TSC 80251G1

Extended 8–bit Microcontroller with Serial Communication Interfaces

Design Guide – 1996

On line information

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

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

Table of Contents

Section III: Electrical and Mechanical Information

Section IV: Application Notes

Section V: Ordering Information

Section VI: TEMIC Addresses

List of Figures

Section I: Introduction to TSC80251G1

Chapter 3: Block Diagram

Chapter 4: Pin Description

Section II: Design Information

Chapter 1: Configuration and Memory Mapping

Chapter 2: Parallel I/O Ports

Chapter 3: Timers/Counters

Chapter 4: Serial I/O Port

Chapter 5: Event and Waveform Controller

Chapter 6: SSLC / Inter–Integrated Circuit (I^2C) Interface

Chapter 7: SSLC / Synchronous Peripheral Interface

Chapter 8: Hardware Watchdog Timer

Chapter 9: Power Monitoring and Management

Chapter 10: Interrupt System

Section III: Electrical and Mechanical Information

Chapter 1: DC Characteristics

Chapter 2: AC Characteristics

Chapter 3: IC Interface AC/DC Characteristics

Chapter 4: EPROM Programming

Chapter 5: Packages

Section IV: Application Notes

Chapter 1: How to Plug a TSC80251G1 Step A in a C51 Board ?

List of Tables

Section I: Introduction to TSC80251G1

Chapter 4: TSC80251G1 Pin Description

Section II: Design Information

Chapter 1: Configuration and Memory Mapping

Chapter 2: Parallel I/O Ports

Chapter 3: Timers/Counters

Chapter 4: Serial I/O Port

Chapter : Event and Waveform Controller

Chapter 6: SSLC / Inter Integrated Circuit Interface

Chapter 9: Power Monitoring and Management

Chapter 10: Interrupt System

Section III: Electrical and Mechanical Information

Chapter 1: DC Characteristics

Chapter 2: AC Characteristics

Chapter 4: EPROM Programming

Chapter 5: Packages

Section IV: How to Plug a TSC80251G1 Step A in a C51 Board ?

General Introduction

Extended 8–bit Microcontroller with Serial Communication Interfaces

The TSC80251G1 products are derivatives of the TEMIC Application Specific Microcontroller family based on the extended 8–bit C251 architecture described below.

This family of products are tailored to microcontroller applications requiring highly increased instruction throughput and addressable memory space combined with an optimized internal power management.

Three major features have been implemented to provide optimized performance to the designer:

© Serial Communication Interfaces: I2C/µWire/SPI and RS232 protocols

© Power Monitoring and Management Unit:
- Power–Fail reset
- Internal clock prescaler
- Power–Down mode (current < 20µA)

© 256 Kbytes of external addressable memory for code and data

1.1. Application Focus

Typical applications for these products are ISDN–terminals, digital and analog subscriber linecards, PABX systems, networking applications, high speed modems, computer peripherals or similar systems in other segments. With the high instruction throughput, the TSC80251G1 products are focusing on all high–end 8–bit to 16–bit applications. They are also well suited to systems where a lower operating frequency is needed to reduce power consumption or Radio Frequency Interference (RFI), while maintaining a high level of CPU power.

1.2. C251 Architecture

The C251 architecture at its lowest performance level, is binary code compatible with the 80C51 architecture. Due to a 3–stage instruction pipeline, the average CPU performance is increased by 5 times, using existing 80C51 code without any modification.
Using the new C251 instruction set, the performance is increased by up to 15 times, at the same clock rate. This performance enhancement is based on a new 16–bit and even partly 32–bit oriented powerful instruction set, and additional internal 8 and 16–bit data busses. A 24–bit address bus will allow an extension of the address space up to 16 Mbytes for future derivatives.
Programming flexibility and C–code efficiency are both increased by the register–based architecture, the 64–Kbyte extended stack space and the new instruction set.
Combining the above features of the C251 core, the final code size could be reduced by a factor of 3, compared to an 80C51 implementation.

The TSC80251G1 derivatives implement Intel's core revision A and all technical information in this document relates to this revision. The latest Intel's core revision is presently being integrated by TEMIC.

Both revisions are upward compatible, so that no problem will appear if a revision A product is replaced by a new one. The major differences are some additional features in the configuration bytes and a modified emulator interface which is not impacting your application.

This document will be released as soon as the first TSC80251G1 product will be available in the new revision.

1.3. TSC80251G1 Products

The TSC80251G1 is available as a ROMless version (TSC80251G1) or with on–chip Mask Programmable ROM (TSC83251G1). The TSC87251G1 is an EPROM version compatible to the Mask ROM version.
The standard production packages are 44 pins PLCC or QFP, 40 pins PDIL.
All products can be delivered as 12 or 16 MHz versions at 5 Volts and in all major temperature ranges.
ROMless and Mask ROM versions are also available in 3 Volts.

1.4. TSC80251G1 Documentation and Tools

The following documentation and Starter tools are available to use of the TEMIC TSC80251G1 derivatives:

© "TSC80251G1 Extended 8–bit Microcontroller with Serial Communication Interfaces Design Guide"
Contains all information about the G1 derivatives (block diagram, memory mapping, peripheral description, electrical mechanical and ordering information) and application notes.

© "TSC80251 Extended 8–bit Microcontrollers Programmer's Guide"
Contains all information for the programmer (architecture, instruction set, programming, software tools).

© "TSC80251G1 Starter Kits"
These kits enable the TSC80251G1 to be evaluated by the designer.
Their contents is:

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

Please visit our WWW for updated versions in ZIP format.

© "TSC80251G1 Development Tools"
See chapter "Tool Vendors" in the Programmer's Guide (Keil, Tasking, Hitex, Metalink, Nohau).

© World Wide Web
Please contact our WWW for possible updated information at http://www.temic.de

© TSC80251 e–mail hotline: C251@temic.fr

Section I

Introduction to TSC80251G1

Core Features

Based on the extended 8–bit C251 architecture, the TSC80251G1 includes a complete set of new or improved C51 compatible peripherals as well as multiple protocol serial interfaces.

The key features of the new C251 architecture are:

- © Intel's MCS 251 compliance
- © Register–based architecture:
  - 40–byte register file
  - Registers accessible as bytes, words, and double word
- © 3-stage instruction pipeline
- © Enriched instruction set
  - 16–bit and 32–bit arithmetic and logic instructions
  - Compare and conditional jump instructions
  - Expanded set of move instructions
- © Reduced instruction set
  - 189 generic instructions
  - Free space for additional instructions in the future
  - Additionally all 80C51 instructions are usable in binary mode
- © 16–bit internal code fetch
- © 64 Kbytes extended stack space
- © Maximum addressable memory of 16 Mbytes

The benefits of this new architecture are:

- © 5 times 80C51 performances in binary mode (80C51 binary code compatibility)
- © 15 times 80C51 performances in source mode (full architecture performance)
- © Efficient C language support: up to a factor 3 of code size reduction (when a C program for 80C51 is recompiled in C251 language)
- © Complete system development support
  - Compatible with existing tools
  - New tools available: C–Compiler, Assembler, Debugger, ICE
- © Reduction of RFI and power consumption (reduced operating frequency)

Product Features

- © C251 core based (MCS 251Intel compliance)
- © Pin–Out compatibility with 80C51 standard products
- © 1 Kbyte of internal RAM
- © TSC83251G1: 16 Kbytes of on–chip masked ROM
- © TSC87251G1: 16 Kbytes of internal programmable ROM (OTP or UV erasable in window package)
- © TSC80251G1: ROMless version
- © External memory space (Code/Data) programmable from 64 Kbytes to 256 Kbytes
- © Four 8–bit parallel I/O Ports (Ports 0, 1, 2 and 3 of the standard 80C51)
- © Three 16–bit Timers/Counters (Timers 0, 1 and 2 of the standard 80C51)
- © Serial I/O Port: full duplex UART (80C51 compatible) with independent Baud Rate Generator
- © EWC: Event and Waveform Controller
  - Compatible with PCA: Programmable Counter Array (5 × 16–bit modules)
  - High–speed output
  - Compare/Capture I/O
  - 8–bit Pulse Width Modulator (PWM)
  - Watchdog Timer capabilities
- © SSLC: Synchronous Serial Link Controller
  - I2C protocol
  - µWire and SPI protocols
- © 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 on Port 1
- © Non Maskable Interrupt input (NMI)
- © Real–time Wait states input (WAIT#)
- © Power Supply: 5 V +/– 10% and 3 V +/– 10% (*)
- © Up to 16 MHz operation and three temperature ranges:
  - Commercial (0 to +70°C)
  - Industrial (–40 to +85°C)
  - Automotive (–40 to +125°C)
- © Packages : PLCC44, CQPJ44 (window), QFP44 and PDIL40 (**)

* Please contact your sales office for availability of 3 V option
** Please contact your sales office for QFP and PDIL availability

