transmission by two CPUs. If a valid stop bit is not found, the software sets FE bit in SCON register.
Software may examine FE bit after each reception to check for data errors. Once set, only software or a reset clear FE bit. Subsequently received frames with valid stop bits cannot clear FE bit. When FE feature is enabled, RI rises on stop bit instead of the last data bit.

6.4. Overrun Error Detection (Modes 1, 2 and 3)

Overrun error detection is provided for the three asynchronous modes. To enable the overrun error detection feature, set SMOD0 bit in PCON register.

This error occurs when a first data received is not read by the CPU before a second one is completely received. In this case OVR flag is set and the second data is lost. Figure 6.4 shows an example of Overrun Error.

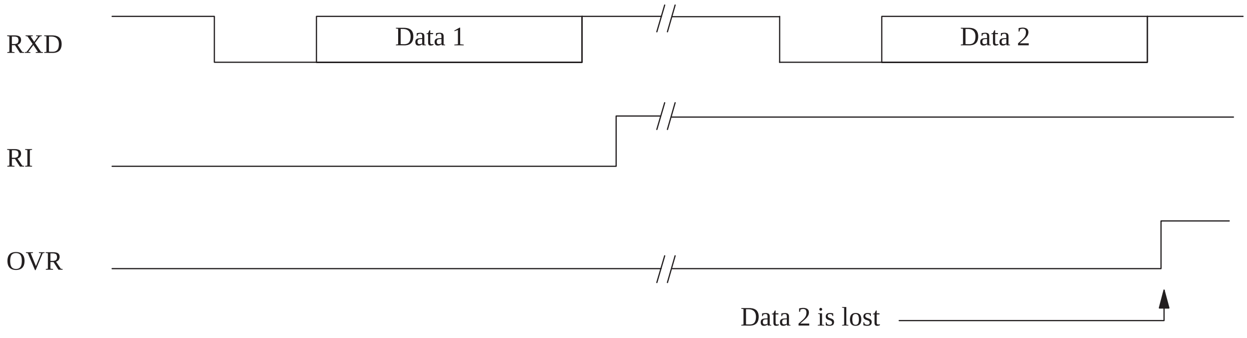


Figure 6.4 Overrun Error (Modes 1, 2 and 3)

In this example Data 1 is received and RI is set. Then Data 2 is received before the CPU has read the first one.

6.5. Multiprocessor Communication (Modes 2 and 3)

Modes 2 and 3 provide a ninth–bit mode to facilitate multiprocessor communication. To enable this feature, set SM2 bit in SCON register. When the multiprocessor communication feature is enabled, the Serial Port can differentiate between data frames (ninth bit clear) and address frames (ninth bit set). This allows the TSC80251G1D to function as a slave processor in an environment where multiple slave processors share a single serial line.
When the multiprocessor communication feature is enabled, the receiver ignores frames with the ninth bit clear. The receiver examines frames with the ninth bit set for an address match. If the received address matches the slaves address,

the receiver hardware sets RB8 and RI bits in SCON register, generating an interrupt.
The addressed slave's software then clears SM2 bit in SCON register and prepares to receive the data bytes. The other slaves are unaffected by these data bytes because they are waiting to respond to their own addresses.

Note:
ES bit must be set in IE register to allow RI bit to generate an interrupt.

6.6. Automatic Address Recognition

The automatic address recognition feature is enabled when the multiprocessor communication feature is enabled (SM2 bit in SCON register is set).
Implemented in hardware, automatic address recognition enhances the multiprocessor communication feature by allowing the Serial Port to examine the address of each incoming command frame. Only when the Serial Port recognizes its own address, the receiver sets RI bit in SCON register to generate an interrupt. This ensures that the CPU is not interrupted by command frames addressed to other devices.
If desired, you may enable the automatic address recognition feature in mode 1. In this configuration, the stop bit takes the place of the ninth data bit. Bit RI is set only when the received command frame address matches the device's address and is terminated by a valid stop bit.
To support automatic address recognition, a device is identified by a given address and a broadcast address.

Note:
The multiprocessor communication and automatic address recognition features cannot be enabled in mode 0 (i.e, setting SM2 bit in SCON register in mode 0 has no effect).

6.6.1. Given Address

Each device has an individual address that is specified in SADDR register; the SADEN register is a mask byte that contains don't–care bits (defined by zeros) to form the device's given address. The don't–care bits provide the flexibility to address one or more slaves at a time. The following example illustrates how a given address is formed.
To address a device by its individual address, the SADEN mask byte must be `1111 1111B`.
For example:

SADDR	=	`0101 0110b`
SADEN	=	`1111 1100b`
Given	=	`0101 01XXb`

The following is an example of how to use given addresses to address different slaves:

Slave A:	SADDR	=	`1111 0001b`
	SADEN	=	`1111 1010b`
	Given	=	`1111 0X0Xb`
Slave B:	SADDR	=	`1111 0011b`
	SADEN	=	`1111 1001b`
	Given	=	`1111 0XX1b`
Slave C:	SADDR	=	`1111 0010b`
	SADEN	=	`1111 1101b`
	Given	=	`1111 00X1b`

The SADEN byte is selected so that each slave may be addressed separately.
For slave A, bit 0 (the LSB) is a don't–care bit; for slaves B and C, bit 0 is a 1. To communicate with slave A only, the master must send an address where bit 0 is clear (e.g. `1111 0000B`).
For slave A, bit 1 is a 0; for slaves B and C, bit 1 is a don't care bit. To communicate with slaves A and B, but not slave C, the master must send an address with bits 0 and 1 both set (e.g. `1111 0011B`).
To communicate with slaves A, B and C, the master must send an address with bit 0 set, bit 1 clear, and bit 2 clear (e.g. `1111 0001B`).

6.6.2. Broadcast Address

A broadcast address is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't–care bits, e.g.:

SADDR	=	0101 0110b
SADEN	=	1111 1100b
(SADDR) or (SADEN)	=	1111 111Xb

The use of don’t–care bits provides flexibility in defining the broadcast address, however in most applications, a broadcast address is FFh.

The following is an example of using broadcast addresses:

Slave A:	SADDR	=	1111 0001b
	SADEN	=	1111 1010b
	Given	=	1111 1X11b,
Slave B:	SADDR	=	1111 0011b
	SADEN	=	1111 1001b
	Given	=	1111 1X11b,
Slave C:	SADDR	=	1111 0010b
	SADEN	=	1111 1101b
	Given	=	1111 1111b,

For slaves A and B, bit 2 is a don’t care bit; for slave C, bit 2 is set. To communicate with all of the slaves, the master must send an address FFh.

To communicate with slaves A and B, but not slave C, the master must send the address FBh.

6.6.3. Reset Addresses

On reset, the SADDR and SADEN registers are initialized to 00h, i.e. the given and broadcast addresses are XXXX XXXXB (all don’t–care bits). This ensures that the Serial Port is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition.

6.7. Baud Rates

The Baud Rate Control register (BDRCON, see Figure 6.11) is added to the TSC80251G1D derivatives in order to manage the new functionality of the UART. Three Baud Rate Generators can supply the transmission clock to the UART: Timer 1, Timer 2 and the internal Baud Rate Generator as detailed below.

6.7.1. Baud Rate for Mode 0

The transmission clock is provided by either the internal Baud Rate Generator or the internal fixed prescaler. This selection is done by setting SRC bit in BDRCON register. The transmission clock selection is shown in Figure 6.5

- When SRC= 0, the Baud Rate is fully compatible with 80C51 microcontrollers: Baud_Rate= $F_{OSC}/12$
- When SRC= 1, the Internal Baud Rate Generator (BRG) is selected and the Baud Rate is variable in two ranges:
  - When SPD= 1, the Fast mode is selected: Baud_Rate= $F_{OSC}/[4\cdot(256–BRL)]$
  - When SPD= 0, the Slow mode is selected: Baud_Rate= $F_{OSC}/[24\cdot(256–BRL)]$.

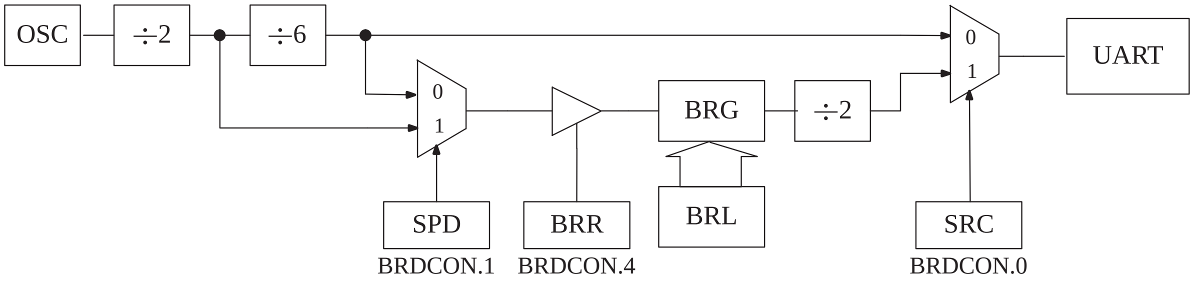


Figure 6.5 Clock Transmission Sources in Mode 0

By default, after a reset, the bit SRC is cleared and the transmission clock is compatible with 80C51 microcontrollers. Setting this bit to one, selects the internal Baud Rate Generator. The 8–bit register BRL is the reload register of the Baud Rate Generator.

6.7.2. Baud Rate for Mode 2

The Baud Rate in mode 2 depends on the value of SMOD1 bit in PCON register. If SMOD1= 0 (default value on reset), the Baud Rate is 1/64 the oscillator frequency. If SMOD1= 1, the Baud Rate is 1/32 the oscillator frequency:

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{64}$$

The Baud Rate configuration for mode 2 is shown in Figure 6.6.

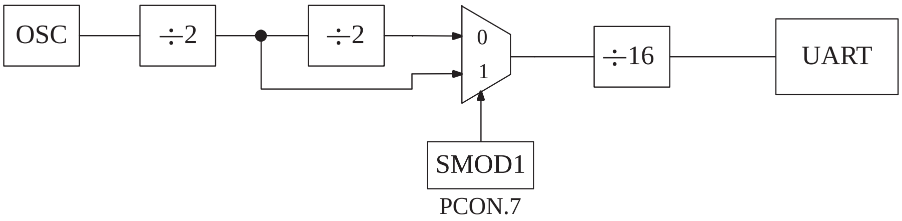


Figure 6.6 UART in Mode 2

6.7.3. Baud Rate for Modes 1 and 3

Three Baud Rate Generators can supply the Baud Rate to the UART: Timer 1, Timer 2 and the internal Baud Rate Generator. It is possible to have different clocks for the transmission and reception.

6.7.3.1. Baud Rate Selection

The Baud Rate Generator for transmit and receive clocks can be selected separately via the BDRCON register (see Figure 6.11).

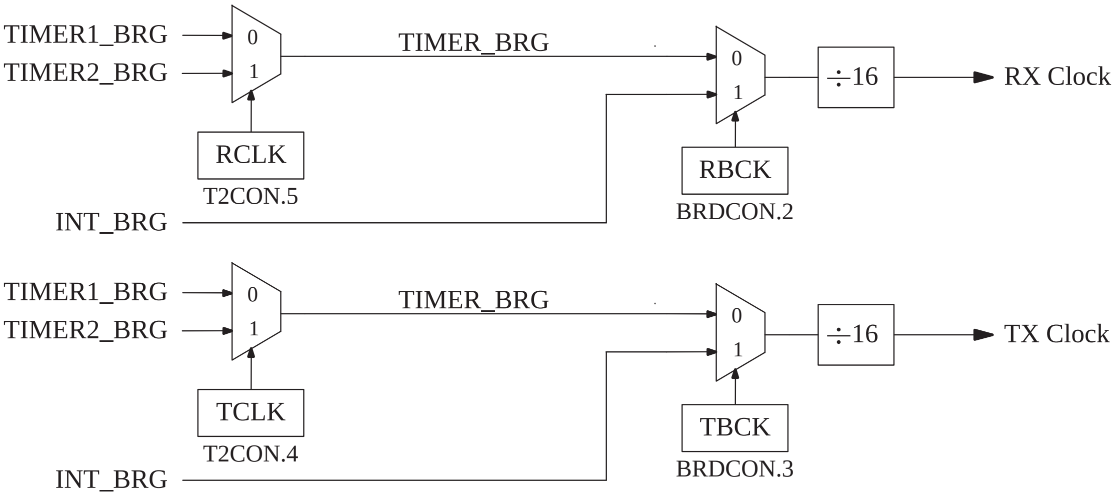


Figure 6.7 Baud Rate Generator Selection

6.7.3.2. Timer 1

When Timer 1 is used as Baud Rate Generator, the Baud Rates in Modes 1 and 3 are determined by the Timer 1 overflow and the value of SMOD1 bit in PCON register:

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{12 \times 32 \times [256 - (\text{TH1})]}$$

$$\text{TH1} = 256 - \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{384 \times \text{Baud\_Rate}}$$

The configuration is shown in Figure 6.8.

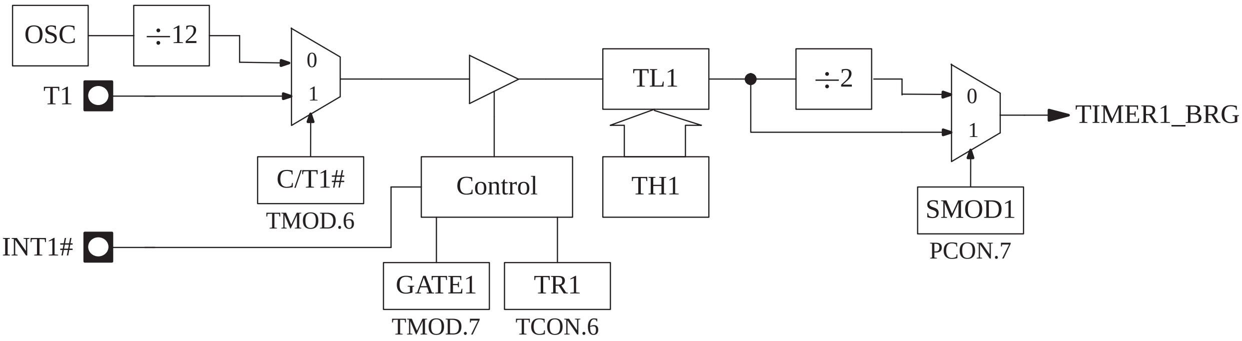


Figure 6.8 Timer 1 as Baud Rate Generator in Modes 1 and 3

Table 6.2 Timer 1 Generated Baud Rates at 12 MHz

Baud Rates	F_{OSC}= 11.0592 MHz			F_{OSC}= 12 MHz		
	SMOD1	TH1	Error (%)	SMOD1	TH1	Error (%)
57600	1	255	0	–	–	–
38400	–	–	–	–	–	–
28800	0	255	0	–	–	–
19200	1	253	0	–	–	–
9600	0	253	0	–	–	–
4800	0	250	0	1	243	0.16
2400	0	244	0	0	243	0.16
1200	0	232	0	0	230	0.16
600	0	208	0	0	204	0.16
300	0	160	0	0	152	0.16

Table 6.3 Timer 1 Generated Baud Rates at 16 MHz

Baud Rates	F_{OSC}= 14.7456 MHz			F_{OSC}= 16 MHz		
	SMOD1	TH1	Error (%)	SMOD1	TH1	Error (%)
57600	–	–	–	–	–	–
38400	0	255	0	–	–	–
28800	–	–	0	–	–	–
19200	0	254	0	–	–	–
9600	0	252	0	1	247	3.55
4800	0	248	0	1	239	2.12
2400	0	240	0	1	221	0.79
1200	0	224	0	1	187	0.64
600	0	192	0	1	117	0.08
300	0	128	0	0	117	0.08

2

Table 6.4 Timer 1 Generated Baud Rates at 24 MHz

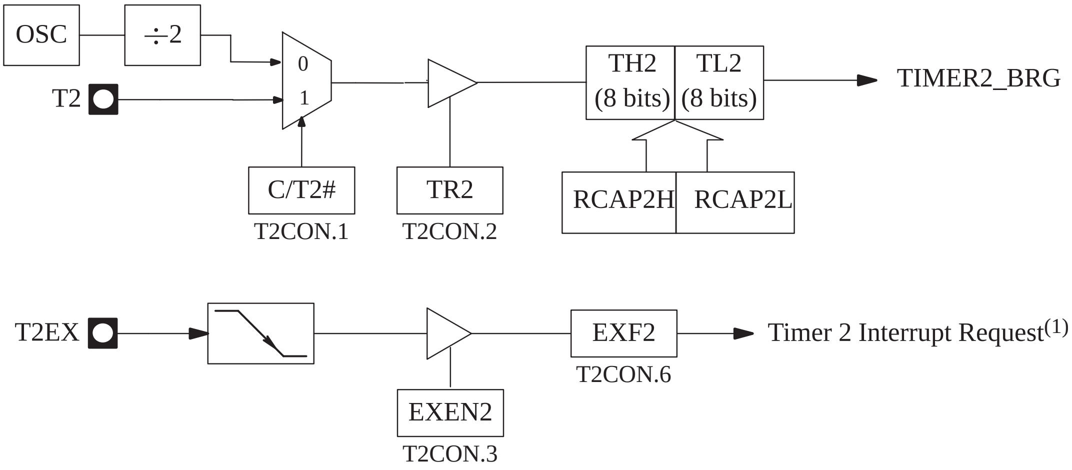
Baud Rates	F_{OSC}= 22.1184MHz			F_{OSC}= 24 MHz		
	SMOD1	TH1	Error (%)	SMOD1	TH1	Error (%)
115200	1	255	0	–	–	–
57600	1	254	0	–	–	–
38400	–	–	–	–	–	–
28800	0	254	0	–	–	–
19200	0	253	0	–	–	–
9600	0	250	0	1	243	0.16
4800	0	244	0	0	243	0.16
2400	0	232	0	0	230	0.16
1200	0	208	0	0	204	0.16
600	0	160	0	0	152	0.16
300	0	64	0	0	48	0.16

6.7.3.3. Timer 2

In this mode, a rollover in TH2 register does not set the TF2 bit in T2CON register. Also, a high–to–low transition at T2EX pin sets the EXF2 bit in T2CON register but does not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). T2EX pin can be used as an additional external interrupt by setting the EXEN2 bit in T2CON.

Note:
Turn the timer off (clear the TR2 bit in T2CON register) before accessing registers. TH2, TL2 RCAP2H and RCAP2L.

You may configure Timer 2 as a timer or a counter. In most applications, it is configured for timer operation (the C/T2# bit is cleared in T2CON register).



Note:

1. When Timer 2 is used in Baud Rate Generator mode, T2EX input provides additional external interrupt input.

Figure 6.9 Timer 2 in Baud Rate Generator Mode

Note that Timer 2 increments every state time ($2T_{OSC}$) when it is in the Baud Rate Generator mode. In the Baud Rate formula that follows, “RCAP2H, RCAP2L” denotes the contents of RCAP2H and RCAP2L taken as a 16–bit unsigned integer:

$$\text{Baud\_Rate} = \frac{F_{OSC}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

$$(\text{RCAP2H, RCAP2L}) = 65536 - \frac{F_{OSC}}{32 \times \text{Baud\_Rate}}$$

Note:

When Timer 2 is configured as a timer and is in Baud Rate Generator mode, do not read or write the TH2 or TL2 registers. The timer is being incremented every state time, and the results of a read or write may not be accurate. In addition, you may read, but not write to RCAP2 registers; a write may overlap a reload and cause write and/or reload errors.

Table 6.5 Timer 2 Generated Baud Rates at 12 MHz

Baud Rates	F_{OSC}= 11.0592 MHz		F_{OSC}= 12 MHz	
	RCAP2H, RCAP2L	Error (%)	RCAP2H, RCAP2L	Error (%)
115200	65533	0	–	–
57600	65530	0	–	–
38400	65527	0	65526	2.34
28800	65524	0	65523	0.16
19200	65518	0	65516	2.34
9600	65500	0	65467	0.16
4800	65464	0	65458	0.16
2400	65392	0	65380	0.16
1200	65248	0	65224	0.16
600	64960	0	64911	0
300	64384	0	64286	0

Table 6.6 Timer 2 Generated Baud Rates at 16 MHz

Baud Rate	F_{OSC}= 14.7456 MHz		F_{OSC}= 16 MHz	
	RCAP2H, RCAP2L	Error (%)	RCAP2H, RCAP2L	Error (%)
115200	65532	0	–	–
57600	65528	0	65527	3.55
38400	65524	0	65523	0.16
28800	65520	0	65519	2.12
19200	65512	0	65510	0.16
9600	65488	0	65484	0.16
4800	65440	0	65432	0.16
2400	65344	0	65428	0.16
1200	65152	0	65119	0.08
600	64768	0	64703	0.04
300	64000	0	63869	0.02

Table 6.7 Timer 2 Generated Baud Rates at 24 MHz

Baud Rate	F_{OSC}= 22.1184 MHz		F_{OSC}= 24 MHz	
	RCAP2H, RCAP2L	Error (%)	RCAP2H, RCAP2L	Error (%)
230400	65533	0	–	–
115200	65530	0	–	–
57600	65524	0	65523	0.16
38400	65518	0	65516	2.4
28800	65512	0	65510	0.16
19200	65500	0	65497	0.16
9600	65488	0	65458	0.16
4800	65392	0	65380	0.16
2400	65248	0	65223	0.16
1200	64960	0	64911	0
600	64384	0	64286	0
300	63232	0	63036	0

6.7.3.4. Internal Baud Rate Generator (BRG)

When the internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow, the value of SPD bit (Speed Mode) in BRCON register and the value of the SMOD1 bit in PCON register:

- SPD= 1

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{2 \times 32 \times [256 - (\text{BRL})]}$$

$$\text{BRL} = 256 - \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{64 \times \text{Baud\_Rate}}$$

- SPD= 0 (Default Mode)

$$\text{Baud\_Rate} = \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{12 \times 32 \times [256 - (\text{BRL})]}$$

$$\text{BRL} = 256 - \frac{2^{\text{SMOD1}} \times F_{\text{OSC}}}{384 \times \text{Baud\_Rate}}$$

The configuration is shown in the Figure 6.10

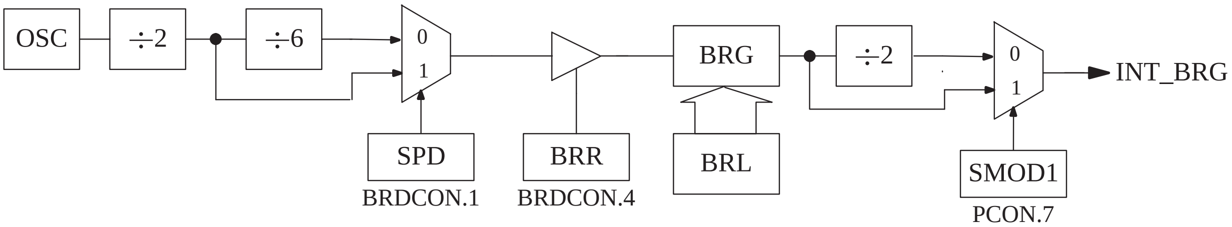


Figure 6.10 Internal Baud Rate Generator in Modes 1 and 3

Table 6.8 Internal Baud Rate Generator at 12 MHz

Baud Rate	**F_{OSC}= 11.0592 MHz**				**F_{OSC}= 12 MHz**			
	SPD	**SMOD1**	**BRL**	**Error (%)**	**SPD**	**SMOD1**	**BRL**	**Error (%)**
115200	1	0	253	0	–	–	–	–
57600	0	1	255	0	–	–	–	–
38400	1	0	247	0	1	1	246	2.34
28800	0	0	255	0	1	1	243	0.16
19200	0	1	253	0	1	1	236	2.34
9600	0	0	253	0	1	1	217	0.16
4800	0	0	250	0	0	1	243	0.16
2400	0	0	244	0	0	0	243	0.16
1200	0	0	232	0	0	0	230	0.16
600	0	0	208	0	0	0	204	0.16
300	0	0	160	0	0	0	152	0.16

Table 6.9 Internal Baud Rate Generator at 16 MHz

Baud Rates	F_{OSC}= 14.7456 MHz				F_{OSC}= 16 MHz			
	SPD	SMOD1	BRL	Error (%)	SPD	SMOD1	BRL	Error (%)
115200	1	0	254	0	–	–	–	–
57600	1	0	252	0	1	1	247	3.55
38400	0	0	255	0	1	1	243	0.16
28800	1	0	248	0	1	1	239	2.12
19200	0	0	254	0	1	0	243	0.16
9600	0	0	252	0	1	0	230	0.16
4800	0	0	248	0	1	0	204	0.16
2400	0	0	240	0	1	0	152	0.16
1200	0	0	224	0	1	0	48	0.16
600	0	0	192	0	0	1	117	0.08
300	0	0	128	0	0	0	117	0.08

Table 6.10 Internal Baud Rate Generator at 24 MHz

Baud Rates	F_{OSC}= 22.1184 MHz				F_{OSC}= 24 MHz			
	SPD	SMOD1	BRL	Error (%)	SPD	SMOD1	BRL	Error (%)
230400	1	1	253	0	–	–	–	–
115200	1	0	253	0	–	–	–	–
57600	1	0	250	0	1	1	243	0.16
38400	0	1	253	0	1	1	246	2.4
28800	1	0	244	0	1	0	243	0.16
19200	0	0	253	0	1	1	217	0.16
9600	0	0	250	0	0	1	243	0.16
4800	0	0	244	0	0	0	243	0.16
2400	0	0	232	0	0	1	204	0.16
1200	0	0	208	0	0	0	204	0.16
600	0	0	160	0	0	1	208	0.16
300	0	0	64	0	0	0	208	0.16

6.8. Registers

BDRCON (S:9Bh)

Baud Rate Control register

7	6	5	4	3	2	1	0
–	–	–	BRR	TBCK	RBCK	SPD	SRC

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
4	BRR	**Baud Rate Run control bit** Clear to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK	**Transmission Baud Rate Generator Selection bit** Clear to select Timer 1 or Timer 2 as Baud Rate Generator. Set to select Internal Baud Rate Generator.
2	RBCK	**Reception Baud Rate Generator Selection bit** Clear to select Timer 1 or Timer 2 as Baud Rate Generator. Set to select Internal Baud Rate Generator.
1	SPD	**Baud Rate Speed control bit** Clear to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC	**Baud Rate Source select bit in Mode 0** Clear to select F_{OSC}/12 as Baud Rate Generator (fixed transmission clock). Set to select the internal Baud Rate Generator.

Reset Value= XXX0 0000b

Figure 6.11 BDRCON Register

BRL (S:9Ah)

Baud Rate Reload Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Baud Rate Data** See Table 6.8 to Table 6.10.

Reset Value= 0000 0000b

Figure 6.12 BRL Register

SADDR (S:A9h)

Slave Individual Address Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Slave Individual Address.**

Reset Value= 0000 0000b

Figure 6.13 SADDR Register

SADEN (S:B9h)

Mask Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Mask Data for Slave Individual Address.**

Reset Value= 0000 0000b

Figure 6.14 SADEN Register

SBUF (S:99h)

Serial Buffer Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Data sent/received by Serial I/O Port.**

Reset Value= XXXX XXXXb

Figure 6.15 SBUF Register

SCON (S:98h)

Serial Control Register

7	6	5	4	3	2	1	0
FE/SM0	**OVR/SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit**. To select this function, set SMOD0 bit in PCON register. Set by hardware to indicate an invalid stop bit. Must be cleared by software.
	SM0	**Serial Port Mode bit 0**. To select this function, clear SMOD0 bit in PCON register. Software writes to bits SM0 and SM1 to select the Serial Port operating mode. Refer to SM1 bit for the mode selections.
6	OVR	**Overrun error bit**. To select this function, set SMOD0 bit in PCON register. Set by hardware to indicate an overwrite of the receive buffer. Must be cleared by software.
	SM1	**Serial Port Mode bit 1.** To select this function, set SMOD0 bit in PCON register. Software writes to bits SM1 and SM0 to select the Serial Port operating mode. SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register F_{OSC}/12 or variable if SRC bit in BDRCON is set 0 1 1 8–bit UART Variable 1 0 2 9–bit UART F_{OSC}/32 or F_{OSC}/64 1 1 3 9–bit UART Variable
5	SM2	**Serial Port Mode bit 2** Software writes to bit SM2 to enable and disable the multiprocessor communication and automatic address recognition features. This allows the Serial Port to differentiate between data and command frames and to recognize slave and broadcast addresses.
4	REN	**Receiver Enable bit** Clear to disable reception in mode 1, 2 and 3, and to enable transmission in mode 0. Set to enable reception in all modes.
3	TB8	**Transmit bit 8** Modes 0 and 1: Not used. Modes 2 and 3: Software writes the ninth data bit to be transmitted to TB8.
2	RB8	**Receiver bit 8** Mode 0: Not used. Mode 1 (SM2 cleared): Set or cleared by hardware to reflect the stop bit received. Modes 2 and 3 (SM2 set): Set or cleared by hardware to reflect the ninth bit received.
1	TI	**Transmit Interrupt flag** Set by the transmitter after the last data bit is transmitted. Must be cleared by software.
0	RI	**Receive Interrupt flag** Set by the receiver after the stop bit of a frame has been received. Must be cleared by software.

Reset Value= 0000 0000b

Figure 6.16 SCON Register

7. Event and Waveform Controller

7.1. Introduction

The Event and Waveform Controller (EWC) is an on–chip peripheral that performs a variety of timing and counting operations, including Pulse Width Modulation (PWM). The EWC provides also the capability for a software Watchdog Timer.

On TSC80251G1D derivatives the EWC is configured in Programmable Counter Array (PCA) mode. This mode has up to five Compare/Capture modules using the same Counter as time base:

- Counter time base may use four clock sources:
  - $F_{OSC}/12$
  - $F_{OSC}/4$
  - Timer 0 overflow (Modes 1, 2 and 3)
  - External input on P1.2 (ECI pin)
- Each module may be programmed in any of the following modes:
  - Rising and/or falling edge Capture
  - Software Timer
  - High–speed output
  - Pulse Width Modulation (PWM)

7.2. Description

7.2.1. Timers/Counters

Figure 7.1 depicts the basic logic of the Counter portion of the PCA. The CH/CL special function register pair operates as a 16–bit Counter. The selected input increments CL (low byte) register. When CL overflows, CH (high byte) register increments after two oscillator periods; when CH overflows, it sets the PCA overflow flag (CF in CCON register) generating a PCA interrupt request if ECF bit in CMOD register is set.

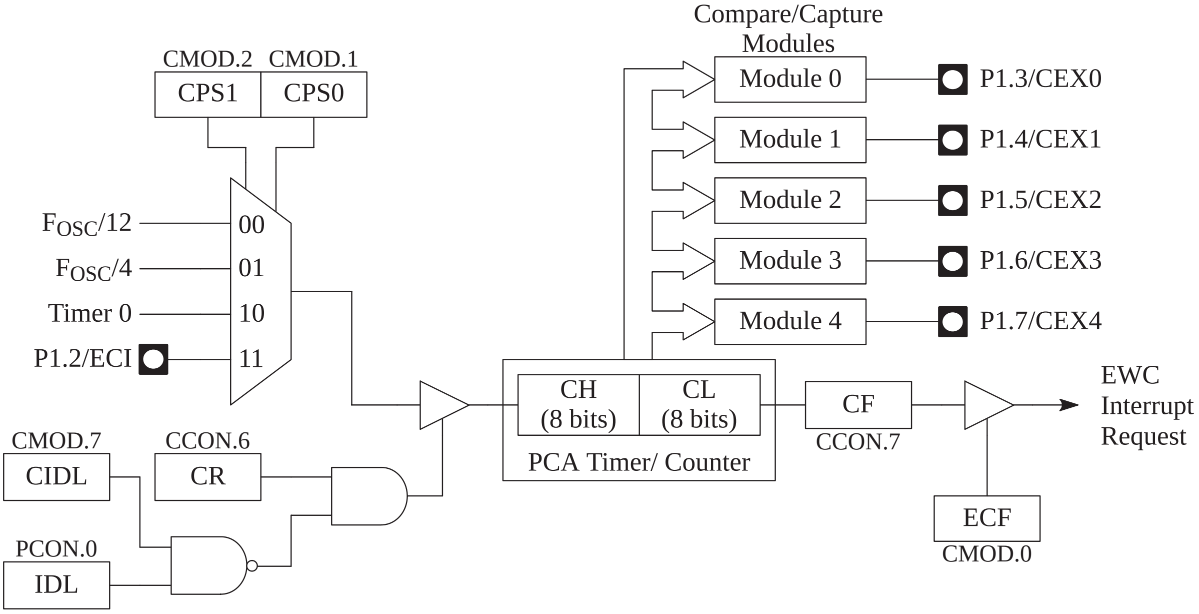


Figure 7.1 EWC Counter in PCA Mode

CPS1 and CPS0 bits in CMOD register select one of four signals as the input to the Counter (see Figure 7.1):

- $F_{OSC}/12$
  Provides a clock pulse at S5P2 of every peripheral cycle. With F_{OSC}= 16 MHz, the Counter increments every 750 ns.
- $F_{OSC}/4$
  Provides clock pulses at S1P2, S3P2, and S5P2 of every peripheral cycle. With F_{OSC}= 16 MHz, the Counter increments every 250 ns.
- Timer 0 overflow
  The CL register is incremented at S5P2 of the peripheral cycle when Timer 0 overflows. This selection provides the PCA with a programmable frequency input.
- External signal on Port 1.2/ECI
  The CPU samples the ECI pin at S1P2, S3P2 and S5P2 of every peripheral cycle. The first clock pulse (S1P2, S3P2 or S5P2) that occurs following a high–to–low transition at the ECI pin increments the CL register. The maximum input frequency for this input selection is $F_{OSC}/8$.

Setting the run control bit (CR in CCON register) turns the PCA Counter on, if the output of the NAND gate (see Figure 7.1) equals logic one. The PCA Counter continues to operate during idle mode unless CIDL bit of CMOD register is set. CPU can read the contents of CH and CL registers at any time. However, writing to them is inhibited while they are counting i.e., when CR bit is set.

7.2.2. Compare/Capture Modules

Each Compare/Capture module is made up of a Compare/Capture register pair (CCAPxH/CCAPxL), a 16–bit comparator and various logic gates and signal transition selectors. The registers store the time or count at which an external event occurred (capture) or at which an action should occur (comparison). For example, in the PWM mode, the low–byte register controls the duty cycle of the output waveform.
The logical configuration of a Compare/Capture module depends on its mode of operation.

Each module can be independently programmed for operation in any of the following modes:

- 16–bit Capture mode with triggering on the positive edge, negative edge or either edge
- Compare modes:
  - 16–bit software Timer
  - 16–bit high–speed output
  - 16–bit Watchdog Timer (module 4 only)
  - 8–bit Pulse Width Modulation

The Compare function provides the capability for operating the five modules as Timers, event Counters or Pulse Width Modulators. Four modes employ the Compare function: 16–bit software Timer mode, high–speed output mode, WDT mode and PWM mode. In the first three of these, the Compare/Capture module continuously compares the 16–bit PCA Counter value with the 16–bit value pre–loaded into the module's CCAPxH/CCAPxL register pair. In the PWM mode, the module continuously compares the value in the low–byte PCA Counter register (CL) with an 8–bit value in the CCAPxL module register. Comparisons are made three times per peripheral cycle to match the fastest PCA Counter clocking rate ($F_{OSC}/4$).

Setting ECOMx bit in a module's mode register (CCAPMx) selects the Compare function for that module. To use the modules in the Compare modes, observe the following general procedure:

  - Select the module's mode of operation.
  - Select the input signal for the PCA Counter.
  - Load the comparison value into the module's Compare/Capture register pair.
  - Set the PCA Counter run Counter bit.
  - After a match causes an interrupt, clear the module's Compare/Capture flag.

- No operation

Bit combinations programmed into a Compare/Capture module's mode register (CCAPMx) determine the operation mode. Figure 7.9 provides bit definition and Table 7.1 lists the bit combinations of the available modes. Other bit combinations are invalid and produce undefined results.

The Compare/Capture modules perform their programmed functions when their common time base, the PCA Counter, runs. The Counter is turned on and off with CR bit in CCON register. To disable any given module, program it for the "no operation" mode. The occurrence of a Capture, software Timer, or high–speed output event in a Compare/Capture module sets the module's Compare/Capture flag (CCFx) in CCON register and generates a PCA interrupt request if the corresponding enable bit in CCAPMx register is set.

The CPU can read or write CCAPxH and CCAPxL registers at any time.

Table 7.1 PCA Module Modes (x= 0, 1, 2, 3, 4)[1]

ECOMx	CAPPx	CAPNx	MATx	TOGx	PWMx	ECCFx	Module Mode
0	0	0	0	0	0	0	No operation
X[2]	1	0	0	0	0	X[2]	16–bit Capture on positive–edge trigger at CEXx
X[2]	0	1	0	0	0	X[2]	16–bit Capture on negative–edge trigger at CEXx
X[2]	1	1	0	0	0	X[2]	16-bit Capture on positive/negative-edge trigger at CEXx
1	0	0	1	0	0	X[2]	Compare software Timer
1	0	0	1	1	0	X[2]	Compare high–speed output
1	0	0	0	0	1	0	Compare 8–bit PWM
1	0	0	1	X[2]	0	X[2]	Compare PCA WDT[3]

Notes:

1. *This table shows the CCAPMx register bit combinations for selecting the operating modes of the PCA Compare/Capture modules. Other bit combinations are invalid.*
2. *X= Do not care.*
3. *For the PCA WDT mode, set also WDTE bit in CMOD register to enable the reset output signal (Module 4 only).*

7.2.2.1. 16-bit Capture Mode

The Capture mode (see Figure 7.2) provides the PCA with the ability to measure periods, pulse widths, duty cycles and phase differences at up to five separate inputs. External I/0 pins CEX0 through CEX4 are sampled for signal transitions (positive and/or negative as specified). When a Compare/Capture module programmed for the Capture mode detects the specified transition, it captures the PCA Counter value. This records the time at which an external event is detected, with a resolution equal to the Counter clock period.

To program a Compare/Capture module for the 16–bit Capture mode, program the CAPPx and CAPNx bits in the module's CCAPMx register as follows:

- To trigger the Capture on a positive transition, set CAPPx and clear CAPNx
- To trigger the Capture on a negative transition, set CAPNx and clear CAPPx
- To trigger the Capture on a positive or negative transition, set both CAPPx and CAPNx

Table 7.1 lists the bit combinations for selecting module modes. For modules in the Capture mode, detection of a valid signal transition at the I/O pin (CEXx) causes hardware to load the current PCA Counter value into the Compare/Capture registers (CCAPxH/CCAPxL) and to set the module's Compare/Capture flag (CCFx) in the CCON register. If the corresponding interrupt enable bit (ECCFx) in the CCAPMx register is set, the PCA sends an interrupt request to the EWC interrupt handler.

Since hardware does not clear the event flag when the interrupt is processed, the user must clear the flag by software. A subsequent Capture by the same module overwrites the existing captured value. To preserve a captured value, save it in RAM with the interrupt service routine before the next Capture event occurs.

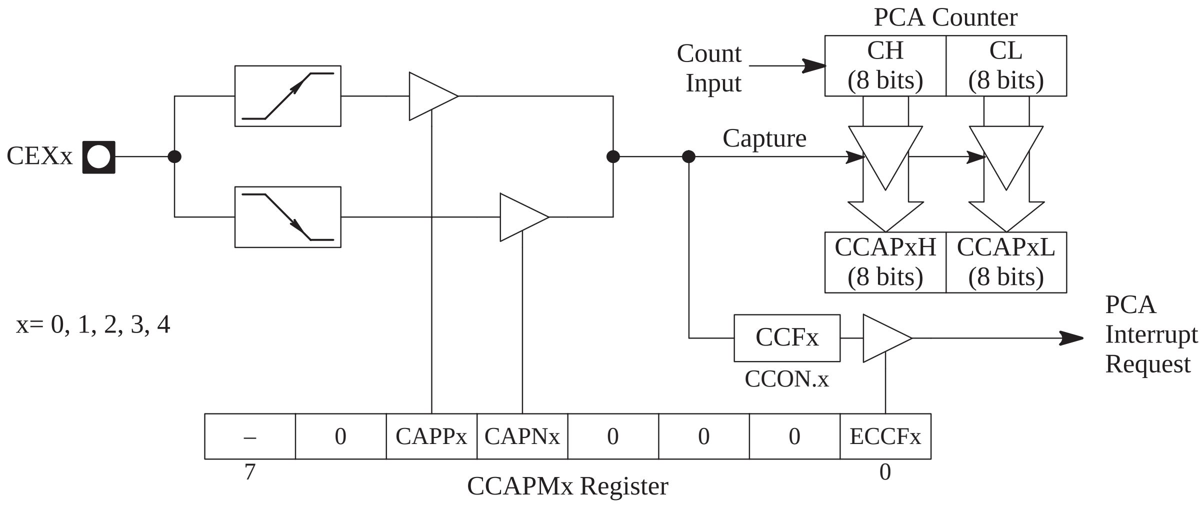


Figure 7.2 PCA 16–bit Capture Mode

7.2.2.2. 16–bit Software Timer Mode

To program a Compare/Capture module for the 16–bit software Timer mode (see Figure 7.3), set the ECOMx and MATx bits in the module's CCAPMx register. Table 7.1 lists the bit combinations for selecting module modes.

A match between the PCA Counter and the Compare/Capture registers (CCAPxH/CCAPxL) sets the module's Compare/Capture flag (CCFx in CCON register). This generates an interrupt request if the corresponding interrupt enable bit (ECCFx in CCAPMx register) is set. Since hardware does not clear the Compare/Capture flag when the interrupt is processed, the user must clear the flag in software. During the interrupt routine, a new 16–bit Compare value can be written to the Compare/Capture registers (CCAPxH/CCAPxL).

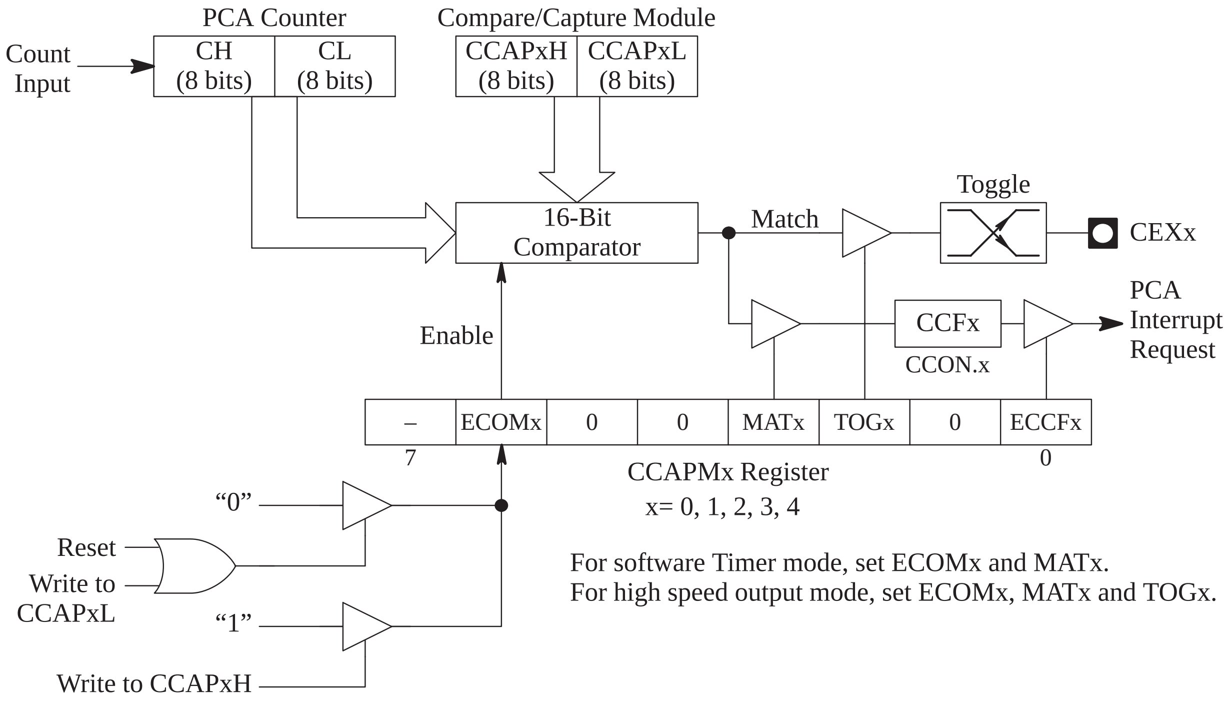


Figure 7.3 PCA Software Timer and High–Speed Output Modes

Note:

To prevent an invalid match while updating these registers, user software should write to CCAPxL first, then CCAPxH. A write to CCAPxL clears the ECOMx bit disabling the Compare–function, while a write to CCAPxH sets the ECOMx bit enabling the Compare function again.

7.2.2.3. High-Speed Output Mode

The high–speed output mode (see Figure 7.3) generates an output signal by toggling the module’s I/0 pin (CEXx) when a match occurs. This provides greater accuracy than toggling pins in software because the toggle occurs before the interrupt request is serviced. Thus, interrupt response time does not affect the accuracy of the output.

To program a Compare/Capture module for the high–speed output mode, set the ECOMx, MATx, TOGx bits in the module’s CCAPMx register. Table 7.1 lists the bit combinations for selecting module modes. A match between the PCA Counter and the Compare/Capture registers (CCAPxH/CCAPxL) toggles the CEXx pin and sets the module’s Compare/Capture flag (ECCFx in CCAPMx register). This generates an interrupt if the corresponding enable bit (CCFx in CCON register) is set. By setting or clearing the CEXx pin in software, the user selects whether the match toggles the pin from low to high or vice versa.

7.2.2.4. Watchdog Timer Mode

A Watchdog Timer (WDT) provides the means to recover from routines that do not complete successfully. A WDT automatically invokes a device reset if it does not regularly receive hold–off signals. Watchdog Timers are used in applications that are subject to electrical noise, power glitches, electrostatic discharges, etc., or where high reliability is required.

In addition to the TSC80251G1D’ 14–bit hardware WDT (see chapter 10.), the PCA provides a 16–bit programmable frequency WDT as a mode option on Compare/Capture module 4. This mode generates a device reset when the count in the PCA Counter matches the value stored in the module 4 Compare/Capture registers. A PCA WDT reset has the same effect as an external reset.

Module 4 is the only PCA module that has the WDT mode (see Figure 7.4). When not programmed as a WDT, it can be used in the other modes.

To program module 4 for the PCA WDT mode:

- Select the desired input for the PCA Counter by programming CPS0 and CPS1 bits in CMOD register (see Figure 7.13).
- Enter a 16–bit comparison value in the Compare/Capture registers (CCAP4H/CCAP4L).
- Enter a 16–bit initial value in the PCA Counter (CH/CL) or use the reset value (0000h).
- The difference between these values multiplied by the PCA input pulse rate determines the running time to ”expiration.”
- Set the Counter Run control bit (CR in CCON register) to start the PCA timer.
- Set ECOM4 and MAT4 bits in CCAPM4 register and WDTE bit in CMOD register. Table 7.1 lists the bit combinations for selecting module modes.
- The PCA WDT generates a reset signal each time a match occurs.
- To hold off a PCA WDT reset, the user has three options:
  - Periodically change the comparison value in CCAP4H/CCAP4L so a match never occurs.
  - Periodically change the PCA Counter value so a match never occurs.
  - Disable the module 4 reset output signal by clearing WDTE bit before a match occurs, then later enable it again.

The first two options are more reliable because the Watchdog Timer is not disabled as in the third option. The second option is not recommended if other PCA modules are in use, since the five modules share a common time base. Thus, in most applications the first option is the best one.

To trigger an “immediate” software reset:

- Clear the Counter Run control bit (CR in CCON register) to stop the PCA timer.
- Set the 16–bit comparison value in the Compare/Capture registers (CCAP4H/CCAP4L) to the 16–bit value in the PCA Counter (CH/CL) + 1.
- Set the Counter Run control bit (CR in CCON register) to start the PCA timer.

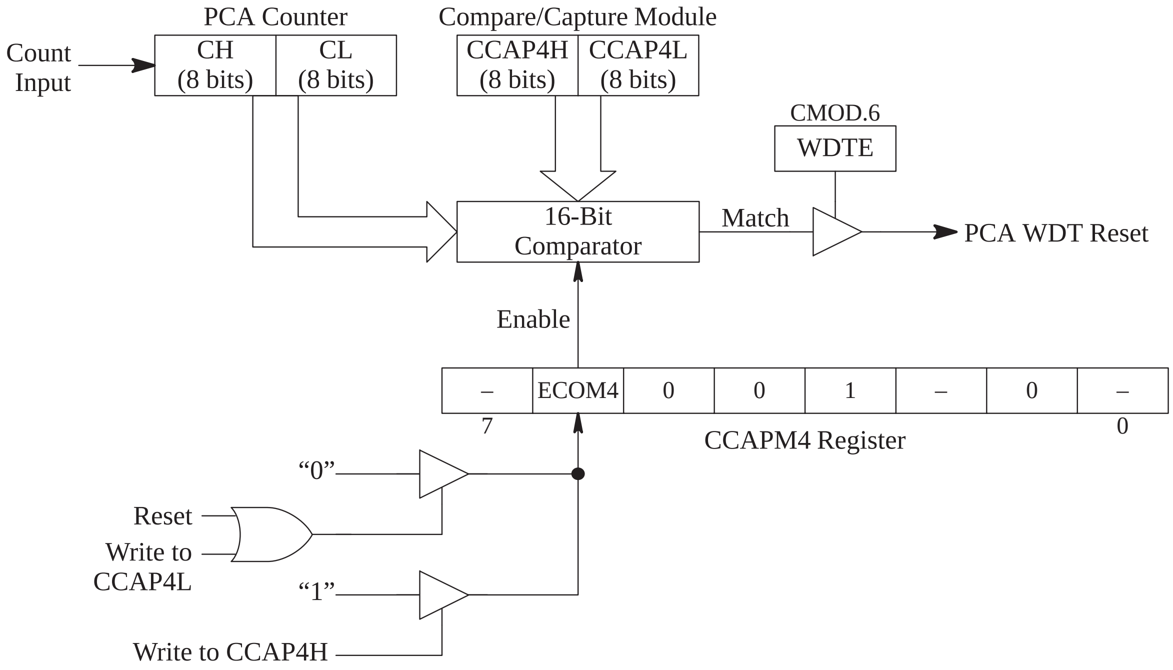


Figure 7.4 PCA Watchdog Timer Mode

7.2.2.5. Pulse Width Modulation Mode

The five PCA Compare/Capture modules can be independently programmed to function as Pulse Width Modulators (PWM). The modulated output, which has an 8–bit pulse width resolution is available on CEXx pin. The PWM output can be used to convert digital data to an analog signal with simple external circuitry.

In this mode, the value in the low byte of the PCA Timer/Counter (CL) is continuously compared with the value in the low byte of the Compare/Capture register (CCAPxL; x= 0, 1, 2, 3, 4). When CL < CCAPxL, the output waveform is low (see Figure 7.5). When a match occurs (CL= CCAPxL), the output waveform goes high and remains high until CL register rolls over from FFh to 00h, ending the period. At roll–over the output returns to low, the value in CCAPxH register is loaded into CCAPxL register, and a new period begins.

The value in CCAPxL register determines the duty cycle of the current period.

The value in CCAPxH register determines the duty cycle of the following period.

Changing the value in CCAPxL over time modulates the pulse width. As depicted in Figure 7.6, the 8–bit value in CCAPxL can vary from 0 (100% duty cycle) to 255 (0.4% duty cycle).

To program a Compare/Capture module for the PWM mode:

- Set ECOMx and PWMx bits in the module’s CCAPMx register. Table 7.1 lists the bit combinations for selecting module modes.
- Select the desired input for the PCA Counter by programming CPS0 and CPS1 bits in CMOD register.
- Enter an 8–bit value in CCAPxL to specify the duty cycle of the first period of the PWM output waveform.
- Enter an 8–bit value in CCAPxH to specify the duty cycle of the second period.
- Set the Counter run bit (CR in CCON register) to start the PCA Counter.

Note:
To change the value in CCAPxL without glitches, write the new value to the high byte register (CCAPxH). This value is shifted by hardware into CCAPxL when CL rolls over from FFh to 00h.

The frequency of the PWM output equals the frequency of the PCA Counter input signal divided by 256. The highest frequency occurs when the $F_{OSC}/4$ input is selected for the PCA Counter. For F_{OSC}= 24 MHz, this is 23.4 KHz.

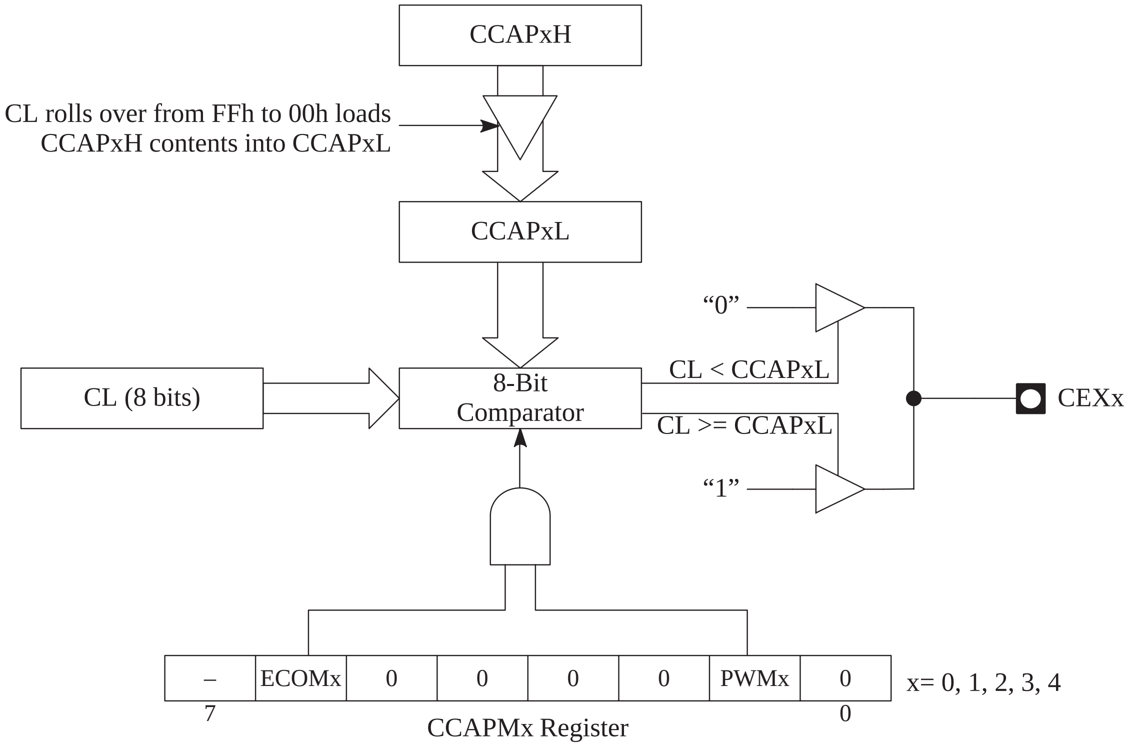


Figure 7.5 PWM Mode

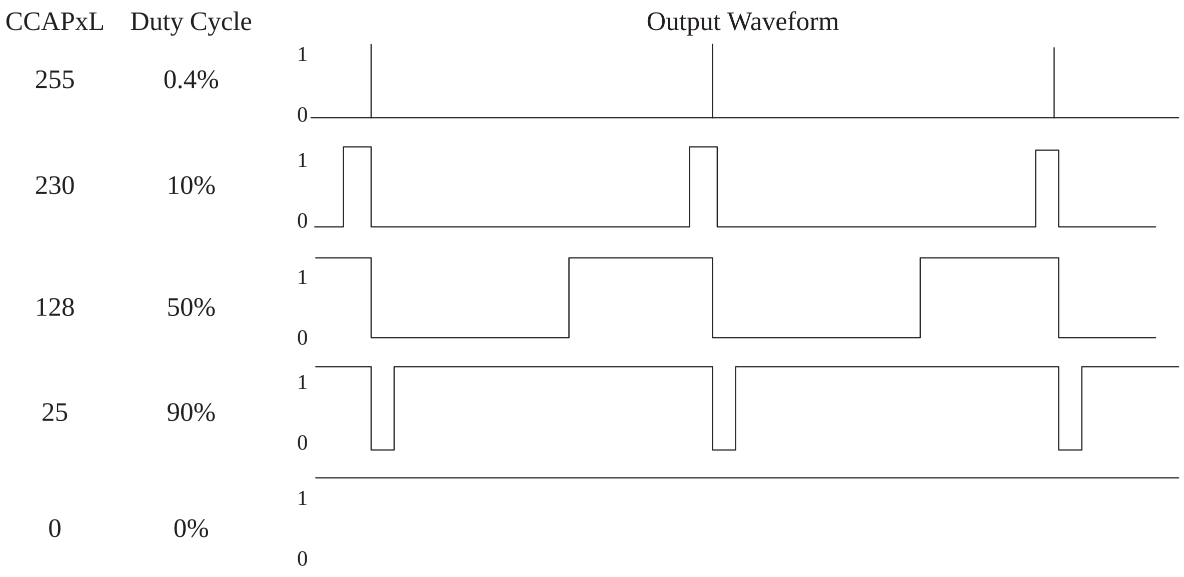


Figure 7.6 PWM Variable Duty Cycle

7.3. Registers

CCAP0H (S:FAh)
CCAP1H (S:FBh)
CCAP2H (S:FCh)
CCAP3H (S:FDh)
CCAP4H (S:FEh)
High Byte Compare/Capture Module x Registers (x= 0, 1, 2, 3, 4)

High Byte Compare/Capture Module x

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High byte of EWC–PCA comparison or capture values.**

Reset Value= 0000 0000b

Figure 7.7 EWC: CCAPxH Registers (x= 0, 1, 2, 3, 4)

CCAP0L (S:EAh)
CCAP1L (S:EBh)
CCAP2L (S:ECh)
CCAP3L (S:EDh)
CCAP4L (S:EEh)
Low Byte Compare/Capture Module x Registers (x= 0, 1, 2, 3, 4)

Low Byte Compare/Capture Module x

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low byte of EWC–PCA comparison or capture values.**

Reset Value= 0000 0000b

Figure 7.8 EWC: CCAPxL Registers (x= 0, 1, 2, 3, 4)

CCAPM0 (S:DAh)
CCAPM1 (S:DBh)
CCAPM2 (S:DCh)
CCAPM3 (S:DDh)
CCAPM4 (S:DEh)
EWC–PCA Compare/Capture Module x Mode registers (x= 0, 1, 2, 3, 4)

7	6	5	4	3	2	1	0
–	ECOMx	CAPPx	CAPNx	MATx	TOGx	PWMx	ECCFx

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
6	ECOMx	**Enable Compare Mode Module x bit** Clear to disable the Compare function. Set to enable the Compare function.. The Compare function is used to implement the software Timer, the high-speed output, the Pulse Width Modulator (PWM) and Watchdog Timer (WDT).
5	CAPPx	**Capture Mode (Positive) Module x bit** Clear to disable the Capture function triggered by a positive edge on CEXx pin. Set to enable the Capture function triggered by a positive edge on CEXx pin
4	CAPNx	**Capture Mode (Negative) Module x bit** Clear to disable the Capture function triggered by a negative edge on CEXx pin. Set to enable the Capture function triggered by a negative edge on CEXx pin.
3	MATx	**Match Module x bit** Set when a match of the PCA Counter with the Compare/Capture register sets CCFx bit in CCON register, flagging an interrupt. Must be cleared by software.
2	TOGx	**Toggle Module x bit** The toggle mode is configured by setting ECOMx, MATx and TOGx bits (see Table 7.1). Set to enable toggle at CEXx pin. Clear to disable toggle at CEXx pin.
1	PWMx	**Pulse Width Modulation Module x Mode bit** Set to configure the module x as an 8-bit Pulse Width Modulator with output waveform on CEXx pin . Must be cleared by software.
0	ECCFx	**Enable CCFx Interrupt bit** Clear to disable CCFx bit in CCON register to generate an interrupt request. Set to enable CCFx bit in CCON register to generate an interrupt request.