Block Diagram

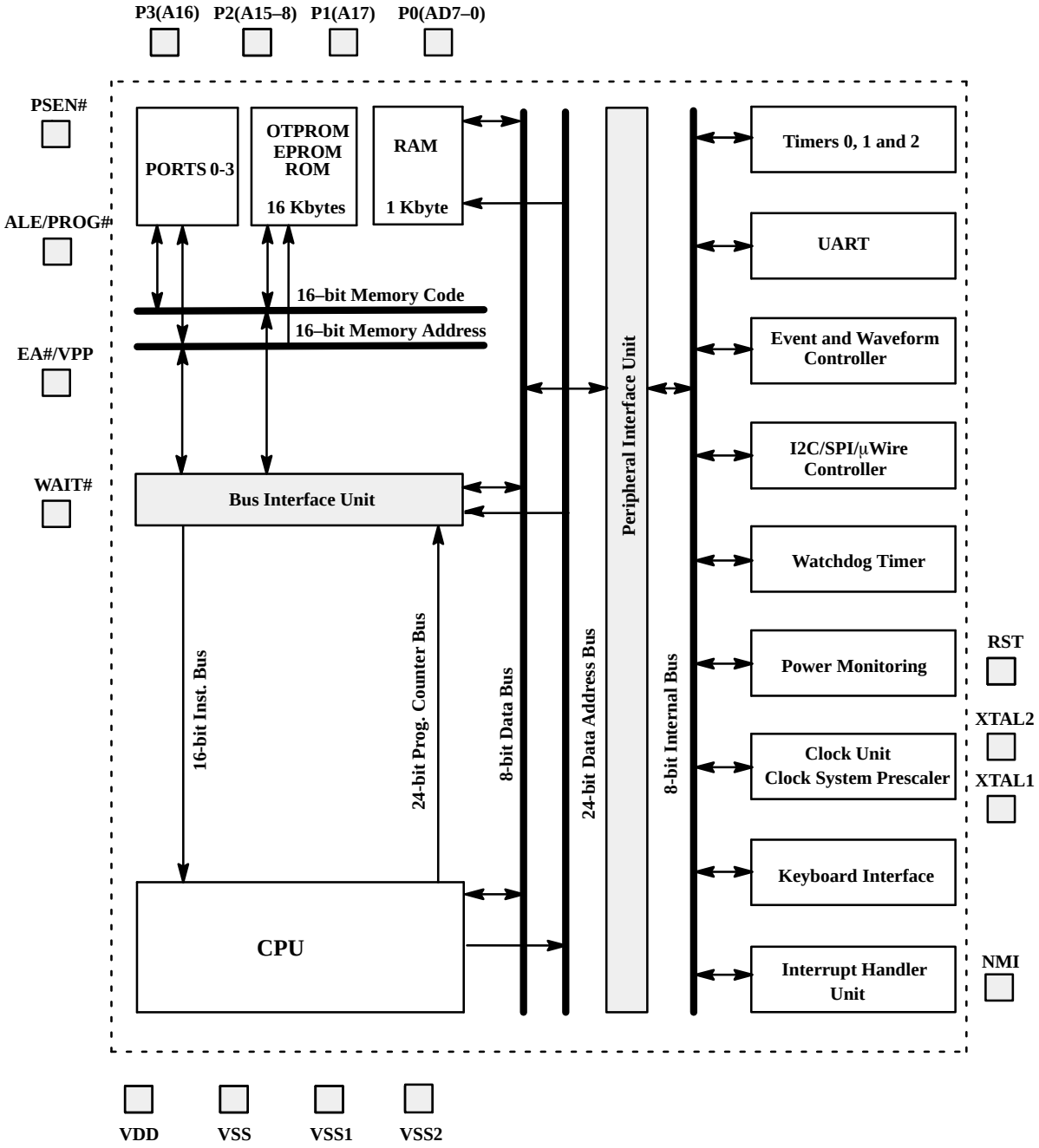


Figure 3.1. TSC80251G1 Block Diagram

Pin Description

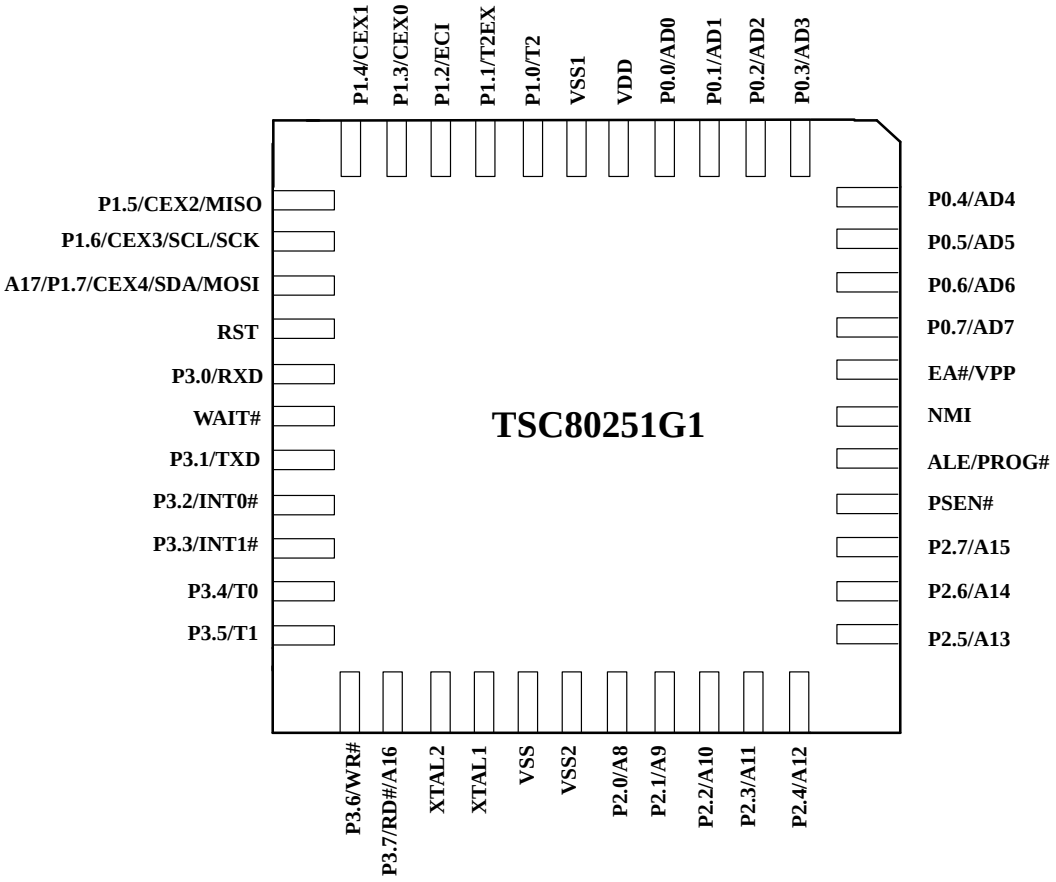


Figure 4.1. TSC80251G1 Pin Description

Caution:
See “Packages” chapter in section III for position of pin #1.

Table 4.1. TSC80251G1 Pin Description

Pin	Type	Description
P0.0:7	I/O	**Port 0** This is an 8–bit open–drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high–impedance inputs. It is also Address/Data lines AD0:7, which are multiplexed lower address lines and data lines for external memory. **External pull–ups are required during program verification.**
P1.0:7	I/O	**Port 1** This is an 8–bit bidirectional I/O port. It receives the low–order address byte during EPROM programming and verification. It serves also the functions of various special features: P1.0 T2 : Timer 2 external clock input/output P1.1 T2EX : Timer 2 external input P1.2 ECI : EWC external clock input P1.3 CEX0 : EWC module 0 Capture input/PWM output P1.4 CEX1 : EWC module 1 Capture input/PWM output P1.5 CEX2 : EWC module 2 Capture input/PWM output MISO : µWire/SPI master input slave output P1.6 CEX3 : EWC module 3 Capture input/PWM output SCL : I2C clock SCK : µWire/SPI serial clock P1.7 A17 : Address line for the 256–Kbyte memory space depending on the byte CONFIG0 CEX4 : EWC module 4 Capture input/PWM output SDA : I2C synchronous serial link data MOSI : µWire/SPI master output slave input Port 1 is also used as a keyboard interface.
P2.0:7	I/O	**Port 2** This is an 8–bit bidirectional I/O port with internal pull-ups. It receives data during EPROM programming and verification. It is also Address lines A8:15, which are upper address lines for external memory.
P3.0:7	I/O	**Port 3** This is an 8–bit bidirectional I/O port with internal pull-ups. It receives the high–order address bits during EPROM programming and verification. It serves also the functions of various special features: P3.0 RXD : Serial Port Receive Data input P3.1 TXD : Serial Port Transmit Data output P3.2 INT0# : External Interrupt 0 P3.3 INT1# : External Interrupt 1 P3.4 T0 : Timer 0 external clock input P3.5 T1 : Timer 1 external clock input P3.6 WR# : Write signal for external access P3.7 A16 : Address line for 128–Kbyte and 256–Kbyte memory space depending on the byte CONFIG0, RD# : Read signal for external access, depending CONFIG0 byte.