Reset Value= X000 0000b

Figure 7.9 CCAPMx Registers (x= 0, 1, 2, 3, 4)

CCON (S:D8h)

Timer/Counter Control Register

7	6	5	4	3	2	1	0
CF	**CR**	**–**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

Bit Number	Bit Mnemonic	Description
7	CF	**PCA Timer/Counter Overflow flag** Set by hardware when the PCA Timer/Counter rolls over. This generates a PCA interrupt request if the ECF bit in CMOD register is set. Must be cleared by software.
6	CR	**PCA Timer/Counter Run Control bit** Clear to turn the PCA Timer/Counter off. Set to turn the PCA Timer/Counter on.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	CCF4	**PCA Module 4 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF4 bit in CCAPM4 register is set. Must be cleared by software.
3	CCF3	**PCA Module 3 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF3 bit in CCAPM3 register is set. Must be cleared by software.
2	CCF2	**PCA Module 2 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF2 bit in CCAPM2 register is set. Must be cleared by software.
1	CCF1	**PCA Module 1 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF1 bit in CCAPM1 register is set. Must be cleared by software.
0	CCF0	**PCA Module 0 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF0 bit in CCAPM0 register is set. Must be cleared by software.

Reset Value= 00X0 0000b

Figure 7.10 CCON Register

CH (S:F9h)

Timer/Counter Registers

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High byte of Timer/Counter.**

Reset Value= 0000 00000b

Figure 7.11 CH Register

CL (S:E9h)

Timer/Counter Registers

Low Byte of Timer/Counter Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low byte of Timer/Counter.**

Reset Value= 0000 00000b

Figure 7.12 CL Register

CMOD (S:D9h)

Timer/Counter Mode Register

7	6	5	4	3	2	1	0
CIDL	**WDTE**	**–**	**–**	**–**	**CPS1**	**CPS0**	**ECF**

Bit Number	Bit Mnemonic	Description
7	CIDL	**PCA Counter Idle Control bit** Clear to let the PCA running during Idle mode. Set to stop the PCA when Idle mode is invoked.
6	WDTE	**Watchdog Timer Enable bit** Clear to disable the Watchdog Timer function on EWC module 4. Set to enable the Watchdog Timer function on EWC module 4.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	CPS1	**EWC Count Pulse Select bits** CPS1 CPS0 Clock source 0 0 Internal Clock, $F_{OSC}/12$ 0 1 Internal Clock, $F_{OSC}/4$ 1 0 Timer 0 overflow 1 1 External clock at ECI/P1.2 pin (Max. Rate= $F_{OSC}/8$)
1	CPS0	
0	ECF	**Enable PCA Counter Overflow Interrupt bit** Clear to disable CF bit in CCON register to generate an interrupt. Set to enable CF bit in CCON register to generate an interrupt.

Reset Value= 00XX X000b

Figure 7.13 CMOD Register

8. SSLC/Inter–Integrated Circuit (I^{2}C) Interface

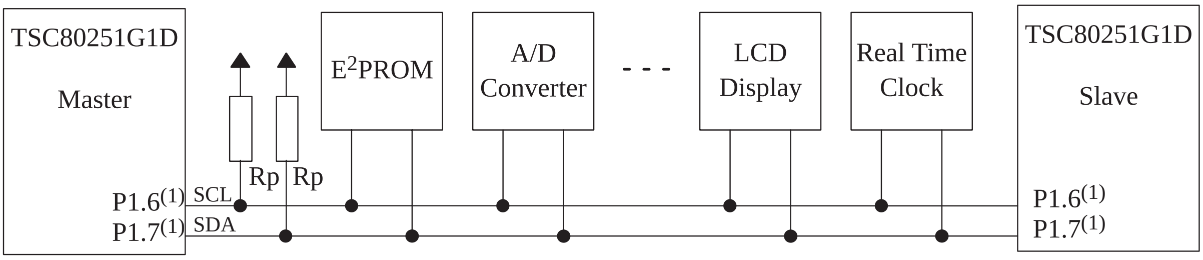
8.1. Introduction

The Synchronous Serial Link Controller (SSLC) provides the selection of synchronous serial interfaces among the three most popular ones:

- Inter–Integrated Circuit (I^{2}C) interface.
- µWire and Serial Peripheral Interface (SPI).

When the SSLC is selected in I^{2}C mode, the SPI mode is no longer available. Conversely, the I^{2}C mode is no longer available when the SSLC is selected in SPI mode.

This chapter describes the I^{2}C interface. In the rest of the chapter SSLC is written in place of I^{2}C interface.
The I^{2}C bus is a bi–directional two–wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (I^{2}C) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. The serial data transfer is limited to 100 Kbit/s in standard mode. Various communication configurations can be designed using this bus. The TSC80251G1D implements the four standard master and slave transfer modes with multimaster capability. Figure 8.1 shows a typical I^{2}C bus configuration using the TSC80251G1D in master and slave modes. All the devices connected to the bus can be master and slave.



Note:
1. The internal pull–up is disabled on P1.6/SCL and P1.7/SDA (open–drain structure) when SSLC is enabled in I^{2}C mode (SSMOD= 0 and SSPE= 1).

Figure 8.1 Typical I^{2}C Bus Configuration using the TSC80251G1D

8.2. Description

The CPU interfaces to the I^{2}C logic via the following four 8–bit special function registers: the Synchronous Serial Control register (SSCON SFR, see Figure 8.7), the Synchronous Serial Data register (SSDAT SFR, see Figure 8.8), the Synchronous Serial Control and Status register (SSCS SFR, see Figure 8.9 and Figure 8.10) and the Synchronous Serial Address register (SSADR SFR, see Figure 8.11).

SSCON is used to enable SSLC, to program the bit rate (see Table 8.1), to enable slave modes, to acknowledge or not a received data, to send a START or a STOP condition on the I^{2}C bus, and to acknowledge a serial interrupt. An hardware reset disables SSLC.

In write mode, SSCS is used to select the I^{2}C mode and to select the bit rate source. In read mode, SSCS contains a status code which reflects the status of the I^{2}C logic and the I^{2}C bus. The three least significant bits are always zero. The five most significant bits contains the status code. There are 26 possible status codes. When SSCS contains F8h, no relevant state information is available and no serial interrupt is requested. A valid status code is available in SSCS after SSI is set by hardware and is still present until SSI has been reset by software. Table 8.2 to Table 8.6 give the status for the master modes and miscellaneous states.

SSDAT contains a byte of serial data to be transmitted or a byte which has just been received. It is addressable while it is not in process of shifting a byte. This occurs when I^{2}C logic is in a defined state and the serial interrupt flag is set. Data in SSDAT remains stable as long as SSI is set. While data is being shifted out, data on the bus is simultaneously shifted in; SSDAT always contains the last byte present on the bus.

SSADR may be loaded with the 7–bit slave address (7 most significant bits) to which SSLC will respond when programmed as a slave transmitter or receiver. The LSB is used to enable general call address (00h) recognition.

Figure 8.2 shows how a data transfer is accomplished on the I^{2}C bus.

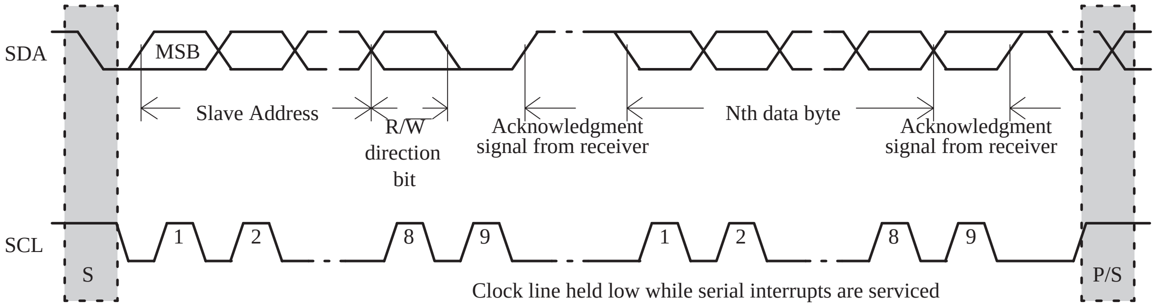

SDA
MSB
Slave Address
R/W direction bit
Acknowledgment signal from receiver
Nth data byte
Acknowledgment signal from receiver
SCL
1
2
8
9
1
2
8
9
S
P/S
Clock line held low while serial interrupts are serviced

Figure 8.2 Complete Data Transfer on I^{2}C Bus

The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave transmitter
- Slave receiver

Data transfer in each mode of operation are shown in Figure 8.3 to Figure 8.6. These figures contain the following abbreviations:

A	Acknowledge bit (low level at SDA)
$\overline{A}$	Not acknowledge bit (high level on SDA)
Data	8–bit data byte
S	START condition
P	STOP condition
MR	Master Receive
MT	Master Transmit
SLA	Slave Address
GCA	General Call Address (00h)
R	Read bit (high level at SDA)
W	Write bit (low level at SDA)

In Figure 8.3 to Figure 8.6, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in SSCS. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When the serial interrupt routine is entered, the status code in SSCS is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table 8.2 to Table 8.6.

8.2.1. Interface and Bit Rate Source Selection

The bit rate can be derived from an external bit rate generator or from the internal I^{2}C bit rate controller. The Clock Source Selection bit (SSBRS) selects the external or internal bit rate (see Figure 8.9). Before the SSLC can be enabled, SSCS must be initialized as follows:

SSBRS	–	–	–	–	–	–	SSMOD
Bit Rate Source	–	–	–	–	–	–	0

SSMOD selects the I^{2}C mode among the two available. When in the external bit rate generator configuration, the bit rate generated depends on the content of the Synchronous Serial Bit Rate register (SSBR, see Figure 8.12). The bit rate generated is given by the following formula:

$$Br= \frac{F_{OSC}}{4 \cdot (\text{SSBR value}+3)}$$

8.2.2. Master Transmitter Mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 8.3). Before the master transmitter mode can be entered, SSCON must be initialized as follows:

SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0
X	1	0	0	0	X	X	X

SSCR0, SSCR1 and SSCR2 define the internal serial bit rate if external bit rate generator is not used. SSPE must be set to enable SSLC. SSSTA, SSSTO and SSI must be cleared.

The master transmitter mode may now be entered by setting the SSSTA bit. The I^{2}C logic will now monitor the I^{2}C bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SSI bit in SSCON) is set, and the status code in SSCS will be 08h. This status must be used to vector to an interrupt routine that loads SSDAT with the slave address and the data direction bit (SLA+W). The serial interrupt flag (SSI) must then be cleared before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, SSI is set again and a number of status code in SSCS are possible. There are 18h, 20h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (SSAA= logic 1). The appropriate action to be taken for each of these status code is detailed in Table 8.2. This scheme is repeated until a STOP condition is transmitted.

SSPE, SSCR2, SSCR1 and SSCR0 are not affected by the serial transfer and are not referred to in Table 8.2. After a repeated START condition (state 10h) SSLC may switch to the master receiver mode by loading SSDAT with SLA+R.

8.2.3. Master Receiver Mode

In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 8.4). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt routine must load SSDAT with the 7–bit slave address and the data direction bit (SLA+R). The serial interrupt flag (SSI) must then be cleared before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, the serial interrupt flag is set again and a number of status code in SSCS are possible. There are 40h, 48h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (SSAA= logic 1). The appropriate action to be taken for each of these status code is detailed in Table 8.3. This scheme is repeated until a STOP condition is transmitted.

SSPE, SSCR2, SSCR1 and SSCR0 are not affected by the serial transfer and are not referred to in Table 8.3. After a repeated START condition (state 10h) SSLC may switch to the master transmitter mode by loading SSDAT with SLA+W.

8.2.4. Slave Receiver Mode

In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 8.5). To initiate the slave receiver mode, SSADR and SSCON must be loaded as follows:

SSA6	SSA5	SSA4	SSA3	SSA2	SSA1	SSA0	SSGC
Own Slave Address							X

The upper 7 bits are the address to which SSLC will respond when addressed by a master. If the LSB (SSGC) is set SSLC will respond to the general call address (00h); otherwise it ignores the general call address.

SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0
Bit Rate	1	0	0	0	1	Bit Rate	Bit Rate

SSCR0, SSCR1 and SSCR2 have no effect in the slave mode. SSPE must be set to enable SSLC. The SSAA bit must be set to enable the own slave address or the general call address acknowledgement. SSSTA, SSSTO and SSI must be cleared.

When SSADR and SSCON have been initialized, SSLC waits until it is addressed by its own slave address followed by the data direction bit which must be logic 0 (W) for SSLC to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status code is detailed in Table 8.4 and Table 8.5. The slave receiver mode may also be entered if arbitration is lost while SSLC is in the master mode (see states 68h and 78h).

If the SSAA bit is reset during a transfer, SSLC will return a not acknowledge (logic 1) to SDA after the next received data byte. While SSAA is reset, SSLC does not respond to its own slave address. However, the I^{2}C bus is still monitored and address recognition may be resumed at any time by setting SSAA. This means that the SSAA bit may be used to temporarily isolate SSLC from the I^{2}C bus.

8.2.5. Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 8.6). Data transfer is initialized as in the slave receiver mode. When SSADR and SSCON have been initialized, SSLC waits until it is addressed by its own slave address followed by the data direction bit which must be logic 1 (R) for SSLC to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status code is detailed in Table 8.6. The slave transmitter mode may also be entered if arbitration is lost while SSLC is in the master mode (see state B0h).

If the SSAA bit is reset during a transfer, SSLC will transmit the last byte of the transfer and enter state C0h or C8h. SSLC is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1's as serial data. While SSAA is reset, SSLC does not respond to its own slave address. However, the I^{2}C bus is still monitored and address recognition may be resumed at any time by setting SSAA. This means that the SSAA bit may be used to temporarily isolate SSLC from the I^{2}C bus.

8.2.6. Miscellaneous States

There are two SSCS codes that do not correspond to a define SSLC hardware state (see Table 8.7). These are discussed hereafter.

Status F8h indicates that no relevant information is available because the serial interrupt flag is not yet set. This occurs between other states and when SSLC is not involved in a serial transfer.

Status 00h indicates that a bus error has occurred during a SSLC serial transfer. A bus error is caused when a START or a STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. When a bus error occurs, SSI is set. To recover from a bus error, the SSSTO flag must be set and SSI must be cleared. This causes SSLC to enter the not addressed slave mode and to clear the SSSTO flag (no other bits in S1CON are affected). The SDA and SCL lines are released and no STOP condition is transmitted.

Note:
SSLC interfaces to the external I^{2}C bus via two port 1 pins: P1.6/CEX3/SCL/SCK/WAIT# (serial clock line) and P1.7/A17/CEX4/SDA/MOSI/WCLK (serial data line). To avoid low level asserting and conflict on these lines when SSLC is enabled in I^{2}C mode, the output latches of P1.6 and P1.7 must be set to logic 1, and the other alternative function (PCA module 3 and 4, real–time synchronous wait state and A17) must not be enabled.

Table 8.1 Serial Clock Rates

SSBRS	SSCR2	SSCR1	SSCR0	Bit frequency (kHz)			F_{OSC} divided by
				F_{OSC}= 12MHz	F_{OSC}= 16MHz	F_{OSC}= 24MHz	
0	0	0	0	47	62.5	93.75	256
0	0	0	1	53.5	71.5	–	224
0	0	1	0	62.5	83	–	192
0	0	1	1	75	100	–	160
0	1	0	0	12.5	16.5	25	960
0	1	0	1	100	–	–	120
0	1	1	0	–	–	–	60
0	1	1	1	0.5 < · < 62.5 max TH1: 254	0.67 < · < 83 max TH1: 254	1 < · < 83 max TH1: 253	96 · (256 – reload value Timer 1)
1	0	X	X	11.6 < · < 100 min SSBR: 27	15.5 < · < 100 min SSBR: 37	23.3 < · < 100 min SSBR: 57	4 · (SSBR value + 3)
1	1	X	X	–	–	–	Reserved

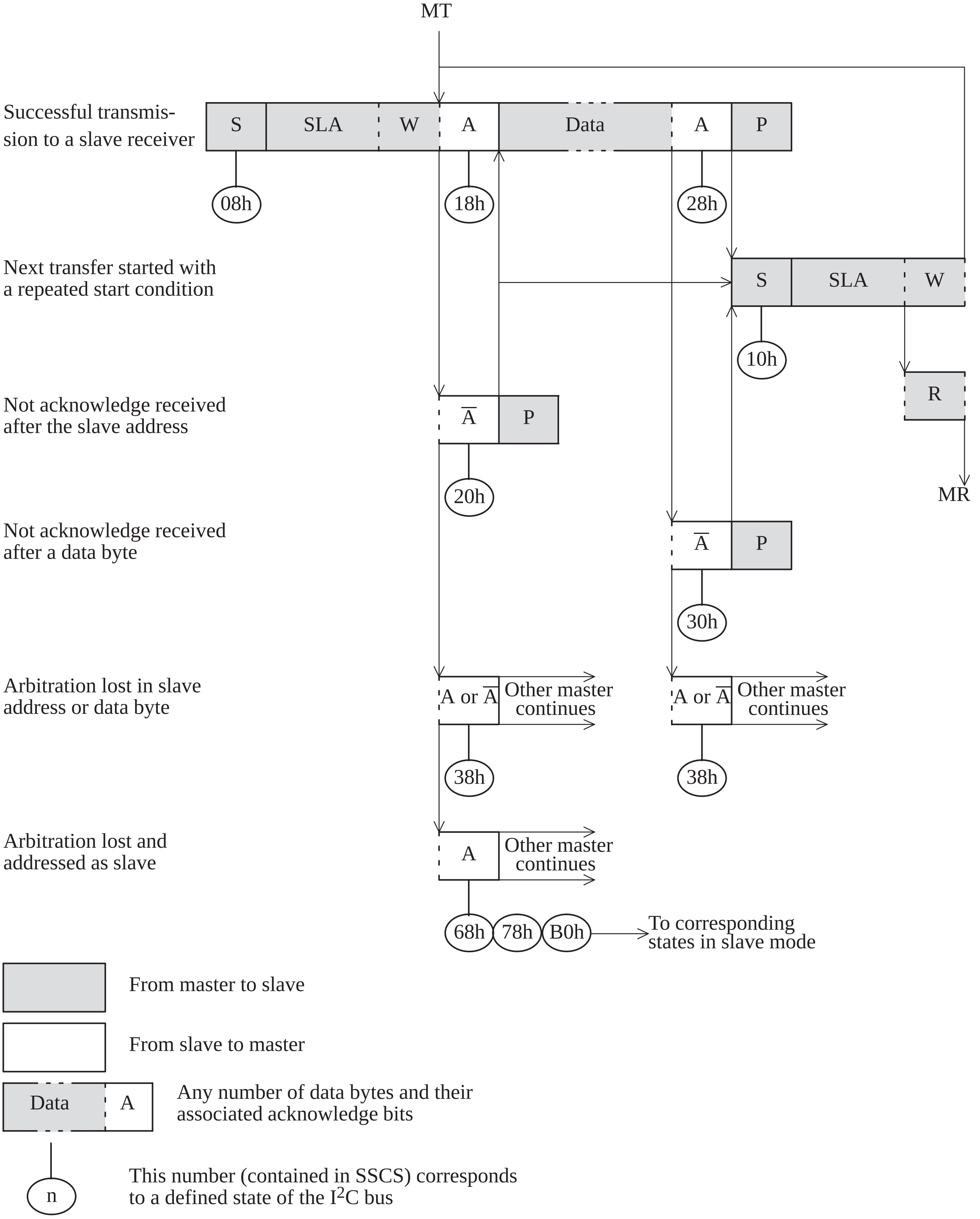

MT
Successful transmission to a slave receiver
S
SLA
W
A
Data
A
P
08h
18h
28h
Next transfer started with a repeated start condition
S
SLA
W
10h
R
MR
Not acknowledge received after the slave address
$\overline{A}$
P
20h
Not acknowledge received after a data byte
$\overline{A}$
P
30h
Arbitration lost in slave address or data byte
A or $\overline{A}$
Other master continues
38h
A or $\overline{A}$
Other master continues
38h
Arbitration lost and addressed as slave
A
Other master continues
68h
78h
B0h
To corresponding states in slave mode
From master to slave
From slave to master
Data
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in SSCS) corresponds to a defined state of the I^{2}C bus

Figure 8.3 Format and States in the Master Transmitter Mode

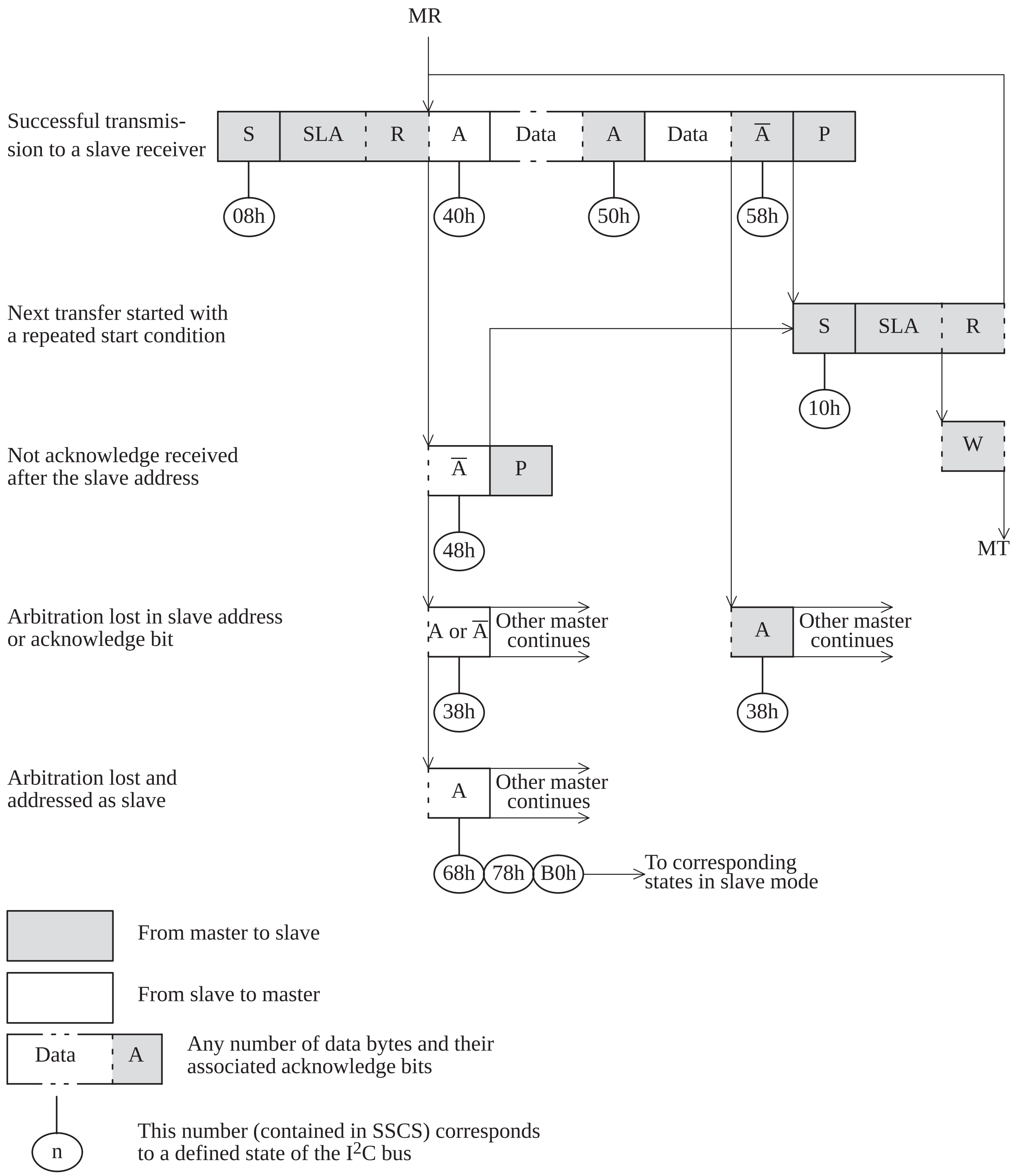

MR
Successful transmission to a slave receiver
S
SLA
R
A
Data
A
Data
$\overline{A}$
P
08h
40h
50h
58h
Next transfer started with a repeated start condition
S
SLA
R
10h
W
MT
Not acknowledge received after the slave address
$\overline{A}$
P
48h
Arbitration lost in slave address or acknowledge bit
A or $\overline{A}$
Other master continues
38h
A
Other master continues
38h
Arbitration lost and addressed as slave
A
Other master continues
68h
78h
B0h
To corresponding states in slave mode
From master to slave
From slave to master
Data
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in SSCS) corresponds to a defined state of the I^{2}C bus

Figure 8.4 Format and States in the Master Receiver Mode

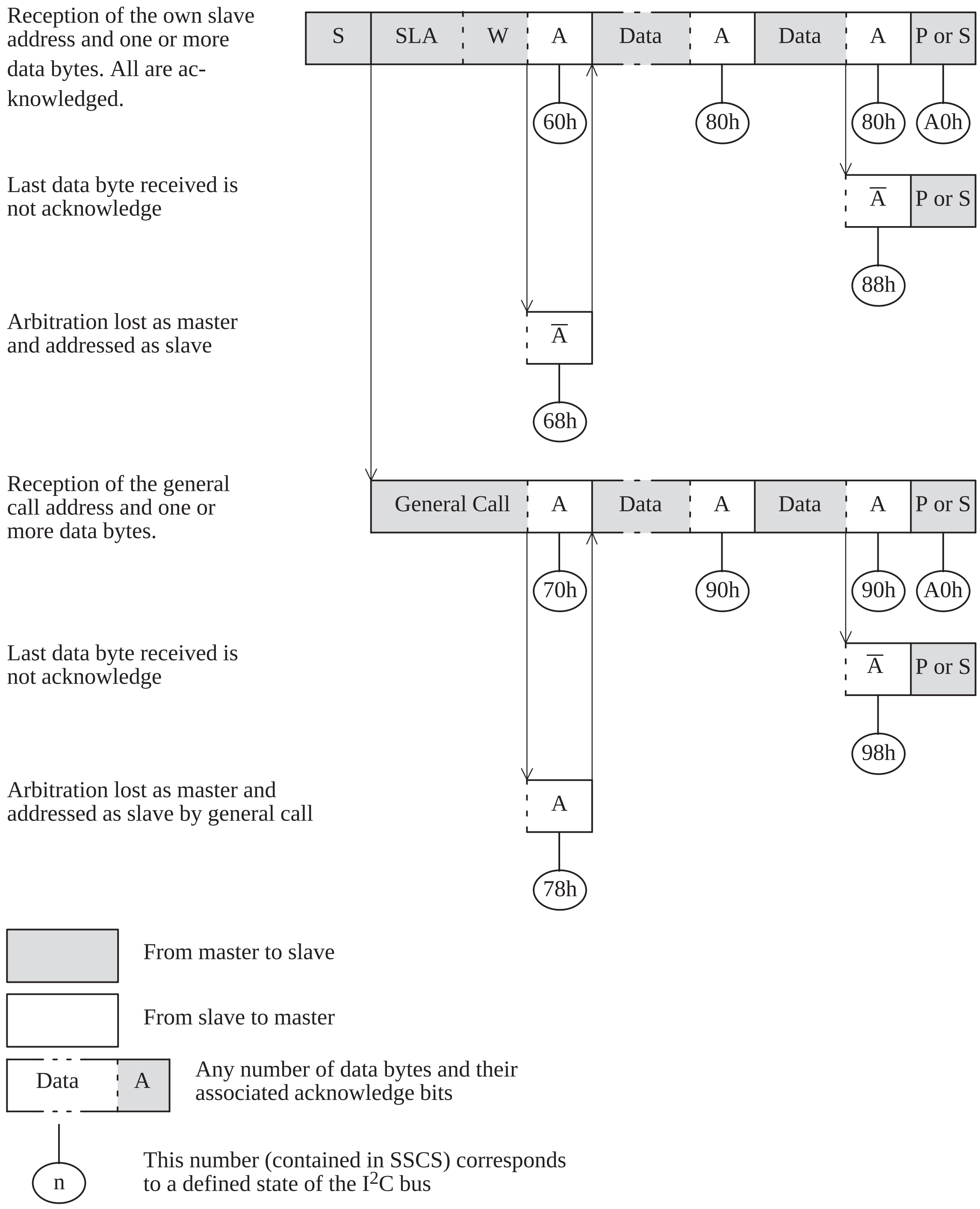

Reception of the own slave address and one or more data bytes. All are acknowledged.
S
SLA
W
A
Data
A
Data
A
P or S
60h
80h
80h
A0h
Last data byte received is not acknowledge
$\overline{A}$
P or S
88h
Arbitration lost as master and addressed as slave
$\overline{A}$
68h
Reception of the general call address and one or more data bytes.
General Call
A
Data
A
Data
A
P or S
70h
90h
90h
A0h
Last data byte received is not acknowledge
$\overline{A}$
P or S
98h
Arbitration lost as master and addressed as slave by general call
A
78h
From master to slave
From slave to master
Data
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in SSCS) corresponds to a defined state of the I^{2}C bus