Pin	Type	Description
ALE/PROG#	I/O	**Address Latch Enable/Program Pulse** It signals the start of an external bus cycle and indicates that valid address information is available on lines A15:8 and AD7:0. An external latch can use ALE to demultiplex the address from address/data bus. It is also used as the Program Pulse input PROG#, during EPROM programming.
PSEN#	O	**Program Store Enable/Read signal output** This output is asserted for a memory address range that depends on bits RD0 and RD1 in CONFIG0 byte.
EA#/VPP	I	**External Access Enable/Programming Supply Voltage** This input 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 OTPROM/EPROM/ROM if the address is within the range of the on–chip OTPROM/EPROM/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. It receives also the Programming Supply Voltage VPP during EPROM programming operation.
WAIT#	I	**Real–time Wait States Input** When this pin is active (low level), the memory cycle is stretched until it becomes high.
NMI	I	**Non Maskable Interrupt** Holding this pin high for 24 oscillator periods triggers an interrupt.
VDD	PWR	**Digital Supply Voltage**
VSS	GND	**Digital Ground**
VSS1	GND	**Digital Ground**
VSS2	GND	**Digital Ground**
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.
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.

Section II

2

Design Information

Configuration and Memory Mapping

1.1. Introduction

The C251 architecture provides generic configuration and memory addressing capabilities. However, the products based on this architecture may provide various derivative features. The configuration and memory mapping features of the TSC80251G1 derivatives are detailed in this chapter.

1.2. Configuration

The TSC80251G1 derivatives provide design flexibility by configuring some operating features during the device reset. These features fall into the following categories:

- external/internal memory access operation,
- external memory interface,
- source/binary mode opcodes,
- selection of the bytes pushed on the stack by an interrupt.

The choice of internal ROM or external memory access is made through the External Access pin (EA#, see paragraph 1.3.2.). The internal memories of the TSC80251G1 derivatives are detailed in “Memory Mapping” paragraph.
The choice of external memory interface is detailed in this chapter.
The choice of source or binary mode and the interrupt processing are discussed in the TSC80251 Programmers’ Guide.
These settings are made based on two configuration bytes (CONFIG0 and CONFIG1, see Figure 1.11. and Figure 1.12. at the end of this chapter).

1.2.1. Page Mode and Wait States

This part deals with the choice of external bus cycle speed configuration. All the external bus cycles are based on states which are made of two cycles of the internal oscillator. The external XTAL1 frequency can be internally divided by the prescaler to reduce the power consumption (See “Power Monitoring and Management” chapter) and the speed of the external cycles is then reduced accordingly.

TSC80251G1 derivatives use two 8–bit Ports (P0, P2) to multiplex a 16–bit address bus and an 8–bit data bus. The first configuration is multiplexing the lower 8–bit address bus and the 8–bit data bus on Port 0; this is the non–page mode which is compatible with the 80C51 derivatives. The second configuration is multiplexing the upper 8–bit address bus and the 8–bit data bus on Port 2; this is the page mode which improves performance. This bus structure is shown on Figure 1.1 and is configured by the PAGE bit of CONFIG0 byte (See Figure 1.11.).

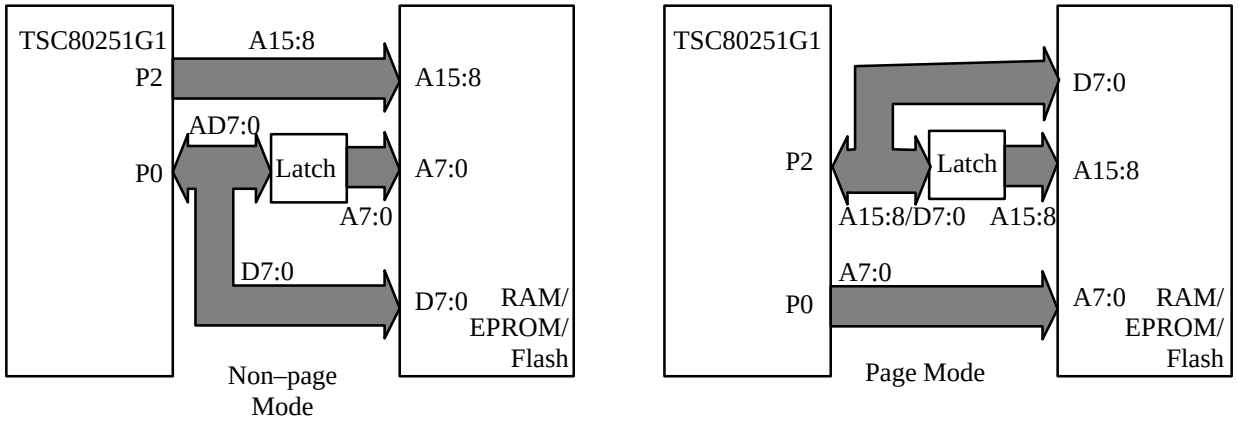
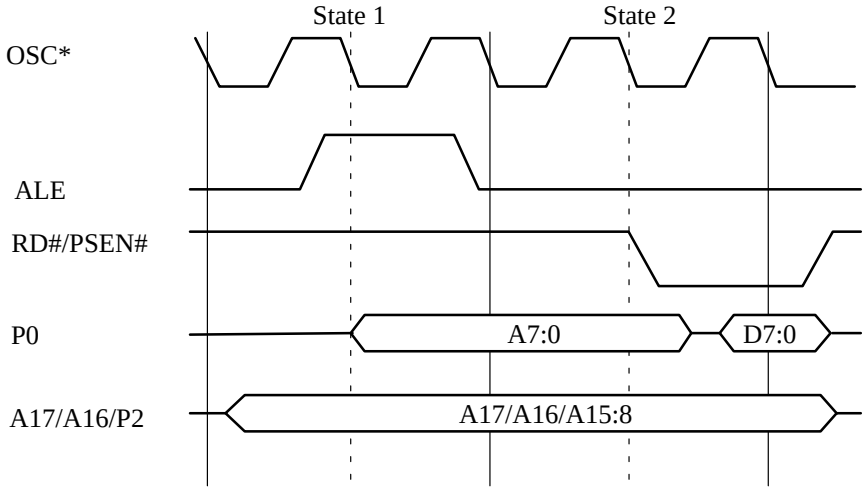


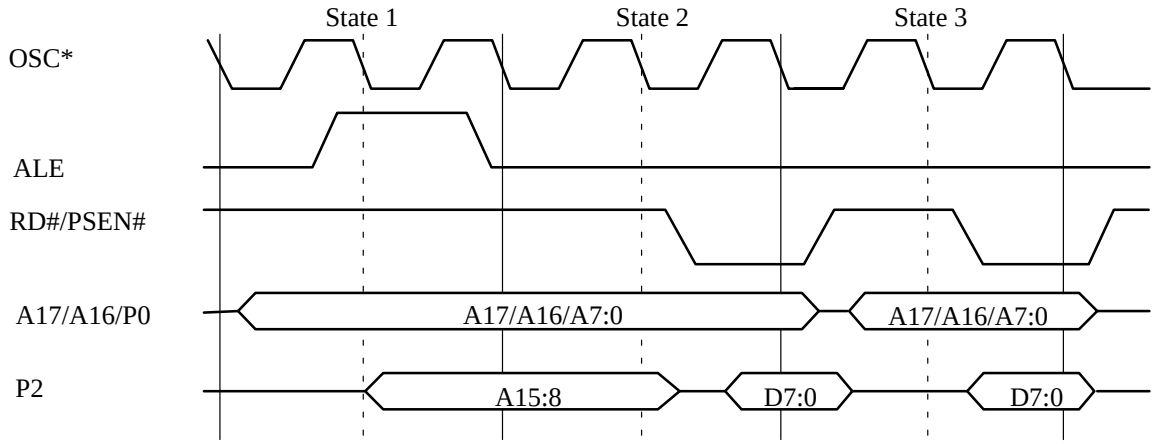
Figure 1.1. Bus Structure in Non–Page Mode and Page Mode

Figure 1.2. highlights the non–page mode configuration with a code fetch cycle. One state is used to latch A7:0 on Port 0, then the data are transferred during the second state.



* The OSC signal is the internal clock signal. It has no link with XTAL1/XTAL2 signals.