Figure 8.5 Format and States in the Slave Receiver Mode

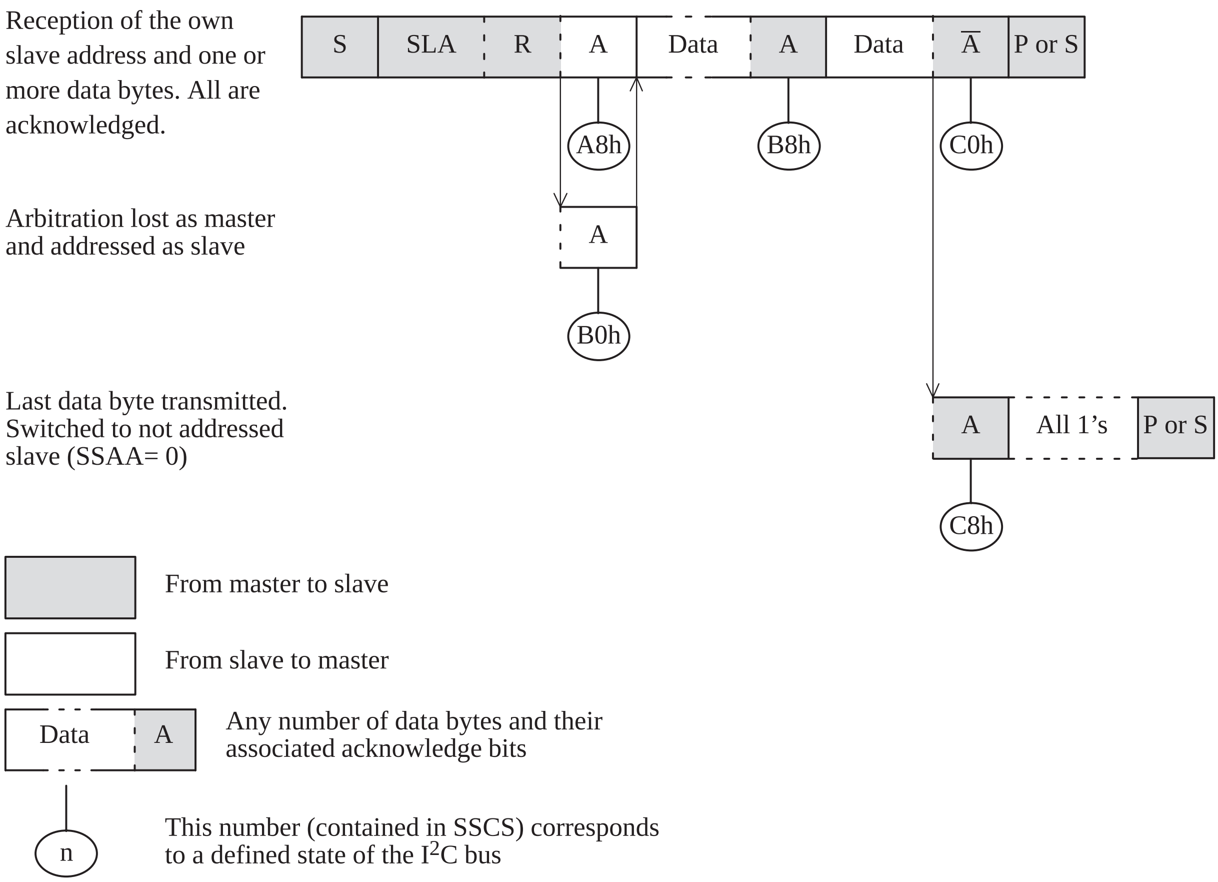

Reception of the own slave address and one or more data bytes. All are acknowledged.
S
SLA
R
A
Data
A
Data
$\overline{A}$
P or S
A8h
B8h
C0h
Arbitration lost as master and addressed as slave
A
B0h
Last data byte transmitted. Switched to not addressed slave (SSAA= 0)
A
All 1’s
P or S
C8h
From master to slave
From slave to master
Data
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in SSCS) corresponds to a defined state of the I²C bus

Figure 8.6 Format and States in the Slave Transmitter Mode

Table 8.2 Status for Master Transmitter Mode

Status	Status of the	Application software response					Next action taken by I^{2}C hardware
Code	I^{2}C bus and	To/from	To SSCON				
SSCS	I^{2}C hardware	SSDAT	SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted;
10h	A repeated START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted;
		Write SLA+R	X	0	0	X	SLA+R will be transmitted Logic will switch to master receiver mode
18h	SLA+W has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted
		No SSDAT action	1	0	0	X	Repeated START will be transmitted
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
20h	SLA+W has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted
		No SSDAT action	1	0	0	X	Repeated START will be transmitted
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset
28h	Data byte has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted
		No SSDAT action	1	0	0	X	Repeated START will be transmitted
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset
30h	Data byte has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted anyway
		No SSDAT action	1	0	0	X	Repeated START will be transmitted
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset
38h	Arbitration lost in SLA+W or data bytes	No SSDAT action	0	0	0	X	I^{2}C bus will be released and not addressed slave mode will be entered
		No SSDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

Table 8.3 Status for Master Receiver Mode

Status	Status of the	Application software response					Next action taken by I^2C hardware
Code	I^2C bus and	To/from	To SSCON				
SSCS	I^2C hardware	SSDAT	SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted	Write SLA+R	X	0	0	X	SLA+R will be transmitted;
10h	A repeated START condition has been transmitted	Write SLA+R	X	0	0	X	SLA+R will be transmitted;
		Write SLA+W	X	0	0	X	SLA+W will be transmitted Logic will switch to master transmitter mode
38h	Arbitration lost in SLA+R or NOT ACK bit	No SSDAT action	0	0	0	X	I^2C bus will be released and not addressed slave mode will be entered
		No SSDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free
40h	SLA+R has been transmitted; ACK has been received	No SSDAT action	0	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	0	0	0	1	Data byte will be received and ACK will be returned
48h	SLA+R has been transmitted; NOT ACK has been received	No SSDAT action	1	0	0	X	Repeated START will be transmitted
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset
50h	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received and NOT ACK will be returned
		Read data byte	0	0	0	1	Data byte will be received and ACK will be returned
58h	Data byte has been received; NOT ACK has been returned	Read data byte	1	0	0	X	Repeated START will be transmitted
		Read data byte	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset
		Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset

Table 8.4 Status for Slave Receiver Mode with Own Slave Address

Status Code SSCS	Status of the I^2C bus and I^2C hardware	Application software response					Next action taken by I^2C hardware
		To/from SSDAT	To SSCON				
			SSSTA	SSSTO	SSI	SSAA	
60h	Own SLA+W has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
68h	Arbitration lost in SLA+R/W as master; own SLA+W has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
80h	Previously addressed with own SLA+W; data has been received; ACK has been returned	Read data byte	X	0	0	0	Data byte will be received and NOT ACK will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned
88h	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data byte	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		Read data byte	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		Read data byte	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free
A0h	A STOP condition or repeated START condition has been received while still addressed as slave	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free

Table 8.5 Status for Slave Receiver Mode with General Call Address

Status Code SSCS	Status of the I^2C bus and I^2C hardware	Application software response					Next action taken by I^2C hardware
		To/from SSDAT	To SSCON				
			SSSTA	SSSTO	SSI	SSAA	
70h	General call address has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
78h	Arbitration lost in SLA+R/W as master; general call address has been received; ACK has been returned	No SSDAT action	X	0	0	0	Data byte will be received and NOT ACK will be returned
		No SSDAT action	X	0	0	1	Data byte will be received and ACK will be returned
90h	Previously addressed with general call; data has been received; ACK has been returned	Read data byte	X	0	0	0	Data byte will be received and NOT ACK will be returned
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned
98h	Previously addressed with general call; data has been received; NOT ACK has been returned	Read data byte	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		Read data byte	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		Read data byte	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		Read data byte	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free
A0h	A STOP condition or repeated START condition has been received while still addressed as slave	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free

Table 8.6 Status for Slave Transmitter Mode

Status	Status of the	Application software response					Next action taken by I^2C hardware
Code	I^2C bus and	To/from	To SSCON				
SSCS	I^2C hardware	SSDAT	SSSTA	SSSTO	SSI	SSAA	
A8h	Own SLA+R has been received; ACK has been returned	Write data byte	X	0	0	0	Last data byte will be transmitted
		Write data byte	X	0	0	1	Data byte will be transmitted
B0h	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Write data byte	X	0	0	0	Last data byte will be transmitted
		Write data byte	X	0	0	1	Data byte will be transmitted
B8h	Data byte in SSDAT has been transmitted; ACK has been received	Write data byte	X	0	0	0	Last data byte will be transmitted
		Write data byte	X	0	0	1	Data byte will be transmitted
C0h	Data byte in SSDAT has been transmitted; NOT ACK has been received	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free
C8h	Last data byte in SSDAT has been transmitted (SSAA= 0); ACK has been received	No SSDAT action	0	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA
		No SSDAT action	0	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1
		No SSDAT action	1	0	0	0	Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free
		No SSDAT action	1	0	0	1	Switched to the not addressed slave mode; own SLA will be recognized; GCA will be recognized if SSGC= logic 1. A START condition will be transmitted when the bus becomes free

Table 8.7 Status for Miscellaneous States

Status	Status of the	Application software response					Next action taken by I^2C hardware
code	I^2C bus and	To/from	To SSCON				
SSCS	I^2C hardware	SSDAT	SSSTA	SSSTO	SSI	SSAA	
F8h	No relevant state information available; SSI= 0	No SSDAT action	No SSCON action				Wait or proceed current transfer
00h	Bus error due to an illegal START or STOP condition	No SSDAT action	0	1	0	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and SSSTO is reset

8.3. Registers

SSCON (S:93h)

Synchronous Serial Control Register

7	6	5	4	3	2	1	0
SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0

Bit Number	Bit Mnemonic	Description
7	SSCR2	**Synchronous Serial Control Rate bit 2** see Table 8.1.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Clear to disable the SSLC in I^{2}C mode. Set to enable the SSLC in I^{2}C mode.
5	SSSTA	**Synchronous Serial Start flag** Clear not to send a START condition on the bus. Set to send a START condition on the bus.
4	SSSTO	**Synchronous Serial Stop flag** Clear not to send a STOP condition on the bus. Set to send a STOP condition on the bus.
3	SSI	**Synchronous Serial Interrupt flag** Set by hardware when a serial interrupt is requested. Must be cleared by software to acknowledge interrupt.
2	SSAA	**Synchronous Serial Assert Acknowledge flag** Clear to disable slave modes. Set to enable slave modes. Slave modes are entered when SLA or GCA (if SSGC set) is recognized. Master Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Master Transmitter Mode in progress This bit has no specific effect when in master transmitter mode. Slave Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Slave Transmitter Mode in progress Clear to isolate slave from the bus after last data byte transmission. Set to enable slave mode.
1	SSCR1	**Synchronous Serial Control Rate bit 1** see Table 8.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** see Table 8.1.

Reset Value= 0000 0000b

Figure 8.7 SSCON register

SSDAT (S:95h)

Synchronous Serial Data Register

7	6	5	4	3	2	1	0
SSD7	**SSD6**	**SSD5**	**SSD4**	**SSD3**	**SSD2**	**SSD1**	**SSD0**

Bit Number	Bit Mnemonic	Description
7:1	SSD7:1	**Synchronous Serial Address bits 7 to 1 or Synchronous Serial Data bits 7 to 1**
0	SSD0	**Synchronous Serial Address bit 0 (R/$\overline{W}$) or Synchronous Serial Data bit 0**

Reset Value= 0000 0000b

Figure 8.8 SSDAT register

SSCS (S:94h) write (SSLC I^{2}C)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSBRS	**–**	**–**	**–**	**–**	**–**	**–**	**SSMOD**

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Synchronous Serial Bit Rate Selection bit** Clear to select the bit rate controlled by SSCR2 to SSCR0. Set to select the programmable bit rate generator.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	SSMOD	**Interface Selection bit 0** Clear to select the SSLC in I^{2}C mode.

Reset Value= 0XXX XXX0b

Figure 8.9 SSCS register: write mode

SSCS (S:94h) read (SSLC I^2C)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSC4	**SSC3**	**SSC2**	**SSC1**	**SSC0**	**0**	**0**	**0**

Bit Number	Bit Mnemonic	Description
7:3	SSC4:0	**Synchronous Serial Status code bits 0 to 4** See Table 8.2 to Table 8.7.
2:0	0	**Always 0.**

Reset Value= F8h

Figure 8.10 SSCS register: read mode

SSADR (S:96h)

Synchronous Serial Address Register

7	6	5	4	3	2	1	0
SSA7	**SSA6**	**SSA5**	**SSA4**	**SSA3**	**SSA2**	**SSA1**	**SSGC**

Bit Number	Bit Mnemonic	Description
7:1	SSA7:1	**Synchronous Serial Slave Address bits 7 to 1.**
0	SSGC	**Synchronous Serial General Call bit** Clear to disable the general call address recognition. Set to enable the general call address recognition.

Reset Value= 0000 0000b

Figure 8.11 SSADR register

SSBR (S:92h)

Synchronous Serial Bit Rate Register

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7:0		**Synchronous Serial Bit Rate Data** Bit rate is given by the formula: Br= F_{OSC} / (4 · (SSBR value + 3)), Br in KHz (F_{OSC} in MHz).

Reset Value= 0000 0000b

Figure 8.12 SSBR register

9. SSLC/Synchronous Peripheral Interface (μWire/SPI)

9.1. Introduction

The Synchronous Serial Link Controller (SSLC) provides the selection of synchronous serial interfaces among the three most popular ones:

- Inter–Integrated Circuit (I^2C) interface.
- μWire and Serial Peripheral Interface (SPI).

When the SSLC is selected in SPI mode, the I^2C mode is no longer available. Conversely, the SPI mode is no longer available when the SSLC is selected in I^2C mode.

This chapter describes the SPI. This synchronous interface allows several SPI microcontrollers or μWire and SPI–type peripherals to be interconnected.

Figure 9.1 shows a typical SPI bus configuration using one TSC80251G1D master and many slave peripherals. The bus is made of three wires connecting all the devices together:

- Master Output Slave Input (MOSI): it is used to transfer data in series from the master to a slave.
  It is driven by the master.
- Master Input Slave Output (MISO): it is used to transfer data in series from a slave to the master.
  It is driven by the selected slave.
- Serial Clock (SCK): it is used to synchronize the data movement both in and out the devices through their MOSI and MISO lines. It is driven by the master for eight clock cycles which allows to exchange one byte on the serial lines.

Each slave peripheral is selected by one Slave Select pin (SS#). If there is only one slave, it may be continuously selected with SS# tied to a low level. Otherwise, the TSC80251G1D may select each device by software through port pins (Pn.x). Special care should be taken not to select two slaves at the same time to avoid bus conflicts.

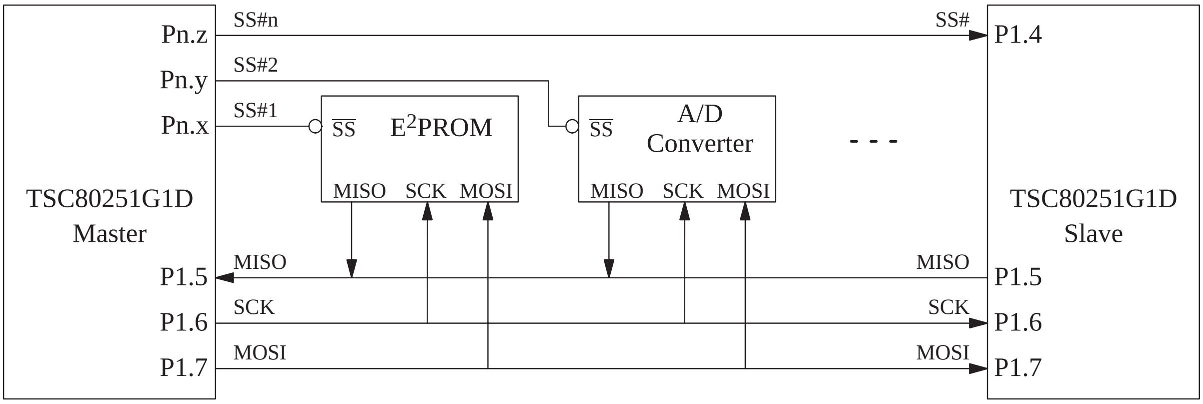


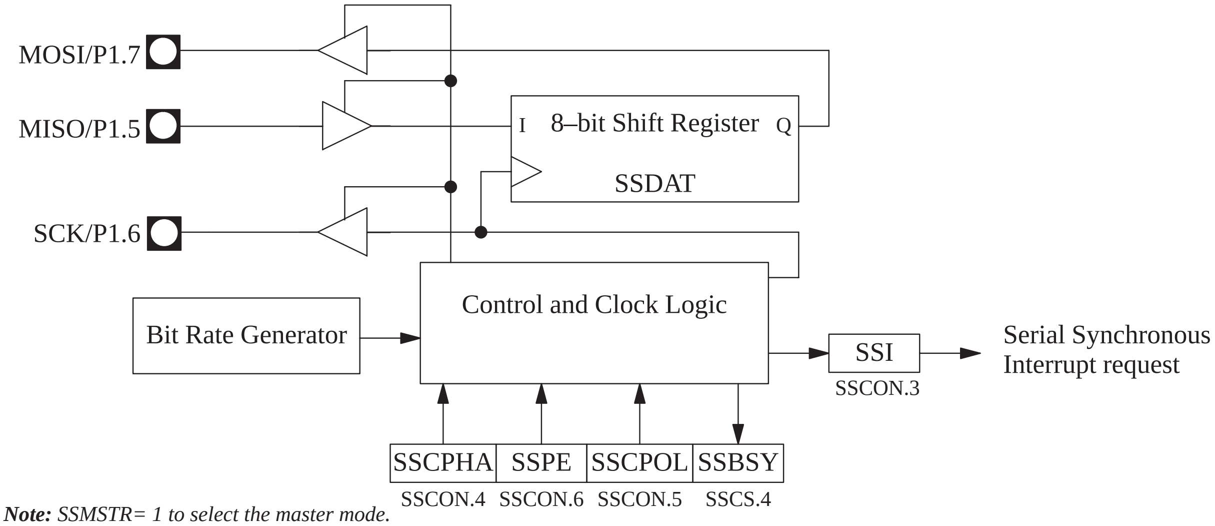
Figure 9.1 Typical SPI Bus Configuration using the TSC80251G1D

To allow easy communication of many TSC80251G1D together or with other SPI masters, a slave SPI mode is also provided. When SPI is configured as a slave, P1.4 may be used as SS#.

9.2. Description

9.2.1. Master Mode

Figure 9.2 shows the SPI block diagram in master mode. Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the data register (SSDAT). The byte begins shifting out on the MOSI pin under the control of the bit rate generator. This generator also controls the shift register of the slave peripheral through the SCK output pin. As the byte shifts out, another byte shifts in from the slave peripheral on the MISO pin.

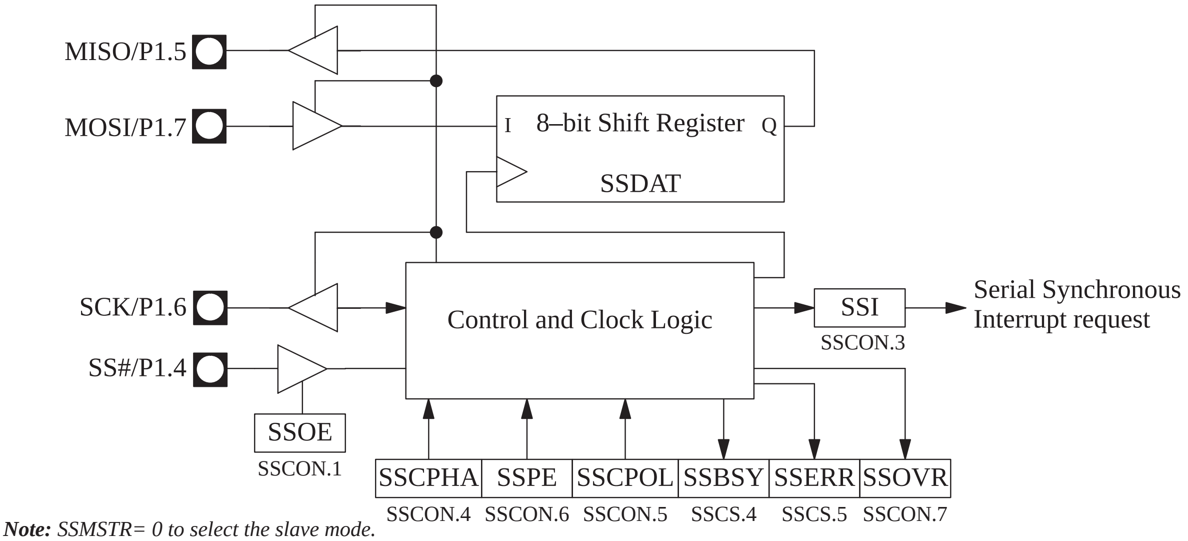


Note: *SSMSTR= 1 to select the master mode.*

Figure 9.2 SPI Master Mode Block Diagram

9.2.2. Slave Mode

Figure 9.3 shows the SPI block diagram in slave mode. In slave mode, before a data transmission occurs, the SS# pin of the slave SPI must be at logic 0. SS# must remain low until the transmission is complete. In the slave SPI module, data enters the shift register through the MOSI pin under the control of the serial clock from the master SPI module provided on the SCK input pin. When the master SPI starts a transmission, the data in the shift register begins shifting out on the MOSI pin.



***Note:** SSMSTR= 0 to select the slave mode.*

Figure 9.3 SPI Slave Mode Block Diagram

9.2.3. Bit Rate

In master mode, the bit rate can be selected from fixed bit rate or from a bit rate generator. The SSCR1 and SSCR0 bits control the fixed bit rate while SSBRS bit and SSBR register control the bit rate generator. (see Table 9.1). Figure 9.4 shows the bit rate generator block diagram.

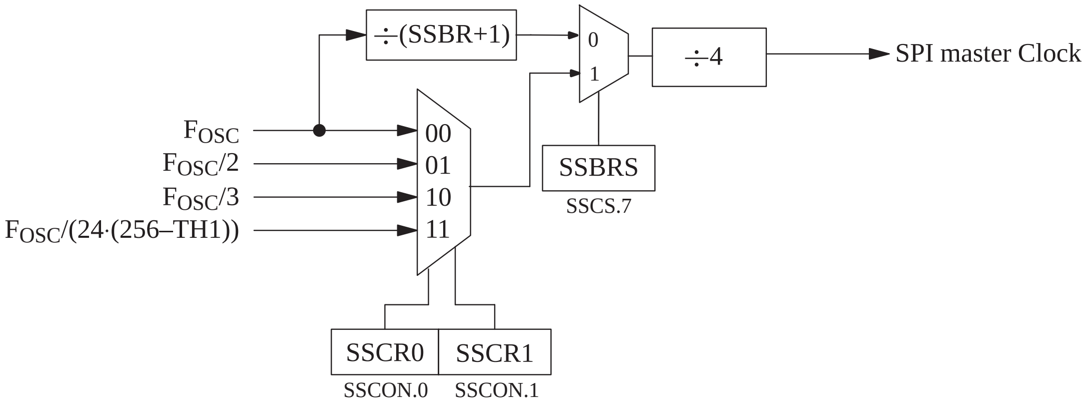


Figure 9.4 Bit Rate Generator Block Diagram

Table 9.1 Serial Clock Rates

SSBRS	SSCR1	SSCR0	Bit frequency (kHz)			F_{OSC} divided by
			F_{OSC}= 12 MHz	F_{OSC}= 16 MHz	F_{OSC}= 24 MHz	
0	0	0	3000	4000	6000	4
0	0	1	1500	2000	3000	8
0	1	0	1000	1333.33	2000	12
0	1	1	0.49 < · < 125	0.65 < · < 167	0.98 < · < 250	96 · (256 – reload value Timer 1) (reload value range: 0–255 in mode 2)
1	X	X	11.7 < · < 3000	15.6 < · < 4000	23.4 < · < 6000	4 · (SSBR value + 1)

9.2.4. Data Transfer

The Synchronous Serial Polarity bit (SSCPOL in SSCON) defines the default SCK line level in idle state[1]. Then the Synchronous Serial Phase bit (SSCPHA in SSCON) defines the edges on which the input data are sampled and the edges on which and the output data are shifted (see Figure 9.5 and Figure 9.6). It applies to both master and slave modes. The MISO signal is output from the selected slave and the MOSI signal is the output from the master. The master is capturing data from the MISO line while the slave is capturing data from the MOSI line. The SS# line is the slave select input of the slave peripheral.

Note:

1. When the peripheral is disabled (SSPE= 0), default SCK line is high level.

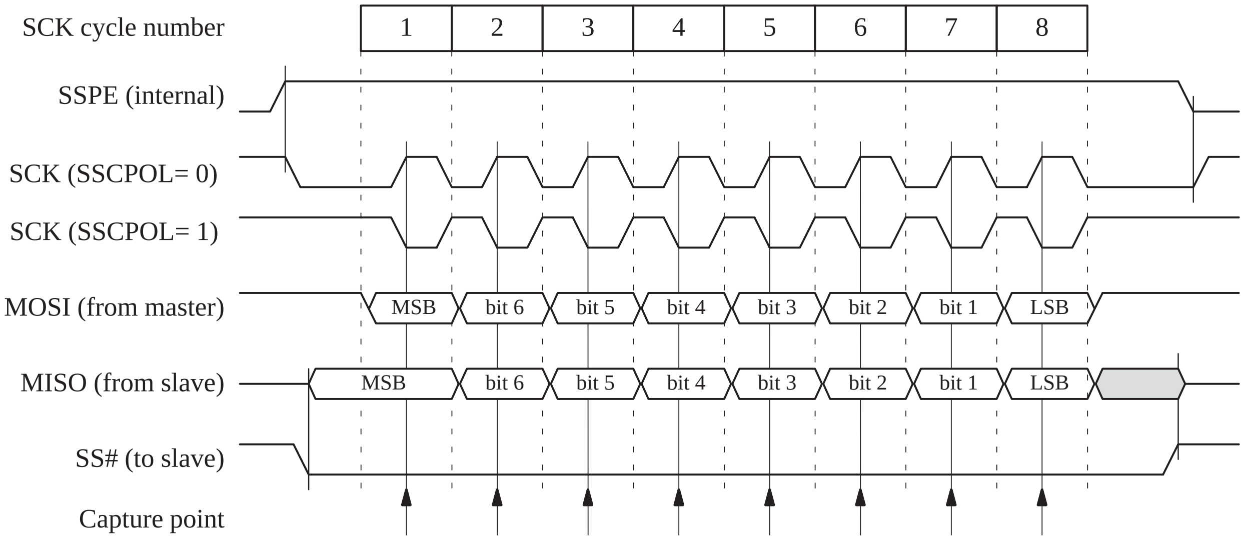


Figure 9.5 Data Transmission Format (SSCPHA= 0)

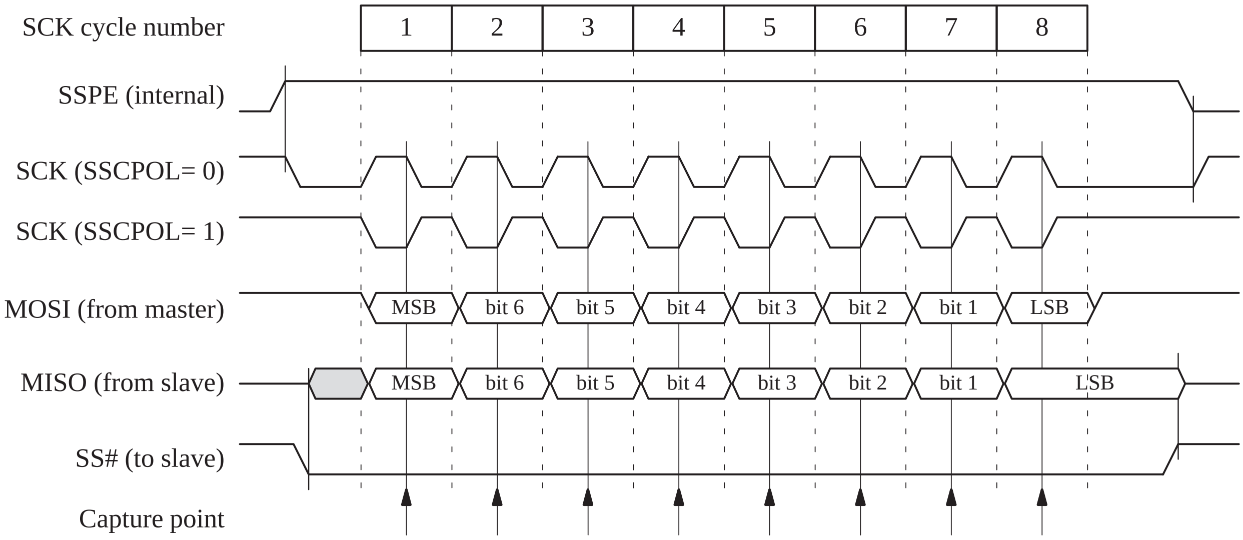


Figure 9.6 Data Transmission Format (SSCPHA= 1)

9.3. Configuration

The SPI configuration is made through three registers:

- The Synchronous Serial Control and Status register (SSCS)
- The Synchronous Serial Control register (SSCON)
- The Synchronous Serial Bit Rate register (SSBR)

Once the SPI is configured, the data exchange is made using:

- SSDAT
- SSCS
- SSCON

After reset SSLC is configured in I^{2}C mode (SSMOD bit in SSCS cleared). Prior any SPI operation and before attempting to read or write any of the above registers, the Synchronous Serial Mode bit (SSMOD in SSCS) must be set to select the µWire/SPI mode. Then the SPI registers are enabled and rest of the configuration can be done.

To avoid unexpected behavior, the Synchronous Serial Peripheral Enable bit (SSPE in SSCON) must not be set before configuring the SPI. This bit may be cleared at any time and disables the SSLC whatever its configuration, then all the SPI outputs are disabled. Once the SPI has been configured (see hereafter), SSPE may be set to enable the SPI operation, then its outputs are enabled and set according to the configuration.

The Synchronous Serial Master bit (SSMSTR in SSCON) allows to choose between the master and the slave mode. After reset, SSMSTR is set and the SPI is in master mode. A previous revision of this peripheral interface was not supporting the slave mode. When the slave mode is not supported, SSMSTR is reserved and it is CLEARED after reset. Then SSOE is RESERVED and must not be set.

When the SSLC is in SPI mode, SSBR, SSCON, SSCS and SSDAT registers may be read and written at any time while there is no on–going exchange. However, special care should be taken not to change the SSCPHA and SSCPOL bits once the SSPE bit has been set.

Note:
The Synchronous Serial Busy bit (SSBSY in SSCS) should be cleared to prevent any spurious data transmission when SSPE is set.

9.3.1. Master Configuration

The SPI operates in master mode when the master bit (SSMSTR in SSCON) is set.

9.3.1.1. Slave Configuration

The SPI operates in slave mode when the master bit (SSMSTR in SSCON) is cleared.

The Synchronous Serial Output Enable (SSOE in SSCS) allows to use P1.4 as SS#. In this case, P1.4 cannot be used for another purpose when the peripheral is enabled. Otherwise MISO is enabled as soon as SSPE is set and P1.4 may be used without restriction.

When SSOE= 0, SS# indicates the beginning and the end of each byte transfer. Hence P1.4 usage is mandatory in this mode. When SSOE= 1, the beginning of a byte transfer is indicated by the first transition of SCK to its active state and its end is indicated by the eighth transition of SCK to its idle state. Hence SS# may remain active and P1.4 usage is optional if the TSC80251G1D is the only slave.

9.3.2. Data Exchange

There are two possible policies to exchange data in master and slave modes:

- polling
- interrupts

9.3.2.1. Master Mode with Polling Policy

Initialization

1. Set the SSMOD bit to select the SSLC in SPI mode.
2. Disable the SSLC interrupts (see chapter 12. “Interrupt System”).
3. Set the SSMSTR bit to select the master mode.
4. Select a bit rate by programming SSBRS and SSCR1:0 bits according to Table 9.1 and SSBR register as required.
5. Select a transfer format by programming SSCPOL and SSCPHA according to Figure 9.5 and Figure 9.6 and depending on the slave.
6. Clear the SSBSY flag to avoid spurious transfer when SPI is enabled.
7. Set the SSPE bit to enable the SPI interface.

Transfer

1. Assert SS# (as required).
2. Write the transmit data to SSDAT (or jump to next step if there no data to transmit).
3. Set SSBSY in SSCS to start exchange.
4. Poll SSBSY flag until it has been cleared by hardware (end of transfer).
5. Read the received data in SSDAT (or jump to next step if there no data to receive).
6. Deassert SS# (as required) and jump to step 1 if exchange is not completed.
7. Data exchange completed.

This policy provides the fastest effective transmission and is well adapted when communicating at high speed with other Microcontrollers. However, the procedure may then be interrupted at any time by higher priority tasks.

9.3.2.2. Master Mode with Interrupt Policy

Initialization

1. Set the SSMOD bit to select the SSLC in SPI mode.
2. Set the SSMSTR bit to select the master mode.
3. Select a bit rate by programming SSBRS and SSCR1:0 bits according to Table 9.1 and SSBR register as required.
4. Select a transfer format by programming SSCPOL and SSCPHA according to Figure 9.5 and Figure 9.6 and depending on the slave.
5. Enable the SSLC interrupts (see chapter 12. “Interrupt System”).
6. Clear the SSBSY flag to avoid spurious transfer when SPI is enabled.
7. Set the SSPE bit to enable the SPI interface.

Start exchange

1. Ensure no data exchange is on–going by checking SSBSY is not set.
2. Assert SS# (as required).
3. Write the transmit data in SSDAT (or jump to next step if there no data to transmit).
4. Set SSBSY to start transfer.

Interrupt service routine

1. Read the received data in SSDAT (or jump to next step if there is no data to receive).
2. Deassert SS# (as required).
3. Write the new transmit data to SSDAT (or jump to next step if there is no data to transmit).
4. Assert SS# (to reselect the slave if required).
5. Clear SSI in SSCON.

6. Set SSBSY to start new transfer (or jump to next step if exchange is completed)
7. Return from interrupt service routine

This policy may be effective when communicating with slow devices.

When the SPI is configured in master mode, SSBR, SSCON, SSCS and SSDAT may be read at any time while a transmission is on–going (i.e. SSBSY is set). Conversely, SSBR, SSCON and SSCS may be written at any time while a transmission is on–going. However, special care should be taken when writing to them:

- Do not change SSBR if SSBRS is set
- Do not change SSCR0 or SSCR1 if SSBRS is cleared
- Do not change SSPHA or SSPOL
- Clearing SSPE would immediately disable the peripheral
- Do not change SSMOD or SSBRS
- Clearing SSBSY would immediately complete the data shifting

Furthermore, as there is no write protection of the shift register, SSDAT should not be written while a transmission is on–going.

9.3.2.3. Slave Mode with Polling Policy

Initialization

1. Set the SSMOD bit to select the SSLC in SPI mode.
2. Disable the SSLC interrupts (see chapter 12. “Interrupt System”).
3. Clear the SSMSTR bit to select the slave mode.
4. Clear SSI, SSOVR and SSERR.
5. Select a transfer format by programming SSCPOL and SSCPHA according to Figure 9.5 and Figure 9.6 and depending on the master.
6. Write a transmit data to SSDAT (or jump to next step if there no data to transmit).
7. Set SSPE bit to enable the SPI interface.

Transfer

1. Poll SSI flag until it has been set by hardware (end of transfer).
2. Read the received data in SSDAT (or jump to next step if there no data to receive).
3. Write a next transmit data to SSDAT (or jump to next step if there no data to transmit).
4. Clear SSI flag.
5. Check SSOVR and SSERR to verify data validity and jump to step 1.

This policy provides the fastest effective transmission and is well adapted when communicating at high speed with other Microcontrollers. However, some data may be lost if the procedure is interrupted by higher priority tasks.

9.3.2.4. Slave Mode with Interrupt Policy

Initialization

1. Set the SSMOD bit to select the SSLC in SPI mode.
2. Clear the SSMSTR bit to select the slave mode.
3. Clear SSI, SSOVR and SSERR.
4. Select a transfer format by programming SSCPOL and SSCPHA according to Figure 9.5 and Figure 9.6 and depending on the slave.
5. Enable the SSLC interrupts (see chapter 12. “Interrupt System”).
6. Write a transmit data to SSDAT (or jump to next step if there no data to transmit).
7. Set the SSPE bit to enable the SPI interface.

Interrupt service routine

1. Read the received data in SSDAT (or jump to step 5 if there is no data to receive).
2. Write the next transmit data to SSDAT (or jump to step 6 if there is no data to transmit)
3. Clear SSI flag to acknowledge interrupt.
4. Check SSOVR and SSERR to verify data validity.
5. Return from interrupt service routine.

This policy may be effective when communicating with slow devices. Then it may be executed at high priority level preventing burst activity on the SPI bus.

When the SPI is configured in slave mode, SSBR, SSCON, SSCS and SSDAT may be read at any time while a transmission is on–going (i.e. SSBSY is set). Conversely, SSBR, SSCON and SSCS may be written at any time while a transmission is on–going. However, special care should be taken when writing to them:

- Do not change SSMSTR, SSPHA or SSPOL
- Clearing SSPE would immediately disable the peripheral
- Do not change SSOE
- Writing SSDAT will cause an overrun error

9.3.3. Error Conditions

The following flags signal SPI slave error conditions.

- Slave Overflow (SSOVR in SSCON)
- Slave fault error (SSERR in SSCS)

9.3.3.1. Slave Overflow

The synchronous slave overflow flag in the synchronous serial control register (SSCS) is set to inform user the data is corrupted when the two following conditions occur:

1. SSDAT register is written while a transfer is on–going (SSBSY set).
2. A shift occurs while SSI is set (data not yet read from SSDAT or data not yet written to SSDAT).

SSOVR may be checked by software to guarantee the data received. SSOVR will remain set until cleared by software, however subsequent transfers will not be affected if it is not reset.

9.3.3.2. Slave Fault Error

The synchronous slave error flag (SSERR) in the synchronous serial status and control register (SSCS) is set to inform user the data is corrupted when the two following conditions occur:

1. SS# is deasserted or SSOE bit is cleared before the end of a transfer.
2. SSOE bit is cleared before the end of a transfer.

In these two cases, the on–going 8–bit shift will be terminated without delay and irrespective of the MOSI and SCK inputs. Then SSI is set and SSBSY is cleared as for any completed shift operation.

SSERR may be checked by software to guarantee the data received. SSERR will remain set until cleared by software, however subsequent transfers will not be affected if it is not reset.

9.4. Registers

SSBR (S:92h)

Synchronous Serial Bit Rate Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Synchronous Serial Bit Rate Data** Bit rate is given by the formula: Br= F_{OSC} / (4 · (SSBR value + 3)), Br in KHz (F_{OSC} in MHz).

Reset Value= 0000 0000b

Figure 9.7 SSBR Register

SSCON (S:93h) (SSLC μWire/SPI)

Synchronous Serial Control Register

7	6	5	4	3	2	1	0
SSOVR	**SSPE**	**SSCPOL**	**SSCPHA**	**SSI**	**SSMSTR**	**SSCR1**	**SSCR0**

Bit Number	Bit Mnemonic	Description
7	SSOVR	**Synchronous Serial Slave Overrun flag** Set by hardware in slave mode when a shift occurs while SSI is set or when SSDAT is written while SSBSY is set. Clear by software to reset the overflow flag. Cannot be set by software.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Clear to disable the SPI interface. Set to enable the SPI interface.
5	SSCPOL	**Synchronous Serial clock Polarity bit** Clear to have the clock output set to 0 in idle state. Set to have the clock output set to 1 in idle state. *Note: When the peripheral is disabled, the clock output is 1.*
4	SSCPHA	**Synchronous Serial Clock Phase bit** Clear to have the data sampled when the clock leave the idle state (see SSCPOL). Set to have the data sampled when the clock return to idle state (see SSCPOL).
3	SSI	**Synchronous Serial Interrupt flag** Set by hardware when an 8–bit shift is completed. Must be cleared by software to acknowledge interrupt.
2	SSMSTR	**Synchronous Serial Master bit** Clear to configure the peripheral in slave mode. Set to configure the peripheral in master mode.
1	SSCR1	**Synchronous Serial Control Rate bit 1** see Table 9.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** see Table 9.1.

Reset Value= 0000 0100b

Note:
After reset SSMOD bit is cleared so that I^2C mode is selected. In this case reading SSCON register will return the I^2C SSCON reset value which is 00h. To read the SPI SSCON value, first set SSMOD bit in SSCS to select the SPI mode.

Figure 9.8 SSCON Register

SSCS (S:94h) read/write (SSLC µWire/SPI)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSBRS	–	SSERR	SSBSY	–	–	SSOE	SSMOD

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Synchronous Serial Bit Rate Selection bit** Clear to select the bit rate controlled by SSCR1 and SSCR0. Set to select the programmable bit rate generator.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	SSERR	**Synchronous Serial Slave Error Flag** Set by hardware when SS# is deasserted or SSOE is cleared before the end of a receiving data. Clear by software to reset error flag.
4	SSBSY	**Synchronous Serial Busy bit** Slave Mode Cleared by hardware when one byte shift is completed (then SSI is set). Set by hardware when one byte exchange begins. Master Mode Cleared by hardware when one byte shift is completed (then SSI is set). Clear to abort the transmission before it is completed (then SSI is not set). Set to start the transmission.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	SSOE	**Synchronous Serial Output Enable bit** Clear in slave mode to have the MISO output enabled by SS# pin (P1.4). Set in slave mode to have the MISO output enabled regardless of P1.4. ***Note:*** *this bit has no effect in master mode.*
0	SSMOD	**Synchronous Serial Mode selection bit** Set to select the SSLC in µWire/SPI mode.

Reset Value= 0X00 XX00b

Note:
After reset SSMOD bit is cleared so that I^{2}C mode is selected. In this case reading SSCS register will read the I^{2}C SSCS reset value which is F8h. To read the SPI SSCS value, first set SSMOD bit in SSCS to select the SPI mode.

Figure 9.9 SSCS Register

SSDAT (S:95h) (SSLC µWire/SPI)

Synchronous Serial Data Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Synchronous Serial Data.**

Reset Value= 0000 0000b

Note:
After reset SSMOD bit is cleared so that I^{2}C mode is selected. In this case reading SSDAT register will read the I^{2}C SSDAT reset value which is 00h. To read the SPI SSDAT value, first set SSMOD bit in SSCS to select the SPI mode.

Figure 9.10 SSDAT Register

10. Hardware Watchdog Timer

10.1. Introduction

The TSC80251G1D derivatives contain a dedicated hardware Watchdog Timer (WDT) that automatically resets the chip if it is allowed to time out. The WDT provides a means of recovering from routines that do not complete successfully due to software or hardware malfunctions. The WDT described in this chapter is not associated with the PCA Watchdog Timer (see chapter 7. “Event and Waveform Controller”), which may be disabled by software and is less reliable.

10.2. Description

The WDT is a 14–bit counter that counts peripheral cycles, i.e. the system clock divided by twelve (see Figure 10.1).

The Hardware Watchdog Timer register (WDTRST, see Figure 10.2) provides control access to the WDT. Two operations control the WDT:

- Chip reset clears and disables the WDT.
- Writing a specific two–byte sequence (1Eh–E1h) to WDTRST register clears and enables the WDT.

If it is not cleared, the WDT overflows on count 3FFFh +1 and forces a chip reset. Table 10.1 shows the time–out period depending on oscillator frequency.

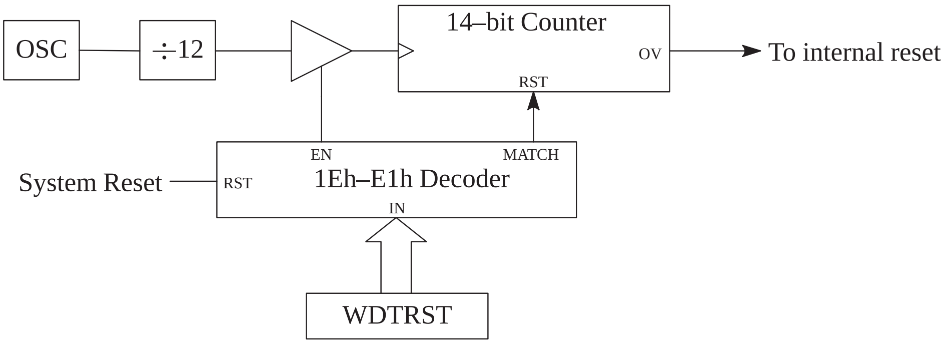


Figure 10.1 WDT Block Diagram

Table 10.1 WDT Time–Out Period

Frequency	Time–out period
12 MHz	16.4 ms
16 MHz	12.3 ms
24 MHz	8.2 ms

WDTRST is a write–only register. Attempts to read it return FFh. The WDT itself is not read or write accessible. The WDT does not drive the external RST pin.

10.3. Using the Hardware WDT

To recover from software malfunctions, the user should control the WDT as follows:

- Following chip reset, write the two–byte sequence 1Eh–E1h to WDTRST register to enable the WDT. Then the WDT begins counting from 0.
- Repeatedly for the duration of program execution, write the two–byte sequence 1Eh–E1h to WDTRST register to clear and enable the WDT before it overflows. The WDT starts over at 0.
  If the WDT overflows, it initiates a chip reset. Chip reset clears the WDT and disables it.

10.4. Hardware WDT during Idle and Power–Down Modes

Operation of the WDT during power reduction modes deserves special attention.
The WDT continues to count while the TSC80251G1D is in Idle mode. This means that the user must dedicate some internal or external hardware to service the WDT during Idle mode. One approach is to use a peripheral Timer to generate an interrupt request when the Timer overflows. The interrupt service routine then clears the WDT, reloads the peripheral Timer for the next service period and puts the TSC80251G1D back into Idle mode.

The Power–Down mode stops all phase clocks. This causes the WDT to stop counting and to hold its count. The WDT resumes counting from where it left off if the Power–Down mode is terminated by INT0#, INT1# or keyboard interrupt. To ensure that the WDT does not overflow shortly after exiting the Power–Down mode, clear the WDT just before entering Power–down mode. The WDT is cleared and disabled if the Power–Down mode is terminated by a reset.

10.5. Registers

WDTRST (S:A6h) write

Hardware Watchdog Timer Reset Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Watchdog Control Value.**

Reset Value= 1111 1111b

Figure 10.2 WDTRST Register

11. Power Monitoring and Management

11.1. Introduction

The power monitoring and management can be used to supervise the Power Supply (VDD) and to start up properly when the TSC80251G1D is powered up.

It consists of the features listed below and explained hereafter:

- Power–On reset
- Power–Fail reset
- Power–Off flag
- Clock prescaler
- Idle mode
- Power–Down mode

All these features are controlled by four 8–bit registers, the Power Management register (POWM), the Power Filter register (PFILT), the Power Control register (PCON) and the Clock Reload register (CKRL) detailed at the end of this chapter.

11.2. Power–On Reset

When the power supply is under V_{RET}, the digital parts of the circuit are not working properly. Then the I/O ports are controlled by the external Reset pin (RST). In order to keep them in a predictable state (see Table 11.1), RST pin must either be driven to a high level for the power rise duration or tied to VDD through an external capacitor. However, the internal oscillator starts when VDD reaches V_{RST+}.

When the power supply rises above V_{RET} and as long as it stays below V_{RST+}, the Power–OFF Flag (POF, see paragraph 11.4.) and the Reset Detection control bit (RSTD, see Figure 11.7) are set and the internal reset begins.

When the power supply rises above V_{RST+}, the internal reset completes after both RST pin has gone low and 64 clock periods on XTAL1 have occurred. This ensures the external oscillator has stabilized. If an external capacitor is connected to RST, it is charged through an internal pull–down resistor R_{RST} which determines the minimal reset period according to the capacitor value. Reducing VDD quickly to 0 V causes the RST pin voltage to momentarily fall below 0 V. This voltage is internally limited and does not harm the device.

Table 11.1 Pin Conditions in Special Operating Modes

Mode	Program Memory	ALE pin	PSEN# pin	Port 0 pins	Port 1 pins	Port 2 pins	Port 3 pins
Reset	Don’t care	Weak High	Weak High	Floating	Weak High	Weak High	Weak High
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Floating	Data	Data	Data
Power–Down	Internal	0	0	Data	Data	Data	Data
Power–Down	External	0	0	Floating	Data	Data	Data

11.3. Power–Fail Detector

The Power–Fail detector is controlled by RSTD bit in POWM register. When enabled, the power supply is continuously monitored and an internal reset is generated(1) if VDD goes below V_{RST-}(2) for at least $2xPFILTxT_{OSC}$. The Power Filter register (PFILT, see Figure 11.6) must be programmed by the user with an 8–bit value depending on the time constant he wants.

If the power supply rises again over V_{RST+}(2), the internal reset completes after 64 oscillator clock periods.

If RSTD is set, the power supply monitoring is disabled. To avoid extra consumption and allows VDD reduction to V_{RET} in Power–Down mode, the power supply monitoring is also disabled in this mode (PD= 1).

Note:
1. *The internal reset is not propagated on the RST pin.*
2. *Refer to "TSC8051G1D Datasheet" for DC characteristics.*

Caution:
When VDD is reduced to V_{RET} in Power–Down mode the VDD voltage is no more monitored. In this case, RAM content may be damage if VDD goes below V_{RET} and circuit behavior is unpredictable unless an external reset is applied.

11.4. Power–Off Flag

When the power is turned off or fails, the data retention is not guaranteed. A Power–Off Flag (POF, see Figure 11.5) allows to detect this condition. POF is set by hardware during a reset which follows a power–up or a power–fail. This is a cold reset. A warm reset is an external or a watchdog reset without power failure, hence which preserves the internal memory content and POF. To use POF, test and clear this bit just after reset. Then it will be set only after a cold reset.

Note:
When power supply monitoring is disabled (RSTD= 1 or in Power–Down mode), POF information is not delivered with the same accuracy. It is recommended to clear and not to take in account the POF value after exit from a power down mode with VDD reduction.

11.5. Clock Prescaler

In order to optimize the power consumption and the execution time needed for a specific task, an internal clock prescaler feature has been implemented to program the system clock frequency. It is possible to work at full speed for all tasks requiring quick response time at low frequency for background tasks which do not need CPU power but power consumption optimizing. Figure 11.1 shows the diagram of the on–chip oscillator where the clock programming block clearly appears. The CPU clock can be programmed via 8–bit CKRL register and by setting CKSRC bit in POWM register. When CKSRC bit in POWM register is cleared the oscillator frequency F_{OSC}= F_{XTAL} . When CKSRC bit in POWM register is set, the oscillator frequency is given by the following formula:

$$F_{OSC} = \frac{F_{XTAL}}{2(CKRL + 1)}$$

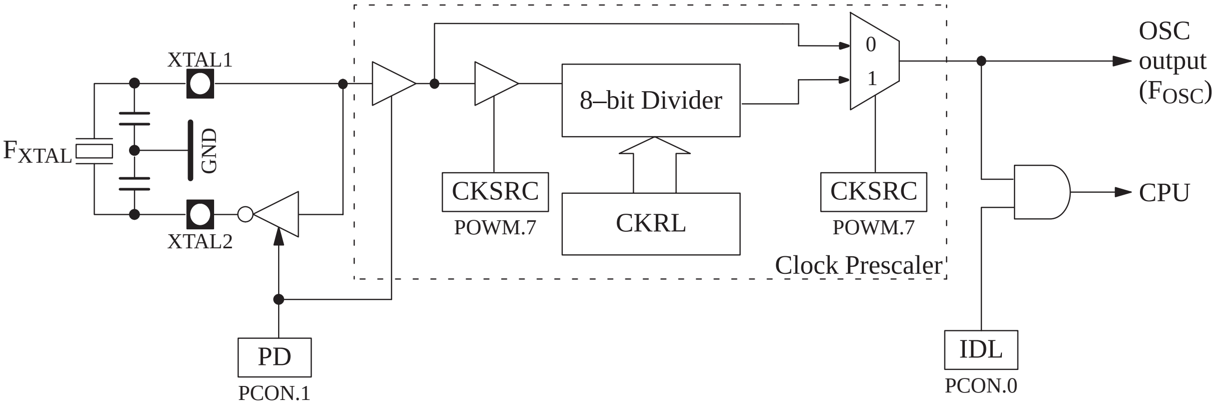


Figure 11.1 On–Chip Oscillator Block Diagram

In all this document, the on–chip oscillator is used to be symbolized by Figure 11.2. Please notice that all the peripherals share the same clock. Special care should be taken when changing it to prevent any peripheral operating failure.

Figure 11.2 On–Chip Oscillator Symbol

11.6. Idle Mode

Idle mode is a power reduction mode that reduces the power consumption to about 40% of the typical running power consumption. In this mode, program execution halts. Idle mode freezes the clock to the CPU at known states while the peripherals continue to be clocked (see Figure 11.1). The CPU status before entering Idle mode is preserved, i.e., the program counter, program status word register, and register file retain their data for the duration of Idle mode. The contents of the SFRs and RAM are also retained. It also pulls ALE and PSEN# high. The status of the Port pins depends upon the location of the program memory:

- Internal program memory:
  The ALE and PSEN# pins are pulled high the Ports 0, 1, 2 and 3 pins are reading data (see Table 11.1).
- External program memory:
  The ALE and PSEN# pins are pulled high; the Port 0 pins are floating and the Ports 1, 2 and 3 pins are reading data (see Table 11.1).

11.6.1. Entering Idle Mode

To enter Idle mode, set IDL bit in PCON register. The TSC80251G1D enters Idle mode upon execution of the instruction that sets IDL bit. The instruction that sets IDL bit is the last instruction executed.

Caution*:*
If IDL bit and PD bit are set simultaneously, the TSC80251G1D enters Power–Down mode. Then it does not go in Idle mode when exiting Power–Down mode.

11.6.2. Exiting Idle Mode

There are two ways to exit Idle mode:

- Generate an enabled interrupt
  Hardware clears IDL bit in PCON register which restores the clock to the CPU. Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Idle mode. The general purpose flags (GF1 and GF0 in PCON register) may be used to indicate whether an interrupt occurred during normal operation or during Idle mode. When Idle mode is exited by an interrupt, the interrupt service routine may examine GF1 and GF0.
- Reset the chip
  A logic high on the RST pin clears IDL bit in PCON register directly and asynchronously. This restores the clock to the CPU. Program execution momentarily resumes with the instruction immediately following the instruction that activated the Idle mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the TSC80251G1D and vectors the CPU to address FF:0000h.

Note:
During the time that execution resumes, the internal RAM cannot be accessed; however, it is possible for the Port pins to be accessed. To avoid unexpected outputs at the Port pins, the instruction immediately following the instruction that activated Idle mode should not write to a Port pin or to the external RAM.

11.6.3. Recovering from Idle Mode

To enable the recovering from Idle mode, set RPD bit in PCON register (beware that this bit sets also the recovering from Power–Down mode, see paragraph 11.7.3.). Then a disabled external interrupt clears IDL bit in PCON register which restores the clock to CPU. Execution continues with the instruction immediately following the instruction that activated Idle mode.

Note:
When RPD bit in PCON register is set, it is still possible to exit Idle mode using an enabled interrupt (see paragraph 11.6.2.).

11.7. Power–Down Mode

The Power–Down mode places the TSC80251G1D in a very low power state. Power–Down mode stops the oscillator, freezes all clock at known states (see Figure 11.1) and disables the Power–Fail reset. The CPU status prior to entering Power–Down mode is preserved, i.e., the program counter, program status word register, and register file retain their data for the duration of Power–Down mode. In addition, the SFRs and RAM contents are preserved. The status of the Port pins depends on the location of the program memory:

- Internal program memory:
  The ALE and PSEN# pins are pulled low, the Ports 0, 1, 2 and 3 pins are reading data (see Table 11.1).
- External program memory:
  The ALE and PSEN# pins are pulled low; the Port 0 pins are floating and the Ports 1, 2 and 3 pins are reading data (see Table 11.1).

Note:
VDD may be reduced to as low as V_{RET} during Power–Down mode to further reduce power dissipation. Take care, however, that VDD is not reduced until Power–Down mode is invoked.

Caution:
As soon as Power–Down mode is entered (PD bit set) the VDD voltage is no more monitored (RSTD bit set) (see paragraphs).

11.7.1. Entering Power–Down Mode

To enter Power–Down mode, set PD bit in PCON register. The TSC80251G1D enters the Power–Down mode upon execution of the instruction that sets PD bit. The instruction that sets PD bit is the last instruction executed.

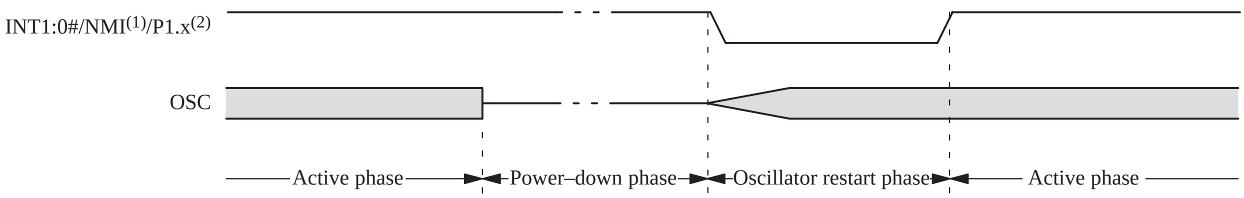
11.7.2. Exiting Power–Down Mode

Caution:
If VDD was reduced during the Power–Down mode, do not exit Power–Down mode until VDD is restored to the normal operating level.

There are two ways to exit the Power–Down mode:

- Generate an enabled external interrupt.
  TSC80251G1D provides capability to exit from Power–Down using INT0#, INT1#, NMI and P1.x (trough keyboard interrupt) inputs. In addition, using P1.x input provides high or low level exit capability (see paragraph 12.4. “Keyboard Interface”).
  Hardware clears PD bit in PCON register which starts the oscillator and restores the clocks to the CPU and peripherals. Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power–Down mode.

Notes:
The external interrupt used to exit Power–Down mode must be configured as level sensitive (INT0# and INT1#) and must be assigned the highest priority. In addition, the duration of the interrupt must be long enough to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted (see Figure 11.3).
Exit from power–down by external interrupt does not affect the SFRs nor the internal RAM content.



Notes:
1. NMI is high level triggered.
2. P1.x can be high or low level triggered.

Figure 11.3 Power–Down Exit Waveform

- Generate a reset.
  A logic high on the RST pin clears PD bit in PCON register directly and asynchronously. This starts the oscillator and restores the clock to the CPU and peripherals. Program execution momentarily resumes with the instruction immediately following the instruction that activated Power–Down mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the TSC80251G1D and vectors the CPU to address FF:0000h.

Notes:
During the time that execution resumes, the internal RAM cannot be accessed; however, it is possible for the Port pins to be accessed. To avoid unexpected outputs at the Port pins, the instruction immediately following the instruction that activated the Power–Down mode should not write to a Port pin or to the external RAM.
Exit from power–down by reset redefines all the SFRs, but does not affect the internal RAM content.

11.7.3. Recovering from Power–Down Mode

To enable the recovering from Power–Down mode, set RPD bit in PCON register (beware that this bit sets also the recovering from Idle mode, see paragraph 11.6.3.). Then a disabled external interrupt clears PD bit in PCON register which starts the oscillator and restores the clock to CPU and peripherals. Execution continues with the instruction immediately following the instruction that activated Power–Down mode.

Notes:
The external interrupt used to exit Power–Down mode must be configured as level sensitive (INT0# and INT1#) and must be assigned the highest priority. In addition, the duration of the interrupt must be long enough to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted (see Figure 11.3).
When RPD bit in PCON register is set, it is still possible to exit Power–Down mode using an enabled external interrupt (see paragraph 11.7.2.).

11.8. Registers

CKRL (S:8Eh)

Clock Reload Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Prescaler Value.**

Reset Value= 0000 1000b

Figure 11.4 CKRL Register

PCON (S:87h)

Power configuration Register

7	6	5	4	3	2	1	0
SMOD1	**SMOD0**	**RPD**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Double Baud Rate bit** Set to double the Baud Rate when Timer 1 is used and mode 1, 2 or 3 is selected in SCON register.
6	SMOD0	**SCON Select bit** When cleared, read/write accesses to SCON.7 are to SM0 bit and read/write accesses to SCON.6 are to SM1 bit. When set, read/write accesses to SCON.7 are to FE bit and read/write accesses to SCON.6 are to OVR bit. SCON is Serial Port Control register.
5	RPD	**Recover from Idle/Power–Down bit** Clear to disable the Recover from Idle and Power–Down modes feature. Set to enable the Recover from Idle and Power–Down modes feature.
4	POF	**Power–Off flag** Set by hardware when VDD rises above V_{RET+} to indicate that the Power Supply has been set off. Must be cleared by software.
3	GF1	**General Purpose flag 1** One use is to indicate wether an interrupt occurred during normal operation or during Idle mode.
2	GF0	**General Purpose flag 0** One use is to indicate wether an interrupt occurred during normal operation or during Idle mode.
1	PD	**Power–Down Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power–Down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence.

Reset Value= 0000 0000b

Figure 11.5 PCON Register

PFILT (S:86h)

Power Filter Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**PFILT Value.**

Reset Value= XXXX XXXXb

Figure 11.6 PFILT Register

POWM (S:8Fh)

Power Management Register

7	6	5	4	3	2	1	0
CKSRC	–	–	–	RSTD	–	–	–

Bit Number	Bit Mnemonic	Description
7	CKSRC	**Clock Source bit** Cleared by hardware after a Power-Up. In that case: F_{OSC}= F_{XTAL}. Set to enable the clock. In that case: F_{OSC}= F_{XTAL} / 2 · (CKRL + 1).
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	RSTD	**Reset Detector Disable bit** Clear to enable the Power–Fail detector. Set to disable the Power–Fail detector.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value= 0XXX 0XXXb

Figure 11.7 POWM Register

12. Interrupt System

12.1. Introduction

The TSC80251G1D, like other control–oriented computer architectures, employs a program interrupt method. This operation branches to a subroutine and performs some service in response to the interrupt. When the subroutine completes, execution resumes at the point where the interrupt occurred. Interrupts may occur as a result of internal TSC80251G1D activity (e.g., Timer overflow) or at the initiation of electrical signals external to the microcontroller (e.g., Serial Port communication). In all cases, interrupt operation is programmed by the system designer, who determines priority of interrupt service relative to normal code execution and other interrupt service routines. Nine of the eleven interrupts are enabled or disabled by the system designer and may be manipulated dynamically.