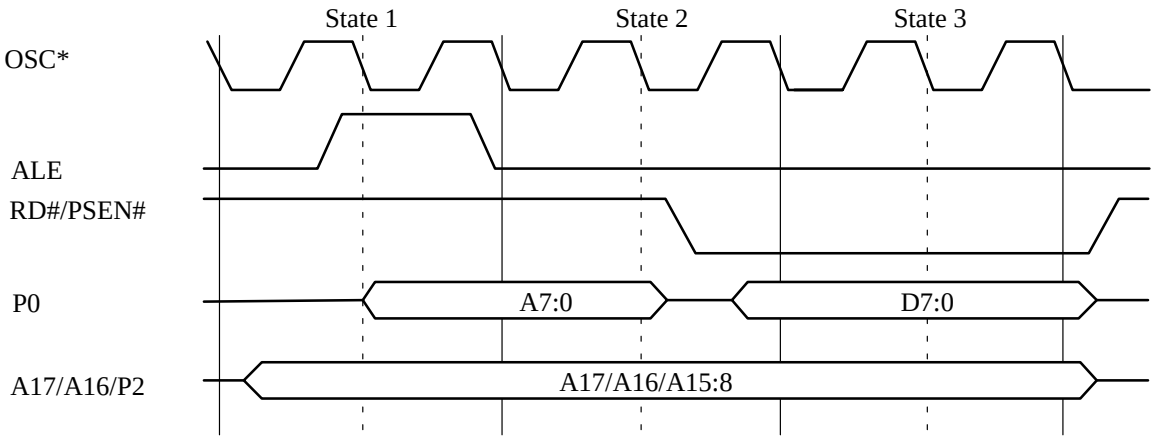
Figure 1.2. External Bus Cycle: Code Fetch, Non–Page Mode



* The OSC signal is the internal clock signal. It has no link with XTAL1/XTAL2 signals.

Figure 1.3. External Bus Cycle: Code Fetch, Page Mode

Three configuration bits are provided to introduce wait states and modulate the access time depending on the external devices. One wait state can be added to extend the address latch time using the XALE bit in CONFIG0 byte. Another wait state can also be added to extend the data access time once the multiplexed addresses have been latched.
Figure 1.4. shows a code fetch in non–page mode with one such wait state. The Wait State A bit (WSA bit in CONFIG0 byte) adds one state for external program/code and data accesses (See segments FF:, FE:, 00: in paragraph 1.2.3.). The Wait State B bit (WSB bit in CONFIG1 byte) adds one state for external data accesses only (See segment 01: in paragraph 1.2.3.).



* The OSC signal is the internal clock signal. It has no link with XTAL1/XTAL2 signals.

Figure 1.4. External Bus Cycle: Code Fetch with One RD#/PSEN# Wait State in Non–Page Mode

1.2.2. Real–time Wait States Input

A WAIT# pin is available on TSC80251G1 to extend the width of the RD#,WR# and PSEN# pulses in order to allow access of slow memories/peripherals. If the WAIT# input is low by the specified time after falls, the TSC80251G1 will hold the bus lines to their values. When the WAIT# input rises the bus cycle continues (See “AC/DC Characteristics” chapter in section III, for timings).

1.2.3. External Memory Signals

For easy reference to the C51 architecture, it is convenient to consider the 24–bit linear address space of the C251 architecture as 256 segments of 64 Kbytes (from segment 00: to segment FF:). Some of these segments are reserved to map the internal registers and, in this section, we only consider the segments which allows to access to the external memory. In the TSC80251G1 derivatives only four segments of the 24–bit internal address space (00:, 01:, FE:, FF:) are implemented to address the external memory. This allows a maximum program or data memory space of 256 Kbytes. Various configurations are possible, depending on the Read configuration bits (RD1:0) which are set in CONFIG0 byte (See Figure 1.11.).

1.2.3.1. How to Address 256 Kbytes

The maximum external memory is provided when RD1:0 = 00, as shown on Figure 1.5. PSEN# is used as a read signal and WR# is used as a write signal. Eighteen address bits are provided externally (A17, A16, P2, P0) to control 256 Kbytes in four segments. In this configuration, the program/code and data spaces share the same external memory segments.

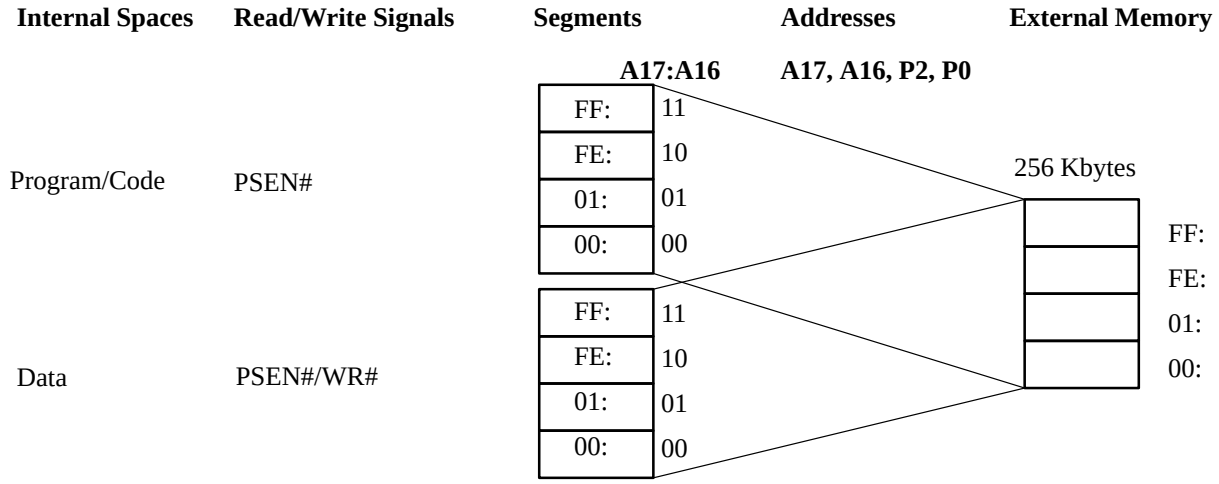


Figure 1.5. Internal/External Memory Segments (RD1:0 = 00)

1.2.3.2. How to Address 128 Kbytes

One I/O pin (P1.7/A17) is saved if 128 Kbytes of external memory are enough, as shown on Figure 1.6. (RD1:0 = 01). PSEN# is used as a read signal and WR# is used as a write signal. Seventeen address bits are provided externally (P0, P2, A16) to control 128 Kbytes in two segments. In this configuration, the program/code and data spaces share the same external memory segments which are replicated twice in each internal space.

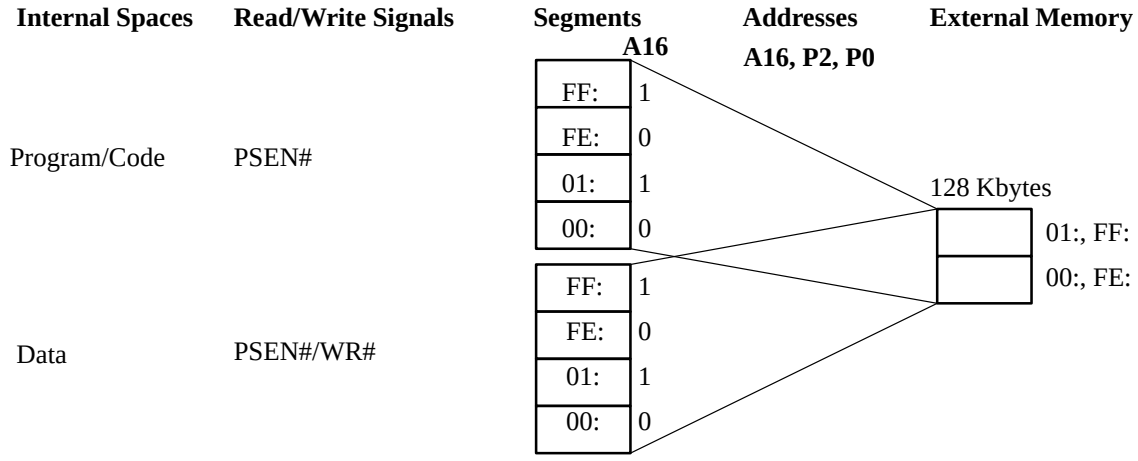


Figure 1.6. Internal/External Memory Segments (RD1:0 = 01)

1.2.3.3. How to Address 64 Kbytes

Two I/O pins (P1.7/A17 and P3.7/A16/RD#) are saved if 64 Kbytes of external memory are enough, as shown on Figure 1.7. (RD1:0 = 10). PSEN# is used as a read signal and WR# is used as a write signal. Sixteen address bits are provided externally (P0, P2) to control 64 Kbytes in one segment. In this configuration, the program/code and data share the same external memory segment which is replicated four times in each internal space.