A typical interrupt event chain occurs as follows:

- An internal or external device initiates an interrupt–request signal.
- This signal, connected to an input pin and periodically sampled by the TSC80251G1D, latches the event into a flag buffer.
- The priority of the flag is compared to the priority of other interrupts by the interrupt handler. A high priority causes the handler to set an interrupt flag.
- This signals the instruction execution unit to execute a context switch. This context switch breaks the current flow of instruction sequences. The execution unit completes the current instruction prior to a save of the program counter (PC) and reloads the PC with the start address of a software service routine.
- The software service routine executes assigned tasks and as a final activity performs a RETI (return from interrupt) instruction. This instruction signals completion of the interrupt, resets the interrupt–in–progress priority and reloads the program counter. Program operation then continues from the original point of interruption.

Table 12.1 Interrupt System Signals

Mnemonic	Type	Description	Multiplexed with
INT0#	I	**External Interrupt 0** This input sets IE0 bit in TCON register. If IT0 bit in TCON register is set, IE0 bit is controlled by a negative edge trigger on INT0#. If IT0 bit in TCON register is cleared, IE0 bit is controlled by a low level trigger on INT0#.	P3.2
INT1#	I	**External Interrupt 1** This input sets IE1 bit in TCON register. If IT1 bit in TCON register is set, IE1 bit is controlled by a negative edge trigger on INT1#. If IT1 bit in TCON register is cleared, IE1 bit is controlled by a low level trigger on INT1#.	P3.3
NMI	I	**Non Maskable Input**	–
P1.X	I	**Keyboard Interrupt Inputs** see paragraph 12.4.	–

The TSC80251G1D has one software interrupt –TRAP instruction (always enabled)– and ten hardware interrupt sources constituted by one non maskable source NMI and nine peripheral maskable interrupt sources: two external (INT0# and INT1#), one for Timer 0, one for Timer 1, one for Timer 2, one for Serial Port, one for Event and Waveform Controller, one for Synchronous Serial Link Controller, one for Keyboard.

Note:
The Non Maskable Interrupt input is the second highest priority interrupt after the TRAP. It is always enabled and can not be disabled by software like others interrupts. NMI is active when a high level is applied on its input during a minimum of 24 oscillators clock periods.

Six interrupt registers are used to control the interrupt system. Two 8–bit registers are used to enable separately the interrupt sources: IE0 and IE1 (see Figure 12.4 and Figure 12.5).

Four 8–bit registers are used to establish the priority level of the nine sources: IPL0, IPH0, IPL1 and IPH1 (see Figure 12.6 to Figure 12.9).

12.2. Interrupt System Priorities

Each of the nine interrupt sources on the TSC80251G1D may be individually programmed to one of four priority levels. This is accomplished by one bit in the Interrupt Priority High registers (IPH0 or IPH1, see Figure 12.6 and Figure 12.7) and one in the Interrupt Priority Low registers (IPL0 or IPL1, see Figure 12.8 and Figure 12.9) This provides each interrupt source four possible priority levels select bits (see Table 12.2).

Table 12.2 Priority levels

IPHxx	IPLxx	Priority Level
0	0	0 Lowest
0	1	1
1	0	2
1	1	3 Highest

A low–priority interrupt is always interrupted by a higher priority interrupt but not by another interrupt of lower or equal priority. Higher priority interrupts are serviced before lower priority interrupts. The response to simultaneous occurrence of equal priority interrupts (i.e. sampled within the same four state interrupt cycle) is determined by a hardware priority–within–level resolver (see Table 12.3). The interrupt control system is shown in Figure 12.1.

Table 12.3 Interrupt Priority Within Level

Interrupt Name	Priority Number	Interrupt Address Vectors	Interrupt request flag cleared by hardware (H) or by software (S)
TRAP	1 Highest Priority Not interruptible	FF:007Bh	–
NMI	2	FF:003Bh	–
INT0#	3	FF:0003h	H if edge, S if level
Timer 0	4	FF:000Bh	H
INT1#	5	FF:0013h	H if edge, S if level
Timer 1	6	FF:001Bh	H
Serial Port	7	FF:0023h	S
Timer 2	8	FF:002Bh	S
EWC	9	FF:0033h	S
Keyboard	10	FF:0043h	S
Reserved	11	FF:004Bh	–
Reserved	12	FF:0053h	–
Reserved	13	FF:005Bh	–
Reserved	14	FF:0063h	–
SSLC	15	FF:006Bh	S
Reserved	16 Lowest Priority	FF:0073h	–

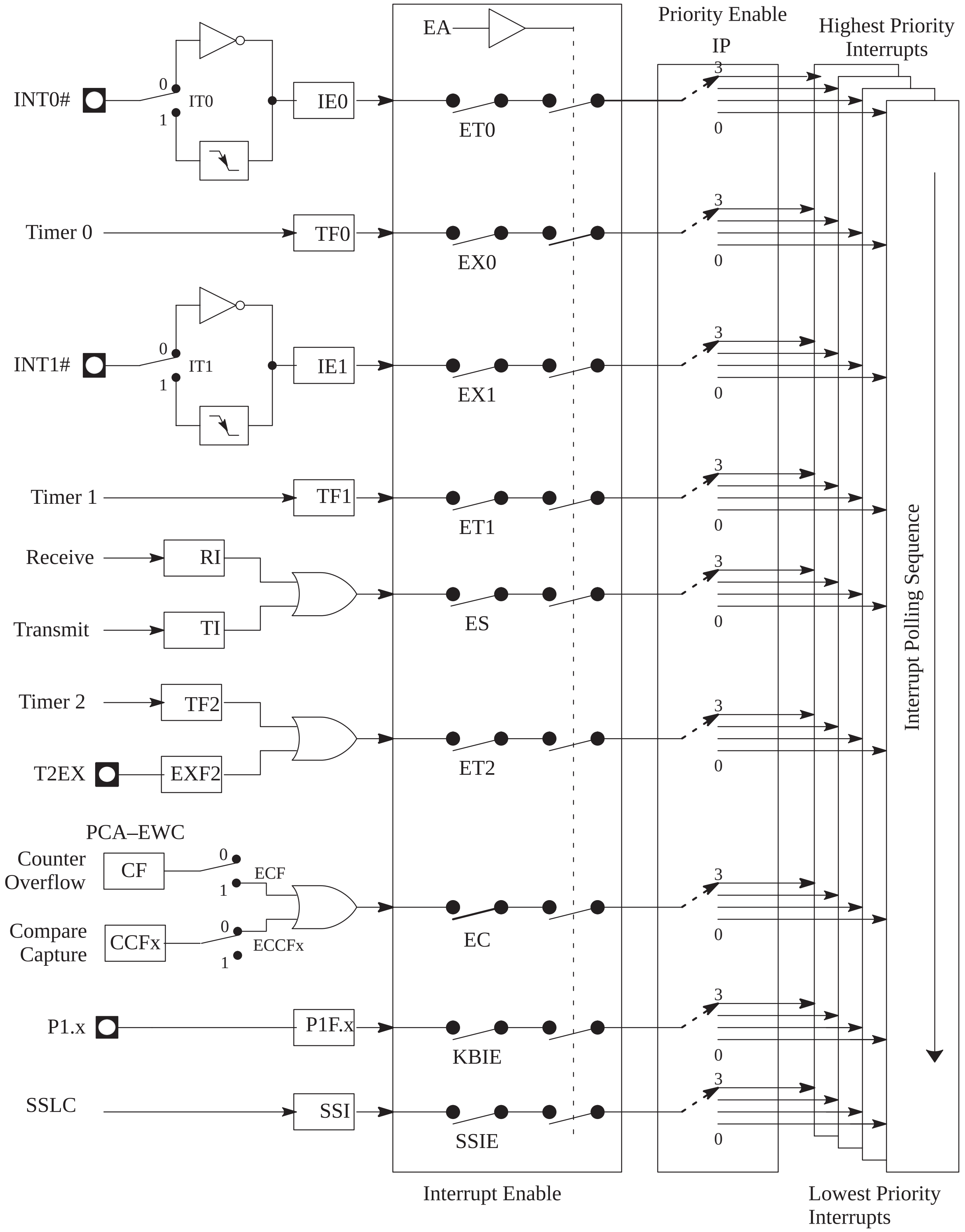


Figure 12.1 Interrupt Control System

12.3. External Interrupts

12.3.1. INT0# and INT1#

External interrupts INT0# and INT1# (INTn#, n= 0 or 1) pins may each be programmed to be level–triggered or edge–triggered, dependent upon bits IT0 and IT1 (ITn, n= 0 or 1) in TCON register. If ITn= 0, INTn# is triggered by a low level at the pin. If ITn= 1, INTn# is negative–edge triggered. External interrupts are enabled with bits EX0 and EX1 (EXn, n= 0 or 1) in IE0 register. Events on INTn# set the interrupt request flag IEn in TCON register. A request flag is cleared by hardware vectors to service routines only if the interrupt is edge triggered. If the interrupt is level–triggered, the interrupt service routine must clear the request flag and the interrupt must be deasserted before the end of the interrupt service routine (> 5 state times). External interrupt pins must be deasserted for at least four state times prior to a request.

INT0# and INT1# inputs provide both the capability to exit from Power–Down mode on low level signals. See paragraph 11.7.2. “Exiting Power–Down Mode”.

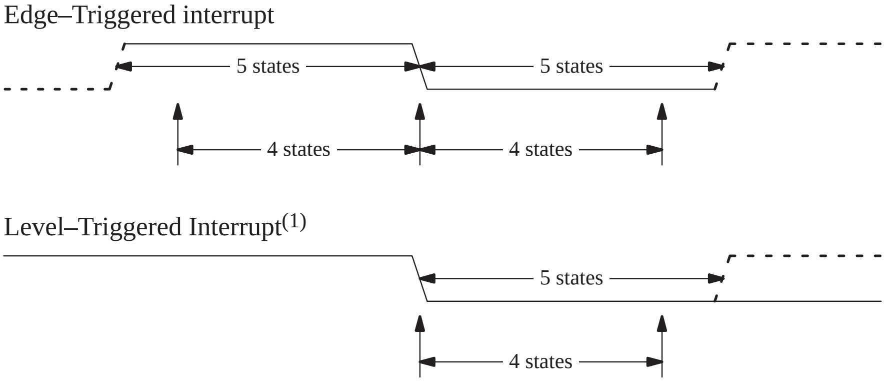
12.3.2. NMI

NMI input is the non maskable interrupt input. Since NMI is high level–triggered input, NMI signal must be deasserted before the end of the interrupt service routine (> 5 state times). External interrupt pins must be deasserted for at least four state times prior to a request.

NMI input provides the capability to exit from Power–Down mode on high level signal. See paragraph 11.7.2. “Exiting Power–Down Mode”.

12.3.3. Input Sampling

External interrupt pins are sampled once every four state times (see Figure 12.2). A level–triggered interrupt pin held low or high for five–state time period guarantees detection. Edge–triggered external interrupts must hold the request pin low for at least five state times. This ensures edge recognition and sets interrupt request bit EXn. The CPU clears EXn automatically during service routine fetch cycles for edge–triggered interrupts.



Note:
The Non Maskable Interrupt input is high level triggered.

Figure 12.2 Minimum Pulse Timings

12.4. Keyboard Interface

Port 1 has some on–chip provisions to interface more easily a keyboard matrix.

Each Port line may be connected to a Keyboard output and has the possibility to detect a programmable level (see Figure 12.3). Port lines are sampled once every state, then a level must be maintained during two states to be recognized (a frame length of 250 ns at 16 MHz).

- The level to detect (high or low) on a Port line is selected by the corresponding Port 1 Level Selection bit in P1LS register (see Figure 12.12).
- The detection of the programmed level sets the corresponding flag in P1F register (see Figure 12.10).
- If the corresponding Port 1 Interrupt Enable bit in P1IE register (see Figure 12.11) is set, the flag setting generates the Keyboard interrupt request.
  But it must be enabled by the KBIE bit in IE1 register (see Figure 12.5).
- The way to exit interrupt service routine is to wait for a level change on the corresponding Port line or to program an interrupt request on the complemented level. Then, the corresponding flag must be cleared by software.

Through the keyboard interface, port 1 provides the capability to exit from Power–Down mode on both low level or high level signals. See paragraph 11.7.2. "Exiting Power–Down Mode".

Note:
The keyboard interface is normally used for level detection, but it may be used in other ways since any pulse is stored in P1F.

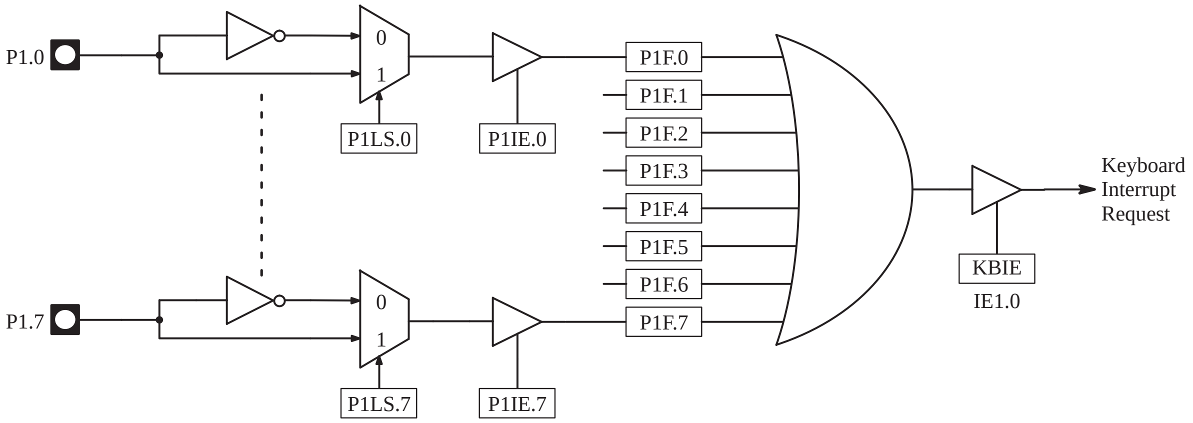


Figure 12.3 Keyboard Interface Interrupt Structure

12.5. Registers

IE0 (S:A8h)

Interrupt Enable Register 0

7	6	5	4	3	2	1	0
EA	EC	ET2	ES	ET1	EX1	ET0	EX0

Bit Number	Bit Mnemonic	Description
7	EA	**Global Interrupt Enable** Clear to disable all interrupts, except the TRAP and NMI interrupts which are always enabled. Set to enable all interrupts that are individually enabled in IE0 and IE1 registers.
6	EC	**EWC–PCA Interrupt Enable** Set to enable the the PCA interrupt.
5	ET2	**Timer 2 Overflow Interrupt Enable** Set this bit to enable the timer 2 overflow interrupt.
4	ES	**Serial I/O Port Interrupt Enable** Set this bit to enable the serial I/O port interrupt.
3	ET1	**Timer 1 Overflow Interrupt Enable** Set this bit to enable Timer 1 overflow interrupt.
2	EX1	**External Interrupt 1 Enable** Set this bit to enable external interrupt 1.
1	ET0	**Timer 0 Overflow Interrupt Enable** Set this bit to enable the Timer 0 overflow interrupt.
0	EX0	**External Interrupt 0 Enable** Set this bit to enable external interrupt 0.

Reset Value= 0000 0000b

Figure 12.4 IE0 Register

IE1 (S:B1h)

Interrupt Enable Register

7	6	5	4	3	2	1	0
–	–	SSIE	–	–	–	–	KBIE

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	SSIE	**SSLC Interrupt Enable bit** Clear to disable the SSLC interrupt. Set to enable the SSLC interrupt.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	KBIE	**Keyboard Interrupt Enable bit** Clear to disable the Keyboard interrupt. Set to enable the Keyboard interrupt.

Reset Value= XX0X XXX0b

Figure 12.5 IE1 Register

IPH0 (S:B7h)

Interrupt Priority High Register 0

7	6	5	4	3	2	1	0
–	IPHC	IPHT2	IPHS	IPHT1	IPHX1	IPHT0	IPHX0

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPHC	**EWC–PCA Counter Interrupt Priority level most significant bit** IPHC IPLC Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
5	IPHT2	**Timr Interrupt Priority level most significant bit** IPHT2 IPLT2 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
4	IPHS	**Serial Port Interrupt Priority level most significant bit** IPHS IPLS Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
3	IPHT1	**Timer 1 Interrupt Priority level most significant bit** IPHT1 IPLT1 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
2	IPHX1	**External Interrupt 1 Priority level most significant bit** IPHX1 IPLX1 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
1	IPHT0	**Timer 0 Interrupt Priority level most significant bit** IPHT0 IPLT0 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
0	IPHX0	**External Interrupt 0 Priority level most significant bit** IPHX0 IPLX0 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority

Reset Value= X000 000b

Figure 12.6 IPH0 Register

IPH1 (S:B3h)

Interrupt Priority High Register 1

7	6	5	4	3	2	1	0
–	–	IPHSS	–	–	–	–	IPHKB

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	IPHSS	**SSLC Interrupt Priority level most significant bit** IPHSS IPLSS Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IPHKB	**Keyboard Interrupt Priority level most significant bit** IPHKB IPLKB Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority

Reset Value= XX0X XXX0b

Figure 12.7 IPH1 Register

IPL0 (S:B8h)

Interrupt Priority Low Register 0

7	6	5	4	3	2	1	0
–	IPLC	IPLT2	IPLS	IPLT1	IPLX1	IPLT0	IPLX0

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPLC	**EWC–PCA Counter Interrupt Priority level less significant bit** Refer to IPHC for priority level.
5	IPLT2	**Timer 2 Interrupt Priority level less significant bit** Refer to IPHT2 for priority level.
4	IPLS	**Serial Port Interrupt Priority level less significant bit** Refer to IPHS for priority level.
3	IPLT1	**Timer 1 Interrupt Priority level less significant bit** Refer to IPHT1 for priority level.
2	IPLX1	**External Interrupt 1 Priority level less significant bit** Refer to IPHX1 for priority level.
1	IPLT0	**Timer 0 Interrupt Priority level less significant bit** Refer to IPHT0 for priority level.
0	IPLX0	**External Interrupt 0 Priority level less significant bit** Refer to IPHX0 for priority level.

Reset Value= X000 0000b

Figure 12.8 IPL0 Register

IPL1 (S:B2h)

Interrupt Priority Low Register 1

7	6	5	4	3	2	1	0
–	–	IPLSS	–	–	–	–	IPLKB

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	IPLSS	**SSLC Interrupt Priority level less significant bit** Refer to IPHSS for priority level.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IPLKB	**Keyboard Interrupt Priority level less significant bit.** Refer to IPHKB for priority level.

Reset Value= XX0X XXX0b

Figure 12.9 IPL1 Register

P1F (S:9Eh)

Port 1 Flag Register

7	6	5	4	3	2	1	0
P1F.7	**P1F.6**	**P1F.5**	**P1F.4**	**P1F.3**	**P1F.2**	**P1F.1**	**P1F.0**

Bit Number	Bit Mnemonic	Description
7	P1F.7	**Port 1 line 7 flag** Set by hardware when the Port line 7 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.7 bit in P1IE register is set. Must be cleared by software.
6	P1F.6	**Port 1 line 6 flag** Set by hardware when the Port line 6 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.6 bit in P1IE register is set. Must be cleared by software.
5	P1F.5	**Port 1 line 5 flag** Set by hardware when the Port line 5 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.5 bit in P1IE register is set. Must be cleared by software.
4	P1F.4	**Port 1 line 4 flag** Set by hardware when the Port line 4 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.4 bit in P1IE register is set. Must be cleared by software.
3	P1F.3	**Port 1 line 3 flag** Set by hardware when the Port line 3 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.3 bit in P1IE register is set. Must be cleared by software.
2	P1F.2	**Port 1 line 2 flag** Set by hardware when the Port line 2 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.2 bit in P1IE register is set. Must be cleared by software.
1	P1F.1	**Port 1 line 1 flag** Set by hardware when the Port line 1 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.1 bit in P1IE register is set. Must be cleared by software.
0	P1F.0	**Port 1 line 0 flag** Set by hardware when the Port line 0 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.0 bit in P1IE register is set. Must be cleared by software.

Reset Value= 0000 0000b

Figure 12.10 P1F Register

P1IE (S:9Dh)

Port 1 Input Interrupt Enable Register

7	6	5	4	3	2	1	0
P1IE.7	**P1IE.6**	**P1IE.5**	**P1IE.4**	**P1IE.3**	**P1IE.2**	**P1IE.1**	**P1IE.0**

Bit Number	Bit Mnemonic	Description
7	P1IE.7	**Port 1 line 7 Interrupt Enable bit** Clear to disable P1F.7 bit in P1F register to generate an interrupt request. Set to enable P1F.7 bit in P1F register to generate an interrupt request.
6	P1IE.6	**Port 1 line 6 Interrupt Enable bit** Clear to disable P1F.6 bit in P1F register to generate an interrupt request. Set to enable P1F.6 bit in P1F register to generate an interrupt request.
5	P1IE.5	**Port 1 line 5 Interrupt Enable bit** Clear to disable P1F.5 bit in P1F register to generate an interrupt request. Set to enable P1F.5 bit in P1F register to generate an interrupt request.
4	P1IE.4	**Port 1 line 4 Interrupt Enable bit** Clear to disable P1F.4 bit in P1F register to generate an interrupt request. Set to enable P1F.4 bit in P1F register to generate an interrupt request.
3	P1IE.3	**Port 1 line 3 Interrupt Enable bit** Clear to disable P1F.3 bit in P1F register to generate an interrupt request. Set to enable P1F.3 bit in P1F register to generate an interrupt request.
2	P1IE.2	**Port 1 line 2 Interrupt Enable bit** Clear to disable P1F.2 bit in P1F register to generate an interrupt request. Set to enable P1F.2 bit in P1F register to generate an interrupt request.
1	P1IE.1	**Port 1 line 1 Interrupt Enable bit** Clear to disable P1F.1 bit in P1F register to generate an interrupt request. Set to enable P1F.1 bit in P1F register to generate an interrupt request.
0	P1IE.0	**Port 1 line 0 Interrupt Enable bit** Clear to disable P1F.0 bit in P1F register to generate an interrupt request. Set to enable P1F.0 bit in P1F register to generate an interrupt request.

Reset Value= 0000 0000b

Figure 12.11 P1IE Register

P1LS (S:9Ch)

Port 1 Level Selector Register

7	6	5	4	3	2	1	0
P1LS.7	P1LS.6	P1LS.5	P1LS.4	P1LS.3	P1LS.2	P1LS.1	P1LS.0

Bit Number	Bit Mnemonic	Description
7	P1LS.7	**Port 1 line 7 Level Selection bit** Clear to enable a low level detection on Port line 7. Set to enable a high level detection on Port line 7.
6	P1LS.6	**Port 1 line 6 Level Selection bit** Clear to enable a low level detection on Port line 6. Set to enable a high level detection on Port line 6.
5	P1LS.5	**Port 1 line 5 Level Selection bit** Clear to enable a low level detection on Port line 5. Set to enable a high level detection on Port line 5.
4	P1LS.4	**Port 1 line 4 Level Selection bit** Clear to enable a low level detection on Port line 4. Set to enable a high level detection on Port line 4.
3	P1LS.3	**Port 1 line 3 Level Selection bit** Clear to enable a low level detection on Port line 3. Set to enable a high level detection on Port line 3.
2	P1LS.2	**Port 1 line 2 Level Selection bit** Clear to enable a low level detection on Port line 2. Set to enable a high level detection on Port line 2.
1	P1LS.1	**Port 1 line 1 Level Selection bit** Clear to enable a low level detection on Port line 1. Set to enable a high level detection on Port line 1.
0	P1LS.0	**Port 1 line 0 Level Selection bit** Clear to enable a low level detection on Port line 0. Set to enable a high level detection on Port line 0.

Reset Value= 0000 0000b

Figure 12.12 P1LS Register

Appendix A

Signal Descriptions

Table of Contents

1. Signal Descriptions

This chapter provides reference information for the external signals of the TSC80251G1D. Pin assignments are shown in Figure 1.1 (DIP package), Figure 1.2 (PLCC package), and Figure 1.3 (QFP package). Pin numbers are referred to in Table 1.1.

Table 1.2 describes each of the signals. It lists the signal type (input, output, power, or ground) and the alternative functions of multifunction pins. Table 1.3 shows how configuration bits RD1:0 (referred to in Table 1.2) configure the A17, A16. RD#, WR# and PSEN# pins for external memory accesses.

1.1. Pinout

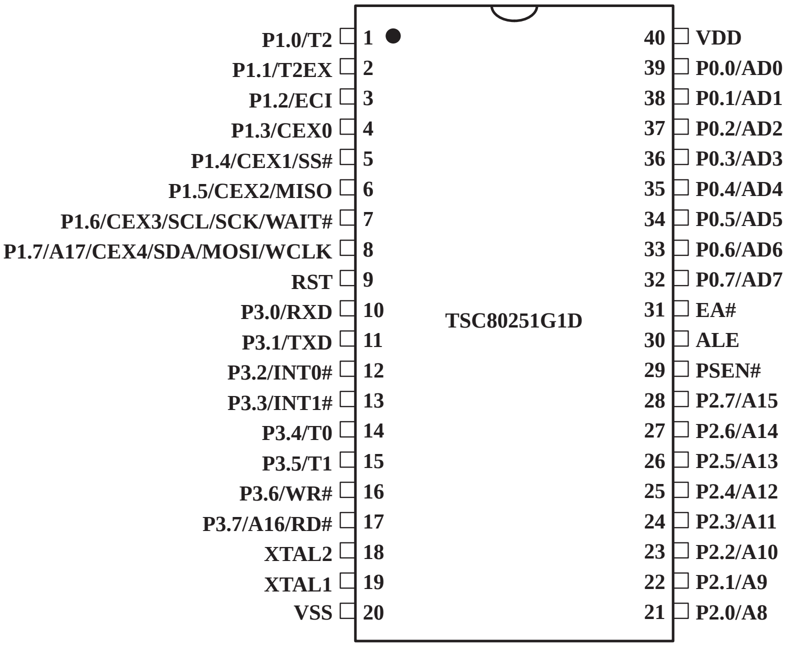


Figure 1.1 TSC80251G1D 40–pin DIP package

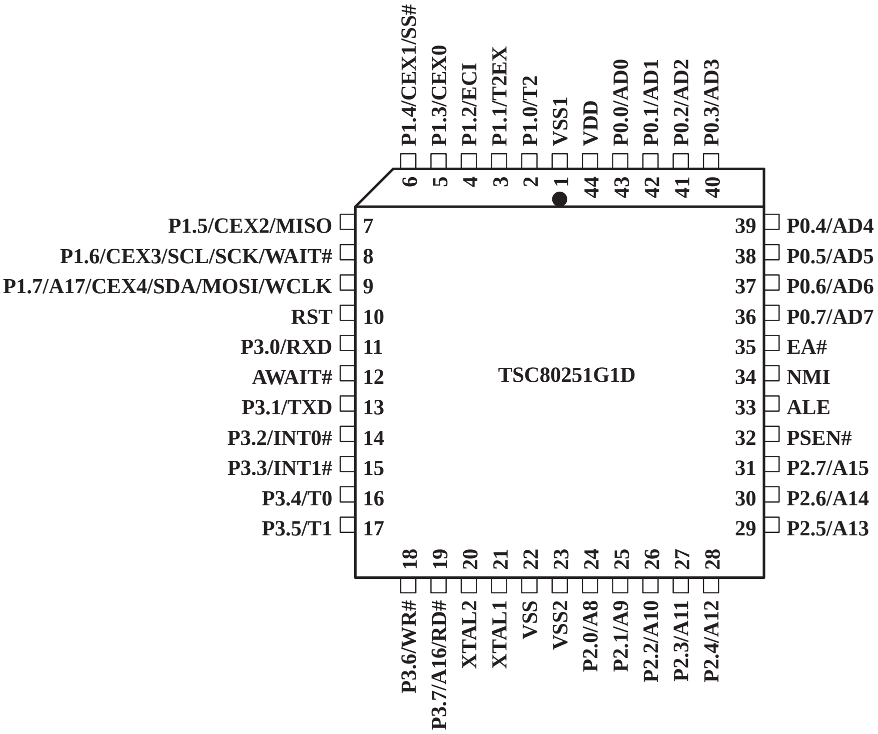


Figure 1.2 TSC80251G1D 44–pin PLCC Package

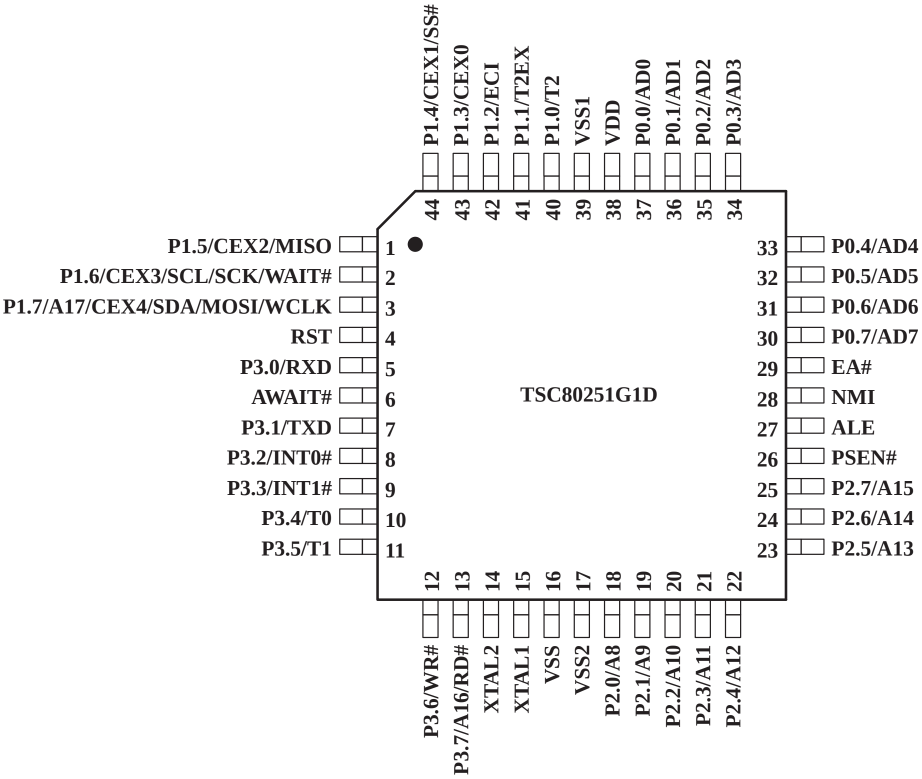


Figure 1.3 TSC80251G1D 44–pin VQFP Package

Table 1.1 TSC80251G1D Pin Assignment

DIP	PLCC	VQFP	Name	DIP	PLCC	VQFP	Name
	1	39	VSS1		23	17	VSS2
1	2	40	P1.0/T2	21	24	18	P2.0/A8
2	3	41	P1.1/T2EX	22	25	19	P2.1/A9
3	4	42	P1.2/ECI	23	26	20	P2.2/A10
4	5	43	P1.3/CEX0	24	27	21	P2.3/A11
5	6	44	P1.4/CEX1/SS#	25	28	22	P2.4/A12
6	7	1	P1.5/CEX2/MISO	26	29	23	P2.5/A13
7	8	2	P1.6/CEX3/SCL/SCK/WAIT#	27	30	24	P2.6/A14
8	9	3	P1.7/A17/CEX4/SDA/MOSI/WCLK	28	31	25	P2.7/A15
9	10	4	RST	29	32	26	PSEN#
10	11	5	P3.0/RXD	30	33	27	ALE
	12	6	AWAIT#		34	28	NMI
11	13	7	P3.1/TXD	31	35	29	EA#
12	14	8	P3.2/INT0#	32	36	30	P0.7/AD7
13	15	9	P3.3/INT1#	33	37	31	P0.6/AD6
14	16	10	P3.4/T0	34	38	32	P0.5/AD5
15	17	11	P3.5/T1	35	39	33	P0.4/AD4
16	18	12	P3.6/WR#	36	40	34	P0.3/AD3
17	19	13	P3.7/A16/RD#	37	41	35	P0.2/AD2
18	20	14	XTAL2	38	42	36	P0.1/AD1
19	21	15	XTAL1	39	43	37	P0.0/AD0
20	22	16	VSS	40	44	38	VDD

1.2. Signals

Table 1.2 TSC80251G1D Signal Descriptions

Signal Name	Type	Description	Alternate Function
A17	O	**18th Address Bit** Output to memory as 18th external address bit (A17) in extended bus applications, depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see Table 1.3).	P1.7
A16	O	**17th Address Bit** Output to memory as 17th external address bit (A16) in extended bus applications, depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see Table 1.3).	P3.7
A15:8(1)	O	**Address Lines** Upper address lines for the external bus.	P2.7:0
AD7:0(1)	I/O	**Address/Data Lines** Multiplexed lower address lines and data for the external memory.	P0.7:0
ALE	O	**Address Latch Enable** ALE signals the start of an external bus cycle and indicates that valid address information are available onlines A16/A17 and A7:0. An external latch can use ALE to demultiplex the address from address/databus.	
AWAIT#	I	**Real–time Asynchronous Wait States Input** When this pin is active (low level), the memory cycle is stretched until it becomes high. When using the TSC80251G1D as a pin–for–pin replacement for a 8xC51 product, AWAIT# can be unconnected without loss of compatibility or power consumption increase (on–chip pull–up).	

Signal Name	Type	Description	Alternate Function
CEX4:0	O	**PCA Input/Output pins** CEXx are input signals for the PCA capture mode and output signals for the PCA compare and PWM modes.	P1.7:3
EA#	I	**External Access Enable** EA# directs program memory accesses to on–chip or off–chip code memory. For EA#= 0, all program memory accesses are off-chip. For EA#= 1, an access is on-chip ROM if the address is within the range of the on–chip ROM; otherwise the access is off-chip. The value of EA# is latched at reset. For devices without ROM on-chip, EA# must be strapped to ground.	
ECI	O	**PCA External Clock input** ECI is the external clock input to the 16–bit PCA timer.	P1.2
MISO	I/O	**SPI Master Input Slave Output line** When SPI is in master mode, MISO receives data from the slave peripheral. When SPI is in slave mode, MISO outputs data to the master controller.	P1.5
MOSI	I/O	**SPI Master Output Slave Input line** When SPI is in master mode, MOSI outputs data to the slave peripheral. When SPI is in slave mode, MOSI receives data from the master controller.	P1.7
INT1:0#	I	**External Interrupts 0 and 1.** INT1#/INT0# inputs set IE1:0 in the TCON register. If bits IT1:0 in the TCON register are set, bits IE1:0 are set by a falling edge on INT1#/INT0#. If bits IT1:0 are cleared, bits IE1:0 are set by a low level on INT1#/INT0#	P3.3:2
NMI	I	**Non Maskable Interrupt** Holding this pin high for 24 oscillator periods triggers an interrupt. When using the TSC80251G1D as a pin–for–pin replacement for a 8xC51 product, NMI can be unconnected without loss of compatibility or power consumption increase (on–chip pull–down).	
P0.0:7	I/O	**Port 0** P0 is an 8–bit open–drain bidirectional I/O port.	AD7:0
P1.0:7	I/O	**Port 1** P1 is an 8–bit bidirectional I/O port with internal pull–ups. P1 provides interrupt capability for a keyboard interface.	
P2.0:7	I/O	**Port 2** P2 is an 8–bit bidirectional I/O port with internal pull–ups.	A15:8
P3.0:7	I/O	**Port 3** P3 is an 8–bit bidirectional I/O port with internal pull–ups.	
PSEN#	O	**Program Store Enable/Read signal output** PSEN# is asserted for a memory address range that depends on bits RD0 and RD1 in UCONFIG0 byte (see Table 1.3).	
RD#	O	**Read or 17th Address Bit (A16)** Read signal output to external data memory depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see Table 1.3).	P3.7
RST	I	**Reset input to the chip** Holding this pin high for 64 oscillator periods while the oscillator is running resets the device. The Port pins are driven to their reset conditions when a voltage greater than V_{IH1} is applied, whether or not the oscillator is running. This pin has an internal pull-down resistor which allows the device to be reset by connecting a capacitor between this pin and VDD. Asserting RST when the chip is in Idle mode or Power–Down mode returns the chip to normal operation.	
RXD	I/O	**Receive Serial Data** RXD sends and receives data in serial I/O mode 0 and receives data in serial modes I/O 1, 2 and 3.	P3.0

Signal Name	Type	Description	Alternate Function
SCL	I/O	**I^{2}C Serial Clock** When I^{2}C controller is in master mode, SCL outputs the serial clock to slave peripherals. When I^{2}C controller is in slave mode, SCL receives clock from the master controller.	P1.6
SCK	I/O	**SPI Serial Clock** When SPI is in master mode, SCK outputs clock to the slave peripheral. When SPI is in slave mode, SCK receives clock from the master controller.	P1.6
SDA	I/O	**I^{2}C Serial Data** SDA is the bidirectional I^{2}C data line.	P1.7
SS#	I	**SPI Slave Select Input** When in Slave mode, SS# enables the slave mode.	P1.4
T1:0	I/O	**Timer 1:0 External Clock Inputs** When timer 1:0 operates as a counter, a falling edge on the T1:0 pin increments the count.	
T2	I/O	**Timer 2 Clock Input/Output** For the timer 2 capture mode, T2 is the external clock input. For the Timer 2 clock–out mode, T2 is the clock output.	P1.0
T2EX	I	**Timer 2 External Input** In timer 2 capture mode, a falling edge initiates a capture of the timer 2 registers. In auto–reload mode, a falling edge causes the timer 2 register to be reloaded. In the up–down counter mode, this signal determines the count direction: 1= up, 0= down.	P1.1
TXD	I/O	**Transmit Serial Data** TXD outputs the shift clock in serial I/O mode 0 and transmits data in serial I/O modes 1, 2 and 3.	P3.1
VDD	PWR	**Digital Supply Voltage** Connect this pin to +5V or +3V supply voltage.	
VSS	GND	**Circuit Ground** Connect this pin to ground.	
VSS1	GND	**Secondary Ground 1** This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the TSC80251G1D as a pin–for–pin replacement for a 8xC51 product, VSS1 can be unconnected without loss of compatibility.	
VSS2	GND	**Secondary Ground 2** This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the TSC80251G1D as a pin–for–pin replacement for a 8xC51 product, VSS2 can be unconnected without loss of compatibility.	
WAIT#	I	**Real–time Synchronous Wait States Input** The real–time WAIT# input is enabled by setting RTWE bit in WCON (S:A7h). During bus cycles, the external memory system can signal ‘system ready’ to the microcontroller in real time by controlling the WAIT# input signal.	P1.6
WCLK	O	**Wait Clock Output** The real–time WCLK output is enabled by setting RTWCE bit in WCON (S:A7h). When enabled, the WCLK output produces a square wave signal with a period of one half the oscillator frequency.	P1.7
WR#	I	**Write** Write signal output to external memory. Asserted for the memory address range specified by configuration byte UCONFIG0, bits RD1:0 (see Table 1.3).	P3.6

Signal Name	Type	Description	Alternate Function
XTAL1	I	**Input to the on–chip inverting oscillator amplifier** To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, its output is connected to this pin. XTAL1 is the clock source for internal timing.	
XTAL2	O	**Output of the on–chip inverting oscillator amplifier** To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, leave XTAL2 unconnected.	

Note:

1. *The description of A15:8/P2.7:0 and AD7:0/P0.7:0 are for the non–page mode chip configuration. If the chip is configured in page mode operation, port 0 carries the lower address bits (A7:0) while port 2 carries the upper address bits (A15:8) and the data (D7:0).*

Table 1.3 Memory Signal Selections (RD1:0)[1]

RD1	RD0	P1.7	P3.7/RD#	PSEN#	WR#	Features
0	0	A17	A16	Asserted for reads to all memory locations	Asserted for writes to all memory locations	256 Kbytes external memory
0	1	I/O pin	A16	Asserted for reads to all memory locations	Asserted for writes to all memory locations	128 Kbytes external memory
1	0	I/O pin	I/O pin	Asserted for reads to all memory locations	Asserted for writes to all memory locations	64 Kbytes external memory
1	1	I/O pin	Asserted for addresses ≤ 7F:FFFFH	Asserted for addresses ≥ 80:0000H	Asserted for writes to 80C51 microcontroller data memory locations	64 Kbytes external memory. Compatible with 80C51 microcontrollers

Note:

1. *RD1:0 are bits 3:2 of configuration byte UCONFIG0 (See Figure 2.7 in the design section).*

Appendix B

Registers

Table of Contents

1. Registers

1.1. Introduction

This chapter is a reference source of information for the TSC80251G1D special function registers (SFRs) and the register file. The SFR map in Table 1.1 provides the address and reset value for each SFR. Table 1.2 through Table 1.10 list the SFRs by functional category. Table 1.11 lists the registers that make up the register file. The remainder of the chapter contains descriptions of the SFRs arranged in alphabetical order.

Note:
Use the prefix "S:" with SFR addresses to distinguish them from other addresses.

1.2. SFR map

Table 1.1 SFR Addresses and Reset Values

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h		CH 0000 0000	CCAP0H 0000 0000	CCAP1H 0000 0000	CCAP2H 0000 0000	CCAP3H 0000 0000	CCAP4H 0000 0000		FFh
F0h	B(1) 0000 0000								F7h
E8h		CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000	CCAP2L 0000 0000	CCAP3L 0000 0000	CCAP4L 0000 0000		EFh
E0h	ACC(1) 0000 0000								E7h
D8h	CCON 00X0 0000	CMOD 00XX X000	CCAPM0 X000 0000	CCAPM1 X000 0000	CCAPM2 X000 0000	CCAPM3 X000 0000	CCAPM4 X000 0000		DFh
D0h	PSW(1) 0000 0000	PSW1(1) 0000 0000							D7h
C8h	T2CON 0000 0000	T2MOD XXXX XX00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000			CFh
C0h									C7h
B8h	IPL0 X000 0000	SADEN 0000 0000					SPH(1) 0000 0000		BFh
B0h	P3 1111 1111	IE1 XX0X XXX0	IPL1 XX0X XXX0	IPH1 XX0X XXX0				IPH0 X000 0000	B7h
A8h	IE0 0000 0000	SADDR 0000 0000							AFh
A0h	P2 1111 1111						WDTRST 1111 1111	WCON XXXX XX00	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	P1LS 0000 0000	P1IE 0000 0000	P1F 0000 0000		9Fh
90h	P1 1111 1111		SSBR 0000 0000	SSCON (2)	SSCS (3)	SSDAT 0000 0000	SSADR 0000 0000		97h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	CKRL 0000 1000	POWM 0XXX 0XXX	8Fh
80h	P0 1111 1111	SP 0000 0111	DPL(1) 0000 0000	DPH(1) 0000 0000	DPXL(1) 0000 0001		PFILT XXXX XXXX	PCON 0000 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

▨ reserved

Notes:
1. *These registers are described in the TSC80251 Programmer's Guide (C251 core registers).*
2. *In I^2C and SPI modes, SSCON is splitted in two separate registers. SSCON reset value is 0000 0000 in I^2C mode and 0000 0100 in SPI mode.*
2. *In read and write modes, SSCS is splitted in two separate registers. SSCS reset value is 1111 1000 in read mode and 0000 0000 in write mode.*

1.3. Registers list

Table 1.2 C251 Core SFRs

Mnemonic	Name	Address
ACC[1]	Accumulator	S:E0h
B[1]	B Register	S:F0h
PSW	Program Status Word	S:D0h
PSW1	Program Status Word 1	S:D1h
SP[1]	Stack Pointer – LSB of SPX	S:81h
SPH[1]	Stack Pointer High – MSB of SPX	S:BEh
DPL[1]	Data Pointer Low byte – LSB of DPTR	S:82h
PH[1]	Data Pointer High byte – MSB of DPTR	S:83h
DPXL[1]	Data Pointer, Extended Low byte of DPX	S:84h

Note:
1. These SFRs can also be accessed by their corresponding registers in the register file.

Table 1.3 I/O Port SFRs

Mnemonic	Name	Address
P 0	Port 0	S:80h
P 1	Port 1	S:90h
P 2	Port 2	S:A0h
P 3	Port 3	S:B0h

Table 1.4 Timer/Counter and WatchDog Timer SFRs

Mnemonic	Name	Address
TL0	Timer/Counter 0 Low Byte	S:8Ah
TH0	Timer/Counter 0 High Byte	S:8Ch
TL1	Timer/Counter 1 Low Byte	S:8Bh
TH1	Timer/Counter 1 High Byte	S:8Dh
TL2	Timer/Counter 2 Low Byte	S:CCh
TH2	Timer/Counter 2 High Byte	S:CDh
TCON	Timer/Counter 0 and 1 Control	S:88h
TMOD	Timer/Counter 0 and 1 Mode Control	S:89h
T2CON	Timer/Counter 2 Control	S:C8h
T2MOD	Timer/Counter 2 Mode Control	S:C9h
RCAP2L	Timer 2 Reload/Capture Low Byte	S:CAh
RCAP2H	Timer 2 Reload/Capture High Byte	S:CBh
WDTRST	WatchDog Timer Reset	S:A6h

Table 1.5 Serial I/O Port SFRs

Mnemonic	Name	Address
SCON	Serial Control	S:98h
SBUF	Serial Data Buffer	S:99h
SADEN	Slave Address Mask	S:B9h
SADDR	Slave Address	S:A9h
BRL	Baud Rate Reload	S:9Ah
BDRCON	Baud Rate Control	S:9Bh

Table 1.6 SSLC SFRs

Mnemonic	Name	Address
SSCON	Synchronous Serial control	S:93h
SSDAT	Synchronous Serial Data	S:95h
SSCS	Synchronous Serial Control and Status	S:94h
SSADR	Synchronous Serial Address	S:96h
SSBR	Synchronous Serial Bit Rate	S:92h

Table 1.7 Event Waveform Control/Programmable Counter Array SFRs

Mnemonic	Name	Address
CCON	EWC–PCA Timer/Counter Control	S:D8h
CMOD	EWC–PCA Timer/Counter Mode	S:D9h
CCAPM0	EWC–PCA Timer/Counter Mode 0	S:DAh
CCAPM1	EWC–PCA Timer/Counter Mode 1	S:DBh
CCAPM2	EWC–PCA Timer/Counter Mode 2	S:DCh
CCAPM3	EWC–PCA Timer/Counter Mode 3	S:DDh
CCAPM4	EWC–PCA Timer/Counter Mode 4	S:DEh
CL	EWC–PCA Timer/Counter Low register	S:E9h
CH	EWC–PCA Timer/Counter High register	S:F9h
CCAP0L	EWC–PCA Compare Capture Module 0 Low Register	S:EAh
CCAP1L	EWC–PCA Compare Capture Module 1 Low Register	S:EBh
CCAP2L	EWC–PCA Compare Capture Module 2 Low Register	S:ECh
CCPA3L	EWC–PCA Compare Capture Module 3 Low Register	S:EDh
CCPA4L	EWC–PCA Compare Capture Module 4 Low Register	S:EEh
CCPA0H	EWC–PCA Compare Capture Module 0 High Register	S:FAh
CCPA1H	EWC–PCA Compare Capture Module 1 High Register	S:FBh
CCPA2H	EWC–PCA Compare Capture Module 2 High Register	S:FCh
CCPA3H	EWC–PCA Compare Capture Module 3 High Register	S:FDh
CCPA4H	EWC–PCA Compare Capture Module 4 High Register	S:FEh

Table 1.8 System Management SFRs

Mnemonic	Name	Address
PCON	Power Control	S:87h
POWM	Power Management	S:8Fh
PFILT	Power Filter	S:86h
CKRL	Clock Reload	S:8Eh
WCON	Synchronous Real–Time Waite State Control	S:A7h

Table 1.9 Interrupt SFRs

Mnemonic	Name	Address
IE0	Interrupt Enable Control 0	S:A8h
IE1	Interrupt Priority Control 1	S:B1h
IPH0	Interrupt Priority Control High 0	S:B7h
IPL0	Interrupt Priority Control Low 0	S:B8h
IPH1	Interrupt Priority Control High 1	S:B2h
IPL1	Interrupt Priority Control Low 1	S:B3h

Table 1.10 Keyboard Interface SFRs

Mnemonic	Name	Address
P1IE	Port 1 Input Interrupt Enable	S:9Dh
P1F	Port 1 Flag	S:9Eh
P1LS	Port 1 Level Selection	S:9Ch

Table 1.11 Register File

Mnemonic	Name	Address
R0 – R7	Four banks of 8 registers. Selects bank 0–3 with bits (RS0, RS1) of PSW	(1)(2)

R8 – R31	R11= Accumulator (ACC)
	R10= B Register
		(1)(3)

R32 – R55	Reserved	(3)
R56 – R63	DR56= the extended data pointer (DPXL, DPH, DPL) DR60= the extended stack pointer (SPH, SPL)	(1)(3)

Notes:

1. *The registers in the register file are normally accessed by mnemonic. Depending on its location, a register can be addressed as a byte, word, and/or dword.*
2. *The four banks of registers are implemented as the lowest bytes of on–chip RAM and are always accessible via addresses 00:0000H–00:001FH.*
3. *Special function registers ACC, B, DPXL, DPH, DPL, SPH and SPL are located in the register file and can be accessed as R11, R10, DR56 and DR60.*

ACC (S:E0h)

Accumulator

Accumulator ACC provides SFR accesses to the accumulator which resides in the register file as byte register R11. Instruction in the MCS™51 architecture use the accumulator as both source and destination for calculations and moves. Instruction in the MCS™251 architecture assign no special significance to R11. Thes instructions can use byte registers Rm (m= 0–15) interchangeably.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Accumulator data.**

Reset Value= 0000 0000b

B (S:F0h)

B Register

The B register provides SFR access to byte register R10 (also named B) in the register file. The B register is used as both a source or destination in multiply and divide operations. For all other operations, the B register is avalaible for use as one of the byte register Rm (m= 0–15).

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**B Data.**

Reset Value= 0000 0000b

BDRCON (S:9Bh)

Baud Rate Control register

7	6	5	4	3	2	1	0
–	–	–	BRR	TBCK	RBCK	SPD	SRC

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
4	BRR	**Baud Rate Run control bit** Clear to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK	**Transmission Baud Rate Generator Selection bit** Clear to select Timer 1 or Timer 2 as Baud Rate Generator. Set to select Internal Baud Rate Generator.
2	RBCK	**Reception Baud Rate Generator Selection bit** Clear to select Timer 1 or Timer 2 as Baud Rate Generator. Set to select Internal Baud Rate Generator.
1	SPD	**Baud Rate Speed control bit** Clear to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC	**Baud Rate Source select bit in Mode 0** Clear to select F_{OSC}/12 as Baud Rate Generator (fixed transmission clock). Set to select the internal Baud Rate Generator.

Reset Value= XXX0 0000b

BRL (S:9Ah)

Baud Rate Reload Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Baud Rate Data** See Table 6.8 to Table 6.10.

Reset Value= 0000 0000b

CCAP0H (S:FAh)

CCAP1H (S:FBh)

CCAP2H (S:FCh)

CCAP3H (S:FDh)

CCAP4H (S:FEh)

High Byte Compare/Capture Module x Registers (x= 0, 1, 2, 3, 4)

High Byte Compare/Capture Module x

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High byte of EWC–PCA comparison or capture values.**

Reset Value= 0000 0000b

CCAP0L (S:EAh)

CCAP1L (S:EBh)

CCAP2L (S:ECh)

CCAP3L (S:EDh)

CCAP4L (S:EEh)

Low Byte Compare/Capture Module x Registers (x= 0, 1, 2, 3, 4)

Low Byte Compare/Capture Module x

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low byte of EWC–PCA comparison or capture values.**

Reset Value= 0000 0000b

CCAPM0 (S:DAh)

CCAPM1 (S:DBh)

CCAPM2 (S:DCh)

CCAPM3 (S:DDh)

CCAPM4 (S:DEh)

EWC–PCA Compare/Capture Module x Mode registers (x= 0, 1, 2, 3, 4)

7	6	5	4	3	2	1	0
–	ECOMx	CAPPx	CAPNx	MATx	TOGx	PWMx	ECCFx

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
6	ECOMx	**Enable Compare Mode Module x bit** Clear to disable the Compare function. Set to enable the Compare function.. The Compare function is used to implement the software Timer, the high-speed output, the Pulse Width Modulator (PWM) and Watchdog Timer (WDT).
5	CAPPx	**Capture Mode (Positive) Module x bit** Clear to disable the Capture function triggered by a positive edge on CEXx pin. Set to enable the Capture function triggered by a positive edge on CEXx pin
4	CAPNx	**Capture Mode (Negative) Module x bit** Clear to disable the Capture function triggered by a negative edge on CEXx pin. Set to enable the Capture function triggered by a negative edge on CEXx pin.
3	MATx	**Match Module x bit** Set when a match of the PCA Counter with the Compare/Capture register sets CCFx bit in CCON register, flagging an interrupt. Must be cleared by software.
2	TOGx	**Toggle Module x bit** The toggle mode is configured by setting ECOMx, MATx and TOGx bits. Set when a match of the PCA Counter with the Compare/Capture register toggles the CEXx pin. Must be cleared by software.
1	PWMx	**Pulse Width Modulation Module x Mode bit** Set to configure the module x as an 8-bit Pulse Width Modulator with output waveform on CEXx pin . Must be cleared by software.
0	ECCFx	**Enable CCFx Interrupt bit** Clear to disable CCFx bit in CCON register to generate an interrupt request. Set to enable CCFx bit in CCON register to generate an interrupt request.

Reset Value= X000 0000b

CCON (S:D8h)

Timer/Counter Control Register

7	6	5	4	3	2	1	0
CF	**CR**	**–**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

Bit Number	Bit Mnemonic	Description
7	CF	**PCA Timer/Counter Overflow flag** Set by hardware when the PCA Timer/Counter rolls over. This generates a PCA interrupt request if the ECF bit in CMOD register is set. Must be cleared by software.
6	CR	**PCA Timer/Counter Run Control bit** Clear to turn the PCA Timer/Counter off. Set to turn the PCA Timer/Counter on.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	CCF4	**PCA Module 4 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF4 bit in CCAPM4 register is set. Must be cleared by software.
3	CCF3	**PCA Module 3 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF3 bit in CCAPM3 register is set. Must be cleared by software.
2	CCF2	**PCA Module 2 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF2 bit in CCAPM2 register is set. Must be cleared by software.
1	CCF1	**PCA Module 1 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF1 bit in CCAPM1 register is set. Must be cleared by software.
0	CCF0	**PCA Module 0 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF0 bit in CCAPM0 register is set. Must be cleared by software.

Reset Value= 00X0 0000b

CH (S:F9h)

Timer/Counter Registers

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High byte of EWC–PCA Timer/Counter.**

Reset Value= 0000 00000b

CL (S:E9h)

Timer/Counter Registers

Low Byte of Timer/Counter Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low byte of EWC–PCA Timer/Counter.**

Reset Value= 0000 00000b

CKRL (S:8Eh)

Clock Reload Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Prescaler Value.**

Reset Value= 0000 1000b

CMOD (S:D9h)

Timer/Counter Mode Register

7	6	5	4	3	2	1	0
CIDL	**WDTE**	**–**	**–**	**–**	**CPS1**	**CPS0**	**ECF**

Bit Number	Bit Mnemonic	Description
7	CIDL	**PCA Counter Idle Control bit** Clear to let the PCA running during Idle mode. Set to stop the PCA when Idle mode is invoked.
6	WDTE	**Watchdog Timer Enable bit** Clear to disable the Watchdog Timer function on EWC module 4. Set to enable the Watchdog Timer function on EWC module 4.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	CPS1	**EWC Count Pulse Select bits** CPS1 CPS0 Clock source 0 0 Internal Clock, $F_{OSC}/12$ 0 1 Internal Clock, $F_{OSC}/4$ 1 0 Timer 0 overflow 1 1 External clock at ECI/P1.2 pin (Max. Rate= $F_{OSC}/8$)
1	CPS0	
0	ECF	**Enable PCA Counter Overflow Interrupt bit** Clear to disable CF bit in CCON register to generate an interrupt. Set to enable CF bit in CCON register to generate an interrupt.

Reset Value= 00XX X000b

DPH (S:83h)

Data Pointer High

DPH provides SFR acccess to register file location 58 (also named DPH). DPH is the upper byte of the 16–bit data pointer DPTR. Instructions in the MCS™51 architecture use DPTR for data moves, code moves and for jump instructions.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Data Pointer High** Bits 8–15 of the extended data pointer DPX (DR56).

Reset Value= 0000 0000b

DPL (S:82h)

Data Pointer Low

DPL provides SFR acccess to register file location 59 (also named DPL). DPL is the lower byte of the 16–bit data pointer DPTR. Instructions in the MCS™51 architecture use DPTR for data moves, code moves and for jump instructions.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Data Pointer Low** Bits 0–7 of the extended data pointer DPX (DR56).

Reset Value= 0000 0000b

DPXL (S:84h)

Data Pointer Extended Low

DPXL provides SFR acccess to register file location 57 (also named DPXL). DPXL is the lower byte of the upper word of the extended data pointer DPX whose lower word is the 16–bit data–pointer DPTR.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Data Pointer Extended Low** Bits 16–23 of the extended data pointer DPX (DR56).

Reset Value= 0000 0001b

IE0 (S:A8h)

Interrupt Enable Register 0

7	6	5	4	3	2	1	0
EA	EC	ET2	ES	ET1	EX1	ET0	EX0

Bit Number	Bit Mnemonic	Description
7	EA	**Global Interrupt Enable** Clear to disable all interrupts, except the TRAP and NMI interrupts which are always enabled. Set to enable all interrupts that are individually enabled in IE0 and IE1 registers.
6	EC	**EWC–PCA Interrupt Enable** Set to enable the the PCA interrupt.
5	ET2	**Timer 2 Overflow Interrupt Enable** Set this bit to enable the timer 2 overflow interrupt.
4	ES	**Serial I/O Port Interrupt Enable** Set this bit to enable the serial I/O port interrupt.
3	ET1	**Timer 1 Overflow Interrupt Enable** Set this bit to enable Timer 1 overflow interrupt.
2	EX1	**External Interrupt 1 Enable** Set this bit to enable external interrupt 1.
1	ET0	**Timer 0 Overflow Interrupt Enable** Set this bit to enable the Timer 0 overflow interrupt.
0	EX0	**External Interrupt 0 Enable** Set this bit to enable external interrupt 0.

Reset Value= 0000 0000b

IE1 (S:B1h)

Interrupt Enable Register

7	6	5	4	3	2	1	0
–	–	SSIE	–	–	–	–	KBIE

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	SSIE	**SSLC Interrupt Enable bit** Clear to disable the SSLC interrupt. Set to enable the SSLC interrupt.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	KBIE	**Keyboard Interrupt Enable bit** Clear to disable the Keyboard interrupt. Set to enable the Keyboard interrupt.

Reset Value= XX0X XXX0b

IPH0 (S:B7h)

Interrupt Priority High Register 0

7	6	5	4	3	2	1	0
–	IPHC	IPHT2	IPHS	IPHT1	IPHX1	IPHT0	IPHX0

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPHC	**EWC–PCA Counter Interrupt Priority level most significant bit** IPHC IPLC Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
5	IPHT2	**Timr Interrupt Priority level most significant bit** IPHT2 IPLT2 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
4	IPHS	**Serial Port Interrupt Priority level most significant bit** IPHS IPLS Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
3	IPHT1	**Timer 1 Interrupt Priority level most significant bit** IPHT1 IPLT1 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
2	IPHX1	**External Interrupt 1 Priority level most significant bit** IPHX1 IPLX1 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
1	IPHT0	**Timer 0 Interrupt Priority level most significant bit** IPHT0 IPLT0 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
0	IPHX0	**External Interrupt 0 Priority level most significant bit** IPHX0 IPLX0 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority

Reset Value= X000 000b

IPH1 (S:B3h)

Interrupt Priority High Register 1

7	6	5	4	3	2	1	0
–	–	IPHSS	–	–	–	–	IPHKB

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	IPHSS	**SSLC Interrupt Priority level most significant bit** IPHSS IPLSS Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IPHKB	**Keyboard Interrupt Priority level most significant bit** IPHPKB IPLKB Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority

Reset Value= XX0X XXX0b

IPL0 (S:B8h)

Interrupt Priority Low Register 0

7	6	5	4	3	2	1	0
–	IPLC	IPLT2	IPLS	IPLT1	IPLX1	IPLT0	IPLX0

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPLC	**EWC–PCA Counter Interrupt Priority level less significant bit** Refer to IPHC for priority level.
5	IPLT2	**Timer 2 Interrupt Priority level less significant bit** Refer to IPHT2 for priority level.
4	IPLS	**Serial Port Interrupt Priority level less significant bit** Refer to IPHS for priority level.
3	IPLT1	**Timer 1 Interrupt Priority level less significant bit** Refer to IPHT1 for priority level.
2	IPLX1	**External Interrupt 1 Priority level less significant bit** Refer to IPHX1 for priority level.
1	IPLT0	**Timer 0 Interrupt Priority level less significant bit** Refer to IPHT0 for priority level.
0	IPLX0	**External Interrupt 0 Priority level less significant bit** Refer to IPHX0 for priority level.