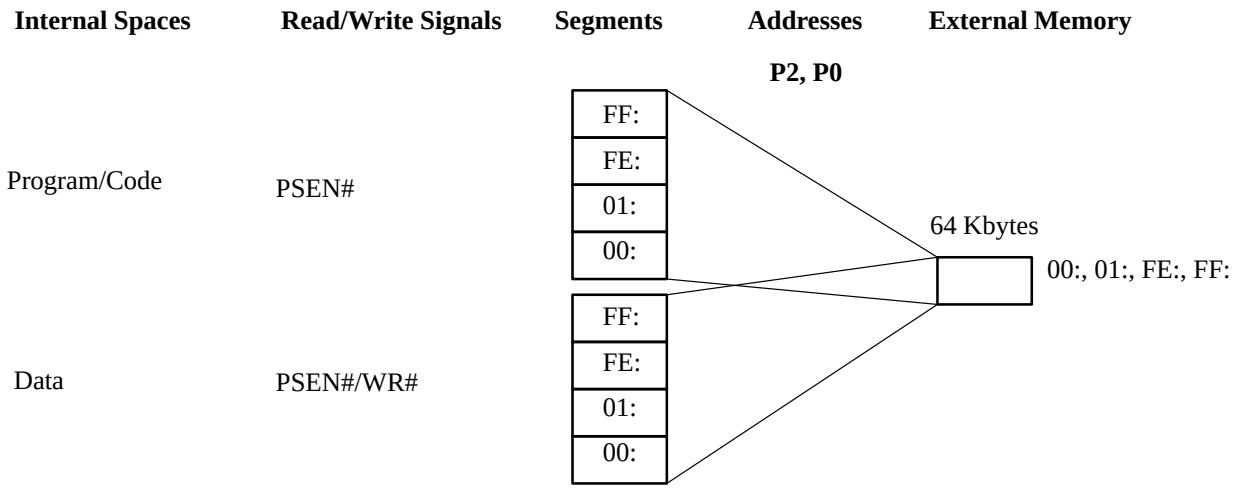


Figure 1.7. Internal/External Memory Segments (RD1:0 = 10)

1.2.3.4. How to Keep C51 Memory Compatibility

The last configuration provides a full compatibility with the C51 architecture, as shown on Figure 1.8. (RD1:0 = 11). PSEN# is used as a read signal for program/code memory while RD# is used as a read signal and WR# is used as a write signal for data memory accesses. 16 address bits are provided externally (P0, P2). In this configuration, the program/code fits in one read–only external memory segment and the data fits in another read–write external memory segment. Each segment is replicated four times in each internal space.

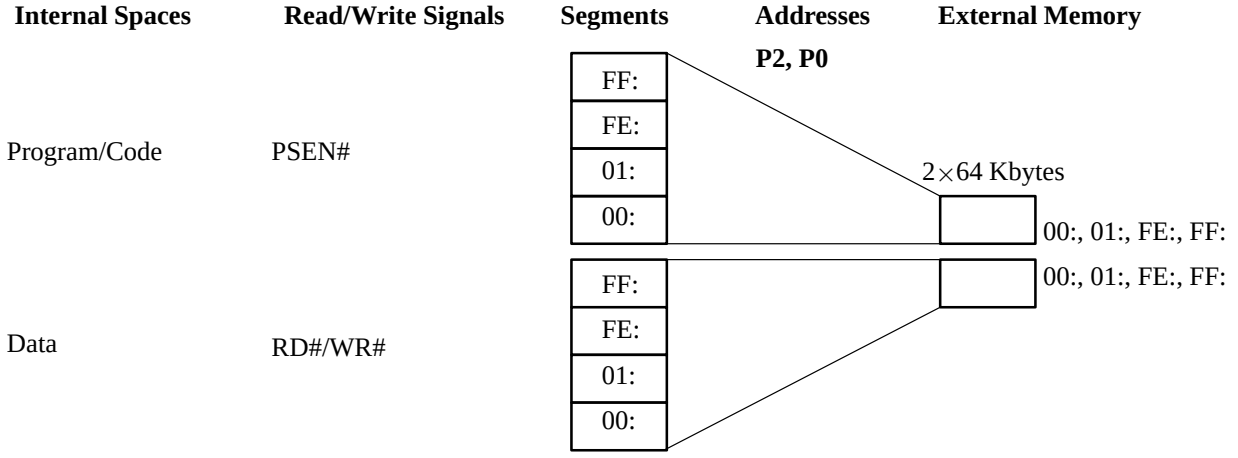


Figure 1.8. Internal/External Memory Segments (RD1:0 = 11)

1.3. Memory Mapping

The specific internal memories of the TSC80251G1 derivatives fall into the following categories:

- 2 Configuration bytes
- 16 Kbytes on–chip ROM or EPROM/OTP program/code memory
- 1 Kbyte on–chip RAM data memory
- Special Function Registers (SFRs)

1.3.1. Configuration Bytes

The configuration bytes, CONFIG0 and CONFIG1, are detailed in Figure 1.11. and Figure 1.12. During reset they are read from a specific ROM area. For the TSC87251G1 EPROM and OTPROM versions, these bytes are programmable in an EPROM area (See “EPROM Programming” chapter). For the TSC83251G1 masked ROM versions, these bytes are additional information provided in a masked ROM area. For the TSC80251G1 ROMless versions, these bytes are configured in factory according to the part number (See “Ordering Information”). These bytes are not accessible by the user during operation and they do not appear in the Memory Mapping of the TSC80251G1 derivatives.

1.3.2. Program/Code Memory

The split of the internal and external program/code memory space is shown on Figure 1.9. If EA# is tied to a high level, the 16–Kbyte internal program memory are mapped in the lower part of segment FF: where the C251 core jumps after reset. The rest of the program/code memory space is mapped to the external memory (See paragraph 1.2.2. to determine to which external memory location each segment actually maps). If EA# is tied to a low level, the internal program/code memory is not used and all the accesses are directed to the external memory.

For the TSC87251G1 EPROM and OTPROM versions, the internal program/code is programmable in EPROM (See “EPROM programming” chapter). For the TSC83251G1 masked ROM versions, the internal program/code is provided in a masked ROM. For the TSC80251G1 ROMless versions, there is no possible internal program/code and EA# must be tied to a low level.

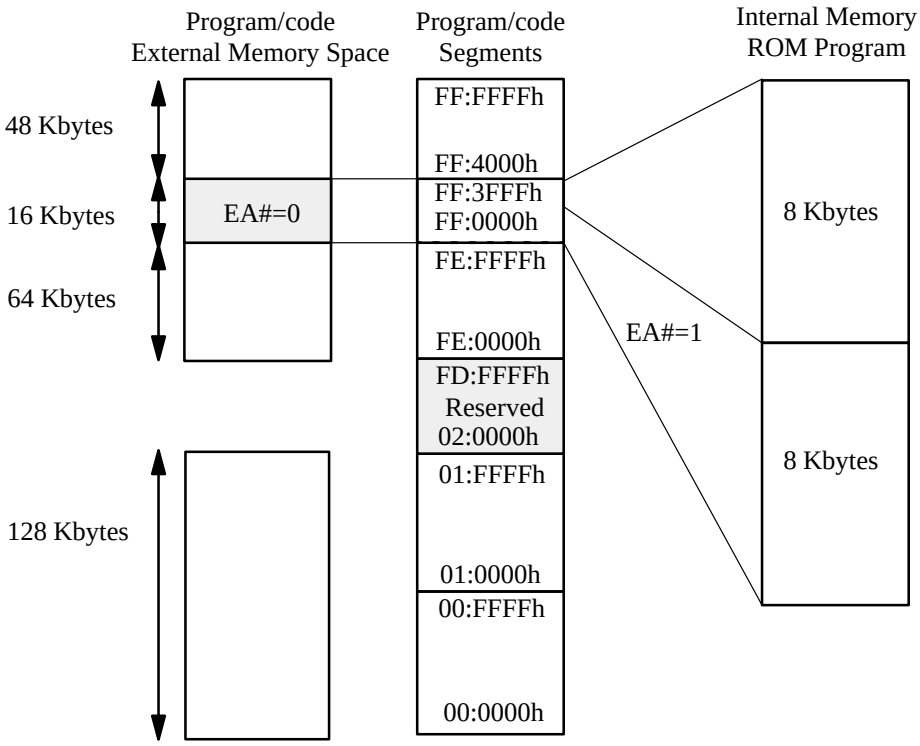


Figure 1.9. Programmable Memory Mapping

Note:

Special care should be taken when the Program Counter (PC) increments:

- If your program executes exclusively from on–chip ROM/OTPROM/EPROM (not from external memory), beware of executing code from the upper eight bytes of the on–chip ROM/OTPROM/EPROM (FF:3FF8h–FF:3FFFF). Because of its pipeline capability, the 80C251G1 may attempt to prefetch code from external memory (at an address above FF:3FF8h/FF:3FFFF) and thereby disrupt I/O Ports 0 and 2. Fetching code constants from these 8 bytes does not affect Ports 0 and 2.
- When PC reaches the end of segment FF:, it loops to the reset address FF:0000h (for compatibility with the C51 Architecture). When PC increments beyond the end of segment FE:, it continues at the reset address FF:0000h (linearity). When PC increments beyond the end of segment 01:, it loops to the beginning of segment 00: (this prevents from it going into the reserved area).