Reset Value= X000 0000b

IPL1 (S:B2h)

Interrupt Priority Low Register 1

7	6	5	4	3	2	1	0
–	–	IPLSS	–	–	–	–	IPLKB

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	IPLSS	**SSLC Interrupt Priority level less significant bit** Refer to IPHSS for priority level.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IPLKB	**Keyboard Interrupt Priority level less significant bit.** Refer to IPHKB for priority level.

Reset Value= XX0X XXX0b

PCON (S:87h)

Power configuration Register

7	6	5	4	3	2	1	0
SMOD1	SMOD0	RPD	POF	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Double Baud Rate bit** Set to double the Baud Rate when Timer 1 is used and mode 1, 2 or 3 is selected in SCON register.
6	SMOD0	**SCON Select bit** When cleared, read/write accesses to SCON.7 are to SM0 bit and read/write accesses to SCON.6 are to SM1 bit. When set, read/write accesses to SCON.7 are to FE bit and read/write accesses to SCON.6 are to OVR bit. SCON is Serial Port Control register.
5	RPD	**Recover from Idle/Power–Down bit** Clear to disable the Recover from Idle and Power–Down modes feature. Set to enable the Recover from Idle and Power–Down modes feature.
4	POF	**Power–Off flag** **S**et by hardware when VDD rises above V_{RET+} to indicate that the Power Supply has been set off. Must be cleared by software.
3	GF1	**General Purpose flag 1** One use is to indicate wether an interrupt occured during normal operation or during Idle mode.
2	GF0	**General Purpose flag 0** One use is to indicate wether an interrupt occured during normal operation or during Idle mode.
1	PD	**Power–Down Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power–Down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence.

Reset Value= 0000 0000b

PFILT (S:86h)

Power Filter Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**PFILT Value.**

Reset Value= XXXX XXXXb

POWM (S:8Fh)

Power Management Register

7	6	5	4	3	2	1	0
CKSRC	–	–	–	RSTD	–	–	–

Bit Number	Bit Mnemonic	Description
7	CKSRC	**Clock Source bit** Cleared by hardware after a Power-Up. In that case: F_{OSC}= F_{XTAL}. Set to enable the clock. In that case: F_{OSC}= F_{XTAL} / 2 · (CKRL + 1).
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	RSTD	**Reset Detector Disable bit** Clear to enable the Reset detector. Set to disable the Reset detector.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value= 0XXX 0XXXb

PSW (S:D0h)
Program Status Word register

CY	AC	FO	RS1	RS0	OV	UD	P
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	CY	**Carry flag** The carry flag is set by an addition (ADD, ADDC) if there is a carry out of the MSB. It is set by a subtraction (SUB, SUBB) or compare (CMP) if a borrow is needed for the MSB. The carry flag is also affected by some rotate and shift instructions, logical bit instructions and bit move instructions, and the multiply (MUL) and decimal adjust (DA) instructions.
6	AC	**Auxiliary Carry flag** The auxiliary flag is affected only by instructions that address 8-bit operands. The AC flag is set if an arithmetic instruction with an 8-bit operand produces a carry out of bit 3 (from addition) or a borrow into bit 3 (from subtraction). Otherwise it is cleared. This flag is useful for BCD arithmetic.
5	FO	**Flag 0** This general-purpose flag is available to the user.
4	RS1	**Register Bank Select bit 1** This bit selects the memory locations that comprise the active bank of the register file (registers R0-R7). RS1 Bank Address 0 0 00h-07h 0 1 08h-0Fh 1 2 10h-17h 1 3 18h-1Fh
3	RS0	**Register Bank Select bit 0** This bit selects the memory locations that comprise the active bank of the register file (registers R0-R7). RS0 Bank Address 0 0 00h-07h 1 1 08h-0Fh 0 2 10h-17h 1 3 18h-1Fh
2	OV	**Overflow flag** This bit is set if an addition or subtraction of signed variables results in an overflow error (i.e., if the magnitude of the sum or differnecce is too great for the seven LSBs in 2's-complement representation). The overflow flag is also set if a multiplication product overflows one byte or if a division by zero is attempted.
1	UD	**User-definable flag** This general-purpose flag is available to the user.
0	P	**Parity bit** This bit indicates the parity of the accumulator. It is set if an odd number of bits in the accumulator are set. Otherwise, it is cleared. Not all instructions update the parity bit.

Reset Value = 0000 0000b

PSW1 (S:D1h)
Program Status Word 1 register

CY	AC	N	RS1	RS0	OV	Z	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	CY	**Carry flag** Identical to the CY bit in the PSW register.
6	AC	**Auxiliary Carry flag** Identical to the AC bit in the PSW register.
5	N	**Negative flag** This bit is set if the result of the last logical or arithmetic operation was negative, i.e., bit15 = 1. Otherwise it is cleared.
4	RS1	**Register Bank Select bit 1** Identical to the RS1 bit in the PSW register.
3	RS0	**Register Bank Select bit 0** Identical to the RS0 bit in the PSW register.
2	OV	**Overflow flag** Identical to the OV bit in the PSW register.
1	Z	**Zero flag** This flag is set if the result of the last logical or arithmetic operation is zero. Otherwise it is cleared.
0	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Reset Value = 0000 0000b

P0 (S:80h)

Port 0

P0 is the SFR that contains data to be driven out from from the port 0 pins. Read–modify–write instructions that read port 0 read this register. The other instructions that read port 0 read the port 0 pins. When port 0 is used for an external bus cycle, the CPU always write FFh to P0 and the former contents of P0 are lost.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0	P0 7:0	**Port 0 Data** Write data to be driven onto the port 0 pins to these bits.

Reset Value= 1111 1111b

P1 (S:90h)

Port 1

P1 is the SFR that contains data to be driven out from from the port 1 pins. Read–modify–write instructions that read port 1 read this register. The other instructions that read port 1 read the port 1 pins.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0	P1 7:0	**Port 1 Data** Write data to be driven onto the port 0 pins to these bits.

Reset Value= 1111 1111b

P1F (S:9Eh)

Port 1 Flag Register

7	6	5	4	3	2	1	0
P1F.7	P1F.6	P1F.5	P1F.4	P1F.3	P1F.2	P1F.1	P1F.0

Bit Number	Bit Mnemonic	Description
7	P1F.7	**Port 1 line 7:0 flag** Set by hardware when the Port line 7 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.7 bit in P1IE register is set. Must be cleared by software.
6	P1F.6	**Port 1 line 6 flag** Set by hardware when the Port line 6 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.6 bit in P1IE register is set. Must be cleared by software.
5	P1F.5	**Port 1 line 5 flag** Set by hardware when the Port line 5 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.5 bit in P1IE register is set. Must be cleared by software.
4	P1F.4	**Port 1 line 4 flag** Set by hardware when the Port line 4 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.4 bit in P1IE register is set. Must be cleared by software.
3	P1F.3	**Port 1 line 3 flag** Set by hardware when the Port line 3 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.3 bit in P1IE register is set. Must be cleared by software.
2	P1F.2	**Port 1 line 2 flag** Set by hardware when the Port line 2 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.2 bit in P1IE register is set. Must be cleared by software.
1	P1F.1	**Port 1 line 1 flag** Set by hardware when the Port line 1 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.1 bit in P1IE register is set. Must be cleared by software.
0	P1F.0	**Port 1 line 0 flag** Set by hardware when the Port line 0 detects a programmed level. It generates a Keyboard interrupt request if the P1IE.0 bit in P1IE register is set. Must be cleared by software.

Reset Value= 0000 0000b

P1IE (S:9Dh)

Port 1 Input Interrupt Enable Register

7	6	5	4	3	2	1	0
P1IE.7	**P1IE.6**	**P1IE.5**	**P1IE.4**	**P1IE.3**	**P1IE.2**	**P1IE.1**	**P1IE.0**

Bit Number	Bit Mnemonic	Description
7	P1IE.7	**Port 1 line 7 Interrupt Enable bit** Clear to disable P1F.7 bit in P1F register to generate an interrupt request. Set to enable P1F.7 bit in P1F register to generate an interrupt request.
6	P1IE.6	**Port 1 line 6 Interrupt Enable bit** Clear to disable P1F.6 bit in P1F register to generate an interrupt request. Set to enable P1F.6 bit in P1F register to generate an interrupt request.
5	P1IE.5	**Port 1 line 5 Interrupt Enable bit** Clear to disable P1F.5 bit in P1F register to generate an interrupt request. Set to enable P1F.5 bit in P1F register to generate an interrupt request.
4	P1IE.4	**Port 1 line 4 Interrupt Enable bit** Clear to disable P1F.4 bit in P1F register to generate an interrupt request. Set to enable P1F.4 bit in P1F register to generate an interrupt request.
3	P1IE.3	**Port 1 line 3 Interrupt Enable bit** Clear to disable P1F.3 bit in P1F register to generate an interrupt request. Set to enable P1F.3 bit in P1F register to generate an interrupt request.
2	P1IE.2	**Port 1 line 2 Interrupt Enable bit** Clear to disable P1F.2 bit in P1F register to generate an interrupt request. Set to enable P1F.2 bit in P1F register to generate an interrupt request.
1	P1IE.1	**Port 1 line 1 Interrupt Enable bit** Clear to disable P1F.1 bit in P1F register to generate an interrupt request. Set to enable P1F.1 bit in P1F register to generate an interrupt request.
0	P1IE.0	**Port 1 line 0 Interrupt Enable bit** Clear to disable P1F.0 bit in P1F register to generate an interrupt request. Set to enable P1F.0 bit in P1F register to generate an interrupt request.

Reset Value= 0000 0000b

P1LS (S:9Ch)

Port 1 Level Selector Register

7	6	5	4	3	2	1	0
P1LS.7	**P1LS.6**	**P1LS.5**	**P1LS.4**	**P1LS.3**	**P1LS.2**	**P1LS.1**	**P1LS.0**

Bit Number	Bit Mnemonic	Description
7	P1LS.7	**Port 1 line 7 Level Selection bit** Clear to enable a low level detection on Port line 7. Set to enable a high level detection on Port line 7.
6	P1LS.6	**Port 1 line 6 Level Selection bit** Clear to enable a low level detection on Port line 6. Set to enable a high level detection on Port line 6.
5	P1LS.5	**Port 1 line 5 Level Selection bit** Clear to enable a low level detection on Port line 5. Set to enable a high level detection on Port line 5.
4	P1LS.4	**Port 1 line 4 Level Selection bit** Clear to enable a low level detection on Port line 4. Set to enable a high level detection on Port line 4.
3	P1LS.3	**Port 1 line 3 Level Selection bit** Clear to enable a low level detection on Port line 3. Set to enable a high level detection on Port line 3.
2	P1LS.2	**Port 1 line 2 Level Selection bit** Clear to enable a low level detection on Port line 2. Set to enable a high level detection on Port line 2.
1	P1LS.1	**Port 1 line 1 Level Selection bit** Clear to enable a low level detection on Port line 1. Set to enable a high level detection on Port line 1.
0	P1LS.0	**Port 1 line 0 Level Selection bit** Clear to enable a low level detection on Port line 0. Set to enable a high level detection on Port line 0.

Reset Value= 0000 0000b

P2 (S:A0h)

Port 2

P2 is the SFR that contains data to be driven out from from the port 2 pins. Read–modify–write instructions that read port 2 read this register. The other instructions that read port 2 read the port 2 pins.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Port 2 Data** Write data to be driven onto the port 2 pins to these bits.

Reset Value= 1111 1111b

P3 (S:B0h)

Port 3

P3 is the SFR that contains data to be driven out from from the port 3 pins. Read–modify–write instructions that read port 3 read this register.The other instructions that read port 3 read the port 3 pins.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Port 3 Data** Write data to be driven onto the port 3 pins to these bits.

Reset Value= 1111 1111b

RCAP2H (S:CBh)

Timer 2 Reload/Capture High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 2 Reload/Capture.**

Reset Value= 0000 0000b

RCAP2L (S:CAh)

Timer 2 Reload/Capture Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 2 Reload/Capture.**

Reset Value= 0000 0000b

SADDR (S:A9h)

Slave Individual Address Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Slave Individual Address.**

Reset Value= 0000 0000b

SADEN (S:B9h)

Mask Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Mask Data for Slave Individual Address.**

Reset Value= 0000 0000b

SBUF (S:99h)

Serial Buffer Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Data sent/received by Serial I/O Port.**

Reset Value= XXXX XXXXb

SCON (S:98h)

Serial Control Register

7	6	5	4	3	2	1	0
FE/SM0	**OVR/SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit**. To select this function, set SMOD0 bit in PCON register. Set by hardware to indicate an invalid stop bit. Must be cleared by software.
	SM0	**Serial Port Mode bit 0**. To select this function, clear SMOD0 bit in PCON register. Software writes to bits SM0 and SM1 to select the Serial Port operating mode. Refer to SM1 bit for the mode selections.
6	OVR	**Overrun error bit**. To select this function, set SMOD0 bit in PCON register. Set by hardware to indicate an overwrite of the receive buffer. Must be cleared by software.
	SM1	**Serial Port Mode bit 1.** To select this function, set SMOD0 bit in PCON register. Software writes to bits SM1 and SM0 to select the Serial Port operating mode. SM0 SM1 Mode Description Baud Rate 0 0 0 Shift Register F_{OSC}/12 or variable if SRC bit in BDRCON is set 0 1 1 8–bit UART Variable 1 0 2 9–bit UART $F_{OSC/}$32 or $F_{OSC/}$64 1 1 3 9–bit UART Variable
5	SM2	**Serial Port Mode bit 2** Software writes to bit SM2 to enable and disable the multiprocessor communication and automatic address recognition features. This allows the Serial Port to differentiate between data and command frames and to recognize slave and broadcast addresses.
4	REN	**Receiver Enable bit** Clear to disable reception in mode 1, 2 and 3, and to enable transmission in mode 0. Set to enable reception in all modes.
3	TB8	**Transmit bit 8** Modes 0 and 1: Not used. Modes 2 and 3: Software writes the ninth data bit to be transmitted to TB8.
2	RB8	**Receiver bit 8** Mode 0: Not used. Mode 1 (SM2 cleared): Set or cleared by hardware to reflect the stop bit received. Modes 2 and 3 (SM2 set): Set or cleared by hardware to reflect the ninth bit received.
1	TI	**Transmit Interrupt flag** Set by the transmitter after the last data bit is transmitted. Must be cleared by software.
0	RI	**Receive Interrupt flag** Set by the receiver after the stop bit of a frame has been received. Must be cleared by software.

Reset Value= 0000 0000b

.SP (S:81h)

Stack Pointer Register

SP provides SFR accesses to location 63 in the register file (also named SP). Please refer to programmer’s guide.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Stack Pointer Register Low** Bits 7:0 of the extended stack pointer, SPX (DR60).

Reset Value= 0000 0111b

SPH (S:BEh)

Stack Pointer Register High

SPH provides SFR accesses to location 62 in the register file (also named SPH). Please refer to programmer’s guide.

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Stack Pointer Register High** Bits 15:8 of the extended stack pointer, SPX (DR60).

Reset Value= 0000 0000b

SSADR (S:96h)

Synchronous Serial Address Register

7	6	5	4	3	2	1	0
SSA7	**SSA6**	**SSA5**	**SSA4**	**SSA3**	**SSA2**	**SSA1**	**SSGC**

Bit Number	Bit Mnemonic	Description
7:1	SSA7:1	**Synchronous Serial Slave Address bits 7 to 1.**
0	SSGC	**Synchronous Serial General Call bit** Clear to disable the general call address recognition. Set to enable the general call address recognition.

Reset Value= 0000 0000b

SSBR (S:92h)

Synchronous Serial Bit Rate Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Synchronous Serial Bit Rate Data** Bit rate is given by the formula: Br= F_{OSC} / (4 · (SSBR value + 3)), Br in KHz (F_{OSC} in MHz).

Reset Value= 0000 0000b

SSCON (S:93h) (SSLC I^{2}C)

Synchronous Serial Control Register

7	6	5	4	3	2	1	0
SSCR2	SSPE	SSSTA	SSSTO	SSI	SSAA	SSCR1	SSCR0

Bit Number	Bit Mnemonic	Description
7	SSCR2	**Synchronous Serial Control Rate bit 2** see Table 8.1.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Clear to disable the I^{2}C interface. Set to enable the I^{2}C interface.
5	SSSTA	**Synchronous Serial Start flag** Clear not to send a START condition on the bus. Set to send a START condition on the bus.
4	SSSTO	**Synchronous Serial Stop flag** Clear not to send a STOP condition on the bus. Set to send a STOP condition on the bus.
3	SSI	**Synchronous Serial Interrupt flag** Set by hardware when a serial interrupt is requested. Must be cleared by software to acknowledge interrupt.
2	SSAA	**Synchronous Serial Assert Acknowledge flag** Clear to disable slave modes. Set to enable slave modes. Slave modes are entered when SLA or GCA (if SSGC set) is recognized. Master Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Master Transmitter Mode in progress This bit has no particular effect when in master transmitter mode. Slave Receiver Mode in progress Clear to force a not acknowledge (high level on SDA). Set to force an acknowledge (low level on SDA). Slave Transmitter Mode in progress This bit has no particular effect when in slave transmitter mode.
1	SSCR1	**Synchronous Serial Control Rate bit 1** see Table 8.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** see Table 8.1.

Reset Value= 0000 0000b

SSCON (S:93h) (SSLC μWire/SPI)

Synchronous Serial Control Register

7	6	5	4	3	2	1	0
SSOVR	**SSPE**	**SSCPOL**	**SSCPHA**	**SSI**	**SSMSTR**	**SSCR1**	**SSCR0**

Bit Number	Bit Mnemonic	Description
7	SSOVR	**Synchronous Serial Slave Overrun flag** Set by hardware in slave mode when a shift occurs while SSI is set or when SSDAT is written while SSBSY is set. Clear by software to reset the overflow flag. Cannot be set by software.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Clear to disable the SPI interface. Set to enable the SPI interface.
5	SSCPOL	**Synchronous Serial clock Polarity bit** Clear to have the clock output set to 0 in idle state. Set to have the clock output set to 1 in idle state. ***Note:*** *When the peripheral is disabled, the clock output is 1.*
4	SSCPHA	**Synchronous Serial Clock Phase bit** Clear to have the data sampled when the clock leave the idle state (see SSCPOL). Set to have the data sampled when the clock return to idle state (see SSCPOL).
3	SSI	**Synchronous Serial Interrupt flag** Set by hardware when an 8–bit shift is completed. Must be cleared by software to acknowledge interrupt.
2	SSMSTR	**Synchronous Serial Master bit** Clear to configure the peripheral in slave mode. Set to configure the peripheral in master mode.
1	SSCR1	**Synchronous Serial Control Rate bit 1** see Table 9.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** see Table 9.1.

Reset Value= 0000 0100b

SSCS (S:94h) write (SSLC I^{2}C)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSBRS	**–**	**–**	**–**	**–**	**–**	**–**	**SSMOD**

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Synchronous Serial Bit Rate Selection bit** Clear to select the bit rate controlled by SSCR2 to SSCR0. Set to select the programmable bit rate generator.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	SSMOD	**Interface Selection bit 0** Clear to select the SSLC in I^{2}C mode.

Reset Value= 0XXX XXX0b

SSCS (S:94h) read (SSLC I^{2}C)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSC4	**SSC3**	**SSC2**	**SSC1**	**SSC0**	**0**	**0**	**0**

Bit Number	Bit Mnemonic	Description
7:3	SSC4:0	**Synchronous Serial Status code bits 0 to 4** See Table 8.2 to Table 8.7.
2:0	0	**Always 0.**

Reset Value= F8h

SSCS (S:94h) read/write (SSLC µWire/SPI)

Synchronous Serial Control and Status Register

7	6	5	4	3	2	1	0
SSBRS	–	SSERR	SSBSY	–	–	SSOE	SSMOD

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Synchronous Serial Bit Rate Selection bit** Clear to select the bit rate controlled by SSCR1 and SSCR0. Set to select the programmable bit rate generator.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	SSERR	**Synchronous Serial Slave Error Flag** Set by hardware when SS# is deasserted before the end of a receiving data. Clear by software to reset error flag.
4	SSBSY	**Synchronous Serial Busy bit** Slave Mode Cleared by hardware when one byte shift is completed (then SSI is set). Set by hardware when one byte exchange begins. Master Mode Cleared by hardware when one byte shift is completed (then SSI is set). Clear to abort the transmission before it is completed (then SSI is not set). Set to start the transmission.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	SSOE	**Synchronous Serial Output Enable bit** Clear in slave mode to have the MISO output enabled by SS# pin (P1.4). Set in slave mode to have the MISO output enabled regardless of P1.4. ***Note:*** *this bit has no effect in master mode*
0	SSMOD	**Synchronous Serial Mode selection bit** Set to select the SSLC in µWire/SPI mode.

Reset Value= 0X00 XX00b

SSDAT (S:95h) (SSLC I^{2}C)

Synchronous Serial Data Register

7	6	5	4	3	2	1	0
SSD7	SSD6	SSD5	SSD4	SSD3	SSD2	SSD1	SSD0

Bit Number	Bit Mnemonic	Description
7:1	SSD7:1	**Synchronous Serial Address bits 7 to 1 or Synchronous Serial Data bits 7 to 1.**
0	SSD0	**Synchronous Serial Address bit 0 (R/$\overline{W}$) or Synchronous Serial Data bit 0.**

Reset Value= 0000 0000b

SSDAT (S:95h) (SSLC µWire/SPI)

Synchronous Serial Data Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Synchronous Serial Data.**

TCON (S:88h)

Timer/Counter Control Register

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

Bit Number	Bit Mnemonic	Description
7	TF1	**Timer 1 Overflow flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 1 register overflows.
6	TR1	**Timer 1 Run Control bit** Clear to turn off Timer/Counter 1. Set to turn on Timer/Counter 1.
5	TF0	**Timer 0 Overflow flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 0 register overflows.
4	TR0	**Timer 0 Run Control bit** Clear to turn off Timer/Counter 0. Set to turn on Timer/Counter 0.
3	IE1	**Interrupt 1 Edge flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	**Interrupt 1 Type Control bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

Reset Value= 0000 0000b

TH0 (S:8Ch)

Timer 0 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 0.**

Reset Value= 0000 0000b

TH1 (S:8Dh)

Timer 1 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 1.**

Reset Value= 0000 0000b

TH2 (S:CDh)

Timer 2 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 2.**

Reset Value= 0000 0000b

TL0 (S:8Ah)

Timer 0 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 0.**

Reset Value= 0000 0000b

TL1 (S:8Bh)

Timer 1 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1.**

Reset Value= 0000 0000b

TL2 (S:CCh)

Timer 2 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 2.**

Reset Value= 0000 0000b

TMOD (S:89h)

Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	**C/T1#**	**M11**	**M01**	**GATE0**	**C/T0#**	**M10**	**M00**

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select bit** Clear for Timer operation: Timer 1 counts the divided–down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select bits** M11 M01 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH1) with 5–bit prescalar (TL1). 0 1 Mode 1: 16–bit Timer/Counter. 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL1). Reloaded from TH1 at overflow. 1 1 Mode 3: Timer 1 halted. Retains count.
4	M01	
3	GATE0	**Timer 0 Gating Control bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select bit** Clear for Timer operation: Timer 0 counts the divided–down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select bit** M10 M00 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH0) with 5–bit prescalar (TL0). 0 1 Mode 1: 16–bit Timer/Counter. 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3: TL0 is an 8–bit Timer/Counter. TH0 is an 8–bit Timer using Timer 1’s TR0 and TF0 bits.
0	M00	

Reset Value= 0000 0000b

T2CON (S:C8h)
Timer/Counter 2 Control Register

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 Overflow flag** TF2 is not set if RCLK= 1 or TCLK= 1. Set by hardware when Timer 2 overflows. Must be cleared by software
6	EXF2	**Timer 2 External flag** EXF2 does not cause an interrupt in up/down counter mode (DCEN= 1). Set by hardware if EXEN2= 1 when a negative transition on T2EX pin is detected.
5	RCLK	**Receive Clock bit** Clear to select Timer 1 as the Timer Receive Baud Rate Generator for the Serial Port in modes 1 and 3. Set to select Timer 2 as the Timer Receive Baud Rate Generator for the Serial Port in modes 1 and 3.
4	TCLK	**Transmit Clock bit** Clear to select Timer 1 as the Timer Transmit Baud Rate Generator for the Serial Port in modes 1 and 3. Set to select Timer 2 as the Timer Transmit Baud Rate Generator for the Serial Port in modes 1 and 3.
3	EXEN2	**Timer 2 External Enable bit** Clear to ignore events on T2EX pin for Timer 2. Set to cause a capture or reload when a negative transition on T2EX pin is detected unless Timer 2 is being used as the Baud Rate Generator for the Serial Port.
2	TR2	**Timer 2 Run Control bit** Clear to turn off Timer 2. Set to to turn on Timer 2.
1	C/T2#	**Timer 2 Counter/Timer Select bit** Clear for Timer operation: Timer 2 counts the divided–down system clock. Set for Counter operation: Timer 2 counts negative transitions on external pin T2.
0	CP/RL2#	**Capture/Reload bit** CP/RL2# is ignored and Timer 2 is forced to auto–reload on Timer 2 overflow if RCLK= 1 or TCLK= 1. Clear to auto–reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2= 1. Set to capture on negative transitions on T2EX pin if EXEN2= 1

Reset Value= 0000 0000b

T2MOD (S:C9h)

Timer/Counter 2 Mode Control Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T2OE	DCEN

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	T2OE	**Timer 2 Output Enable bit** Clear to disable the programmable clock output to external pin T2 in the Timer 2 clock–out mode. Set to enable the programmable clock output to external pin T2 in the Timer 2 clock–out mode.
0	DCEN	**Down Count Enable bit** Clear to configure Timer 2 as an up Counter. Set to configure Timer 2 as an up/down Counter.

Reset Value= XXXX XX00b

WCON (S:A7h)

Real–Time Synchronous Wait State Control Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	RTWCE	RTWE

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	RTWCE	**Real–Time Synchronous WAIT CLOCK enable** Clear to disable synchronous WAIT CLOCK. Set to enable the synchronous WAIT CLOCK on port 1.7 (WCLK). The square wave output signal is one–half the oscillator frequency.
0	RTWE	**Real–Time Synchronous WAIT# enable** Clear to disable real–time synchronous wait state. Set to enable real–time synchronous wait state input on port 1.6 (WAIT#).

Reset Value= 00XX X000b

WDTRST (S:A6h) write

Hardware Watchdog Timer Reset Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Watchdog Control Data.**

Reset Value= 1111 1111b

Section 5

Sales Locations

Sales Locations

Europe Sales Offices

Finland

TEMIC Nordic AB
c/o Atmel OY
Kappelitie 6B
FIN–02200
Tel: 358 9 4520 8219
Fax: 358 9 529 619

France

TEMIC France
Les Quadrants – 3,
avenue du centre
B.P. 309
78054 St.–Quentin–en–Yvelines Cedex
Tel: 33 1 30 60 70 00
Fax: 33 1 30 60 71 11

Germany

TEMIC Semiconductor GmbH
Erfurter Strasse 31
85386 Eching
Tel: 49 89 3 19 70–0
Fax: 49 89 3 19 46 21

TEMIC Semiconductor GmbH
Kruppstrasse 6
45128 Essen
Tel: 49 2 01 24 73 00
Fax: 49 2 01 2 47 30 47

TEMIC Semiconductor GmbH
Theresienstrasse 2
Postfach 3535, PLZ 74025
74072 Heilbronn
Tel: 49 71 31 67 0
Fax: 49 71 31 67 23 40

Italy

TEMIC Italiana
Via Grosio, 10/8
20151 Milano
Tel: 39 02 38 03 71
Fax: 39 02 38 03 72 34

Spain

TEMIC Iberica
Principe de Vergara, 112
28002 Madrid
Tel: 34 91 564 51 81
Fax: 34 91 562 75 14

Sweden

TEMIC Nordic AB
Kavallerivaegen 24, Rissne
Box 2042
17202 Sundbyberg
Tel: 46 8 587 48 800
Fax: 46 8 587 48 850

United Kingdom

TEMIC U.K. Ltd.
Easthampstead Road
Bracknell, Berkshire RG12 1LX
Tel: 44 1344 707 300
Fax: 44 1344 427 371

North America Sales Offices

Western

TEMIC North America
c/o Atmel Corporation
2325 Orchard Parkway
San Jose
California 95131
Tel: 1 408 441 0311
Fax: 1 408 436 4200

Eastern

TEMIC North America Inc.
180 Mount Airy Road,
Ste. 100
Basking Ridge
New Jersey 07920
Tel: 1 908 630 9200
Fax: 1 908 630 9201

Asia Pacific / Japan Sales Offices

China

TEMIC Shanghai
c/o Atmel Corp
4th floor, Block A
Shanghai Eastern Business Bldg
586 Fanyu Road, Shanghai
200052 China
Tel: 86 21 6280 9241
Fax: 86 21 6283 8816

Hong Kong

TEMIC Hong Kong Ltd.
Ste. 1701 World Finance Centre,
South Tower, Harbour City
17 Canton Road, Tsimshatsui, Kowloon
Tel: 852 23 789 789
Fax: 852 23 755 733

Japan

TEMIC Semiconductors
c/o Atmel Japan K.K.
Tonetsushinkawa Bldg.
1–24–8, Shinkawa, chuku
Tokyo 104–0033
Tel: 81 3 3523 3551
Fax: 81 3 3523 7581

Rep. of Singapore

TEMIC Singapore Pte Ltd
Keppel Building, #03–00
25 Tampines Street 92
Singapore 528877
Tel: 65 788 6668
Fax: 65 787 9819

Korea

TEMIC Korea Ltd.
Suite 605, Singsong Bldg.
25–4 Yoido–dong
Youngdeungpo–Ku
150–010 Seoul
Tel: 82 2 785 1136
Fax: 82 2 785 1137

Taiwan, R.O.C.

TEMIC Taiwan, c/o Atmel Corp.
9F–1 NO.266
SEC.1 Wen HWA 2RD
Lin Kon Hsiang
Taipei Hsien
Tel: 886 2 2609 5581
Fax: 886 2 2600 2735