1.3.3. Data Memory

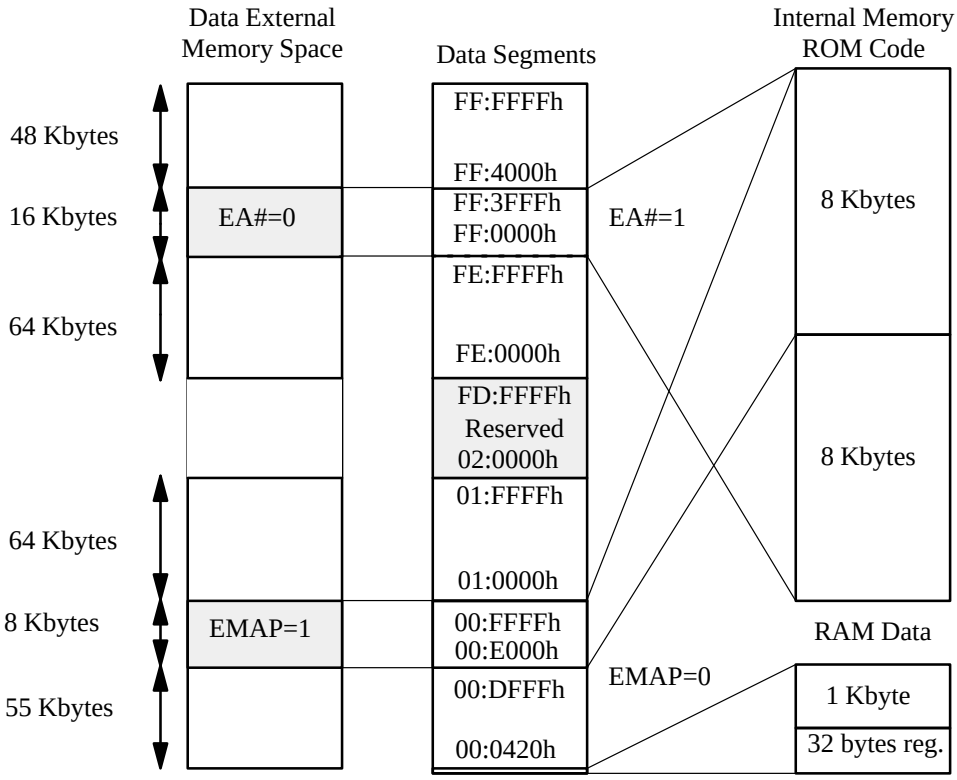


Figure 1.10. Data Memory Mapping

The split of the internal and external data memory space is shown on Figure 1.10. All the TSC80251G1 derivatives feature an internal 1 Kbyte RAM. This memory is mapped in the data space just over the 32 bytes of registers area (See TSC80251 Programmers' Guide). Hence, the lowermost 96 bytes of the internal RAM are bit addressable. This internal RAM is not accessible through the program/code memory space.

For computation with the internal ROM code of the TSC83251G1 and TSC87251G1 versions, its upper 8 Kbytes are also mapped in the segment 00: if the EPROM Map configuration bit is cleared (EMAP bit in CONFIG1 byte, see Figure 1.2.). However, if EA# is tied to a low level, the TSC80251G1 derivative is running as a ROMless and the code is actually fetched in the corresponding external memory (i.e. the upper 8 Kbytes of the lower 16 Kbytes of segment FF:). If EMAP bit is set, the internal ROM is not accessible through the segment 00:.

All the accesses to the portion of the data space with no internal memory mapped onto are redirected to the external memory, see paragraph 1.2.3. to determine to which external memory location each segment actually maps.

1.3.4. Special Function Registers

The Special Function Registers (SFRs) of the TSC80251G1 derivatives fall into the following categories:

- C251 core registers (SP, SPH, DPL, DPH, DPXL, PSW, PSW1, ACC, B)
- Port registers (P0, P1, P2, P3)
- Timer registers (TCON, TMOD, TL0, TL1, TH0, TH1, T2CON, T2MOD, RCAP2L, RCAP2H, T2L, T2H)
- Serial Port and Baud Rate Generator registers (SCON, SBUF, SADDR, SADEN, BDRCON, BRL)
- Event and Waveform Controller registers:
  - Counters (CCON, CMOD)
  - Compare/Capture (CCAPM0, CCAPM1, CCAPM2, CCAPM3, CCAPM4, CCAP0L, CCAP1L, CCAP2L, CCAP3L, CCAP4L, CCAP0H, CCAP1H, CCAP2H, CCAP3H, CCAP4H)
- Synchronous Serial Link Controller registers (SSBR, SSCON, SSCS, SSDAT)
- Power monitoring/management and clock control registers (PCON, PFILT, POWM, CKRL)
- Hardware Watchdog Timer register (WDTRST)
- Keyboard interrupt registers (P1F, P1E, P1LS)
- Interrupt system registers (IE0, IE1, IPL0, IPL1, IPH0, IPH1)

SFRs are placed in a reserved internal memory segment S: which is not represented in the internal memory mapping. The relative addresses within S of these SFRs: are provided together with their reset values in Table 1.2. All the SFRs are bit–addressable using the C251 instruction set. The C251 core registers are in italics in this table and they are described in the TSC80251 Programmers' Guide. The other registers are detailed in the following sections which fully describe each peripheral unit.

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		F8h
F0h	*B*** *0000 0000*								F0h
E8h		CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000	CCAP2L 0000 0000	CCAP3L 0000 0000	CCAP4L 0000 0000		E8h
E0h	ACC 0000 0000								E0h
D8h	CCON 0010 0000	CMOD 0011 1000	CCAPM0 1000 0000	CCAPM1 1000 0000	CCAPM2 1000 0000	CCAPM3 1000 0000	CCAPM4 1000 0000		D8h
D0h	*PSW* *0000 0000*	*PSW1* *0000 0000*							D0h
C8h	T2CON 0000 0000	T2MOD XXXX XX00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000			C8h
C0h									C0h
B8h	IPL0 0000 0000	SADEN 0000 0000					*SPH* *0000 0000*		B8h
B0h	P3 1111 1111	IE1 XX0X XXX0	IPL1 XX0X XXX0	IPH1 XX0X XXX0				IPH0 X000 0000	B0h
A8h	IE0 0000 0000	SADDR 0000 0000							A8h
A0h	P2 1111 1111						WDTRST 1111 1111		A0h
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	P1LS 0000 0000	P1E 0000 0000	P1F 0000 0000		98h
90h	P1 1111 1111		SSBR 0000 0000	SSCON 0000 0000	SSCS 0XXX XXX0	SSDAT 0000 0000			90h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	CKRL 0000 0000	POWM 0XXX 0XXX	88h
80h	P0 1111 1111	SP 0000 0111	*DPL* *0000 0000*	*DPH* *0000 0000*	*DPXL* *0000 0001*		PFILT XXXX XXXX	PCON 0000 0000	80h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Italicized registers are described in the TSC80251 Programmer's Guide (C251 core registers).

reserved

S:00h – S:7Fh unimplemented
S:100h – S:1FFh unimplemented

CONFIG0
Configuration byte 0

–	–	WSA	XALE	RD1	RD0	PAGE	SRC
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to CONFIG0.
6	–	**Reserved** Set this bit when writing to CONFIG0.
5	WSA	**Wait State A bit** Clear to generate one external wait state for memory regions 00:, FE:, and FF:. Set for no wait states for these regions.
4	XALE	**Extend ALE bit** Clear to extend the time of the ALE pulse from T_{OSC} to 3.T_{OSC}, which adds one external wait state. Set the time of the ALE pulse to T_{OSC}.
3	RD1	**RD# and PSEN# Function Select bits** RD1 RD0 RD# P1.7 PSEN# Range 0 0 A16 A17 PSEN# is the read signal for both external data and program address space (256 Kbytes). 0 1 A16 I/O pin PSEN# is the read signal for both external data and program address space (128 Kbytes). 1 0 I/O pin I/O pin PSEN# is the read signal for both external data and program address space (64 Kbytes). 1 1 RD# I/O pin 64–Kbyte code memory space 64–Kbyte data memory space
2	RD0	
1	PAGE	**Page Mode Select bit** Clear for page mode with A15:8/D7:0 on Port 2 and A7:0 on Port0. Set for non–page mode with A15:8 on Port 2 and A7:0/D7:0 on Port 0 (compatible with 80C51 microcontrollers).
0	SRC	**Source Mode/Binary Mode Select bit** Clear for Binary Mode (Binary Code compatible with 80C51 Microcontrollers). Set for Source Mode.

Figure 1.11. Configuration Byte 0

Notes:

- To configure the TSC80251G1 in C51 binary mode, use the following bit values in CONFIG0 byte: 1101 1110b. This configuration is already programmed in –B option microcontroller (See “Ordering Information”).
- To configure the TSC80251G1 in C251 default mode, use the following bit values in CONFIG0 byte: 1111 1111b. This configuration is already programmed in –A option microcontroller (See “Ordering Information”).

CONFIG1
Configuration byte 1

–	–	–	INTR	WSB	–	–	EMAP
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to CONFIG1.
6	–	**Reserved** Set this bit when writing to CONFIG1.
5	–	**Reserved** Set this bit when writing to CONFIG1.
4	INTR	**Interrupt Mode bit** Clear so that the interrupts push 2 bytes onto the stack (the 2 lower bytes of the PC register). Set so that the interrupts push 4 bytes onto the stack (the 3 bytes of the PC register and the PSW1 register).
3	WSB	**Wait State B bit** Clear to generate one external wait state for memory region 01:. Set for no wait states for region 01:.
2	–	**Reserved** Set this bit when writing to CONFIG1.
1	–	**Reserved** Set this bit when writing to CONFIG1.
0	EMAP	**EPROM Map bit** Clear to map the upper 8 Kbytes of on–chip code memory (FF:2000h-FF:3FFFh) to 00:C000h-00:FFFFh. Set not to map the upper 8 Kbytes of on–chip code memory (FF:2000h-FF:3FFFh).

Figure 1.12. Configuration Byte 1

Notes:

- To configure the TSC80251G1 in C51 binary mode, use the following bit values in CONFIG1 byte: 1110 0111b. This configuration is already programmed in –B option microcontroller (See “Ordering Information”).
- To configure the TSC80251G1 in C251 default mode, use the following bit values in CONFIG1 byte: 1111 1111b. This configuration is already programmed in –A option microcontroller (See “Ordering Information”).

Parallel I/O Ports

2.1. Introduction

The TSC80251G1 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 TSC80251G1 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 2.1. , Table 2.2. , Table 2.3. and Table 2.4.).

Table 2.1. 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
P0.1	I/O	AD1 A1	Address/Data line 1 (Non–page mode) Address line 1 (Page mode)	I/O
P0.2	I/O	AD2 A2	Address/Data line 2 (Non–page mode) Address line 2 (Page mode)	I/O
P0.3	I/O	AD3 A3	Address/Data line 3 (Non–page mode) Address line 3 (Page mode)	I/O
P0.4	I/O	AD4 A4	Address/Data line 4 (Non–page mode) Address line 4 (Page mode)	I/O
P0.5	I/O	AD5 A5	Address/Data line 5 (Non–page mode) Address line 5 (Page mode)	I/O
P0.6	I/O	AD6 A6	Address/Data line 6 (Non–page mode) Address line 6 (Page mode)	I/O
P0.7	I/O	AD7 A7	Address/Data line 7 (Non–page mode) Address line 7 (Page mode)	I/O

Table 2.2. 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
P1.6	I/O	CEX3 SCL SCK P1.6	EWC module 3 Capture input/PWM output I^2C serial clock µWire/SPI serial clock Keyboard input 6	I/O O O I
P1.7	I/O	A17 CEX4 SDA MOSI P1.7	Address line 17 EWC module 4 Capture input/PWM output I^2C serial data µWire/SPI master output slave input Keyboard input 7	I/O I/O I/O I/O I

Table 2.3. 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 2.4. 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^2C = Inter–Integrated Circuit
- PWM = Pulse Width Modulation
- SPI = Serial Peripheral Interface
- Refer to Table 9.1. for Pin conditions in Special Operating Modes.

2.2. I/O Configurations

Each Port SFR operates via type–D latches, as illustrated in Figure 2.1. 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 “Read–Modify–Write Instructions” paragraph). Each I/O line may be independently programmed as input or output.

2.3. Port 1 and Port 3

Figure 2.1. 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 2.2. and Table 2.4.).
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 2.1.). The operation of Ports 1 and 3 is discussed further in “Quasi–Bidirectional Port Operation” paragraph.

Note:
To allow proper operation of alternate functions on one pin, the corresponding latch must be set to logical one. Otherwise a logical zero is forced on the pin.

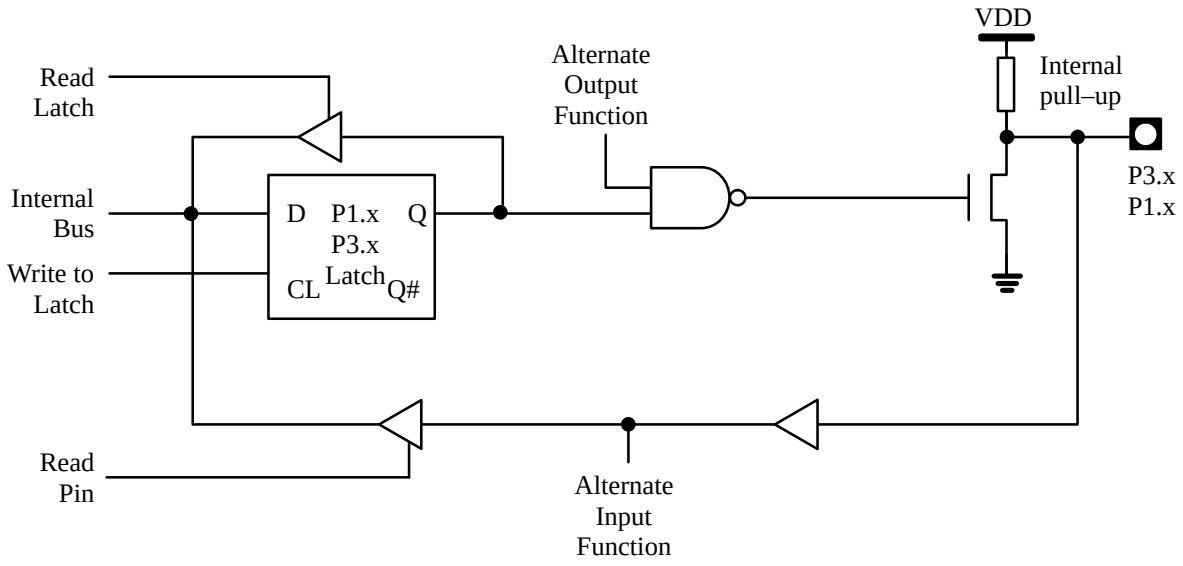


Figure 2.1. Port 1 and Port 3 Structure

2.4. 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 2.2. , differs from the other Ports in not having internal pull–ups. Figure 2.3. 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.

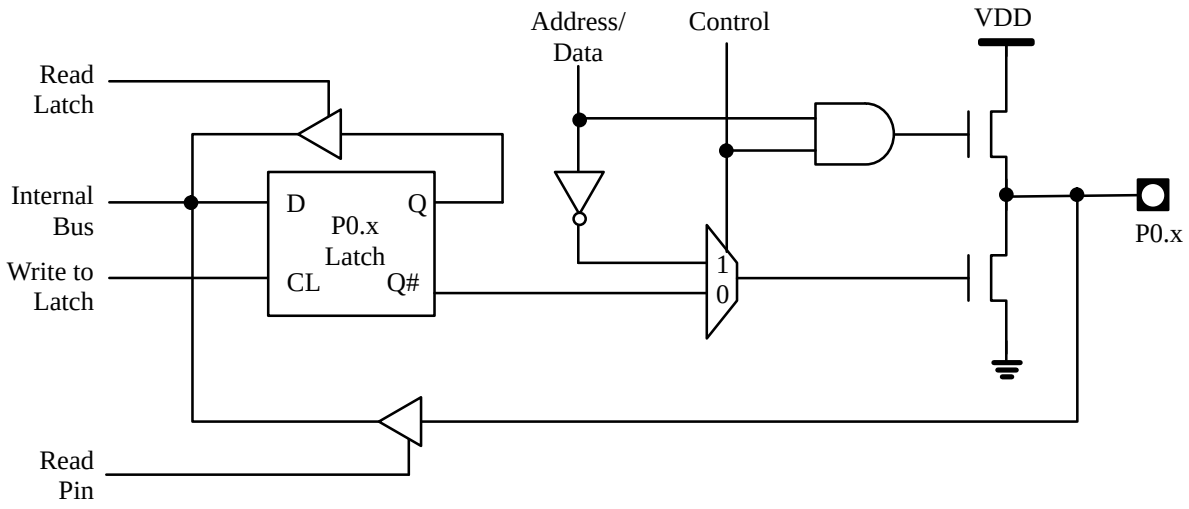


Figure 2.2. Port 0 Structure

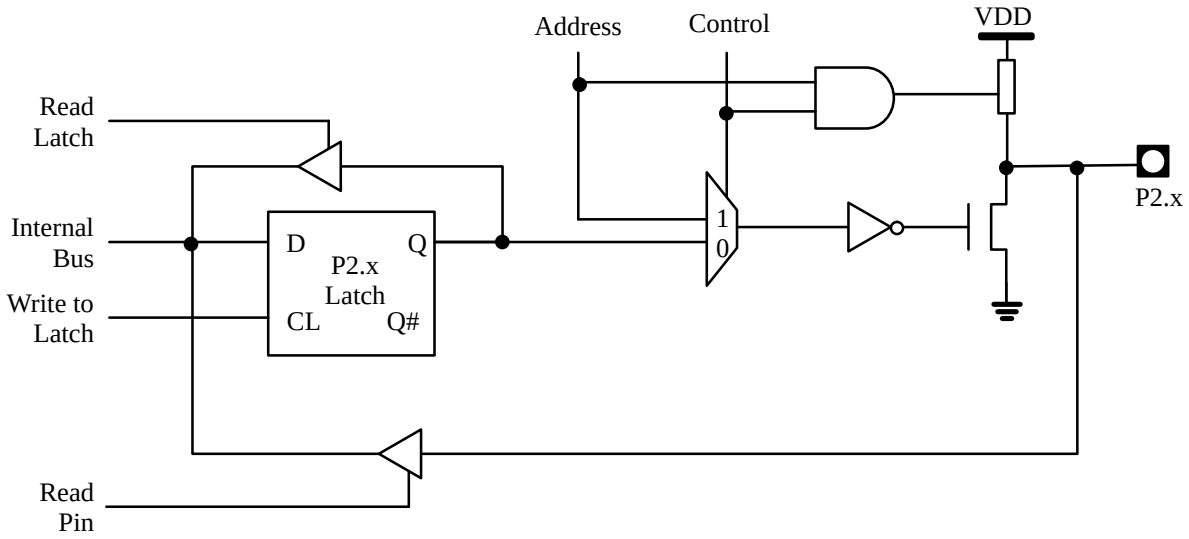


Figure 2.3. Port 2 Structure

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. “External Memory Access” paragraph discusses the operation of Port 0 and Port 2 as the external address/data bus.

Notes:

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

2.5. 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 2.5.). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Table 2.5. 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 specifical 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.

2.6. 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 pin floats 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.

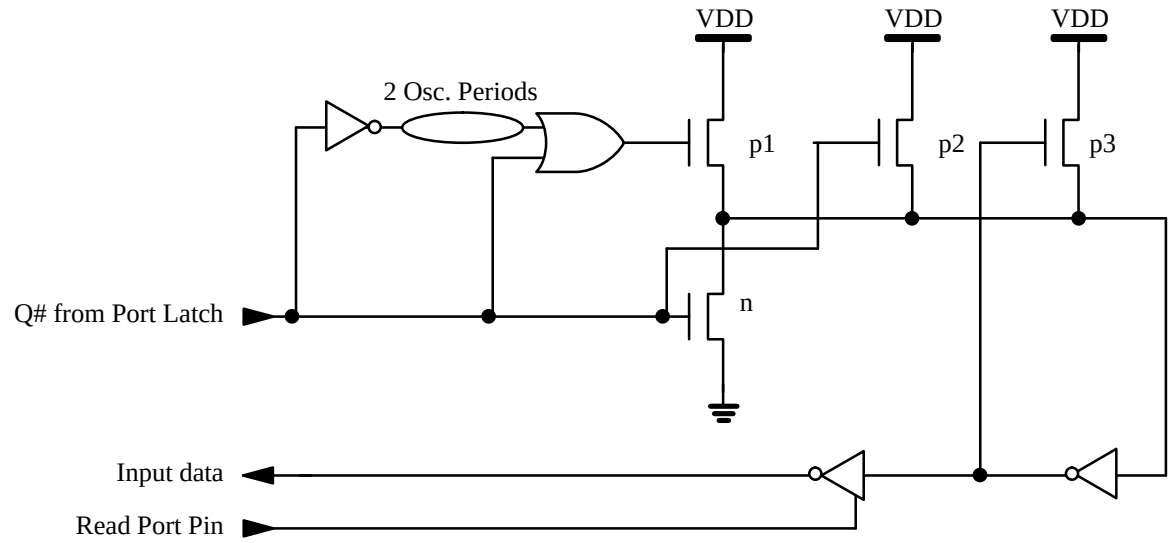


Figure 2.4. Internal Pull–Up Configurations

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 to aid this logic transition (See Figure 2.4.). This increases switch speed. The extra pull–up briefly sources 100 times normal internal circuit current. 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.

2.7. Port Loading

Output buffers of Port 1, Port 2 and Port 3 can each sink 1.6 mA at logic zero (See V_{OL} specification in “Electrical and Mechanical Information” section). 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 2.4.). A logic zero input turns off pFET #3. This leaves only pFET #2 weakly in support of the transition. In external bus mode, Port 0 output buffers each sink 3.2 mA at logic zero (See V_{OL1} specification in “Electrical and Mechanical Information” section). 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.

2.8. External Memory Access

The external bus structure is different for page mode and non–page mode. In non–page mode (used by 80C51 microcontrollers), Port 2 outputs the upper address byte; the lower address byte and the data are multiplexed on Port 0. In page mode, the upper address byte and the data are multiplexed on Port 2, while Port 0 outputs the lower address byte.
The TSC80251G1 CPU writes FFh to the P0 register for all external memory bus cycles. This overwrites previous information in Port 0. In contrast, the P2 register is unmodified for external bus cycles. When address bits or data bits are not on the P2 pins, the bit values in Port 2 appear on the Port 2 pins.

In non–page mode, Port 0 uses a strong internal pull–up FET to output ones or a strong internal pull–down FET to output zeros for the lower address byte and the data. Port 0 is in a high–impedance state for data input. In page mode, Port 0 uses a strong internal pull–up FET to output ones or a strong internal pull–down FET to output zeros for the lower address byte or a strong internal pull–down FET to output zeros for the upper address byte.
In non–page mode, Port 2 uses a strong internal pull–up FET to output ones or a strong internal pull–down FET to output zeros for the upper address byte. In page mode, Port 2 uses a strong internal pull–up FET to output ones or a strong internal pull–down FET to output zeros for the upper address byte and data. Port 2 is in a high–impedance state for data input.

Note:
In external bus mode, Port 0 outputs do not require external pull–ups.

There are two types of external memory accesses: external program memory and external data memory. External program memories use signal PSEN# as a read strobe. 80C51 microcontrollers use RD# (read) or WR# (write) to strobe memory for data accesses. Depending on its RD0 and RD1 configuration bits, the TSC80251G1 uses PSEN# or RD# for data reads (See “Configuration and Memory Mapping” chapter).
During instruction fetches, external program memory can transfer instructions with 16–bit addresses for binary compatible code or with the external bus configured for extended memory addressing (17–bit or 18–bit).
External data memory transfers use an 8–bit, 16–bit, 17–bit or 18–bit address bus, depending on the instruction and the configuration of the external bus. Table 2.6. lists the instructions that can be used for the these bus widths.

Table 2.6. Instructions for External Data Moves

Bus Width	Instructions
8	MOVX @Ri MOV @Rm MOV dir8
16	MOVX @DPTR MOV @WRj MOV @WRj+dis MOV dir16

Table 2.6. Instructions for External Data Moves

Bus Width	Instructions
17	MOV @DRk MOV @DRk+dis
18	MOV @DRk MOV @DRk+dis

Note:
Avoid MOV P0 instructions for external memory accesses. These instructions can corrupt input code bytes at Port 0.

External signal ALE (address latch enable) facilitates external address latch capture. The address byte is valid after the ALE pin drives V_{OL} . For write cycles, valid data is written to Port 0 just prior to the write pin (WR#) asserting V_{OL} . Data remains valid until WR# is undriven. For read cycles, data returned from external memory must appear at Port 0 before the read pin (RD#) is undriven. Waits states, by definition, affect bus–timing.

Timers/Counters

3.1. Introduction

The TSC80251G1 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.
Each Timer/Counter employs two 8–bit Timer registers, used separately or in cascade, to maintain the count. The Timer registers and associated control and capture registers are implemented as addressable Special Function Registers (SFRs). Table 3.1. briefly describes the SFRs referred to in this chapter. 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.

These registers are described in Table 3.1. and at the end of this chapter.

Table 3.1. Timer/Counter SFRs

Mnemonic	Description	Address
TL0 TH0	**Timer 0 registers** Used separately as two 8–bit Counters or in cascade as one 16–bit Counter. Counts an internal clock signal with frequency F_{OSC} /12 (Timer operation) or an external input (event Counter operation).	S:8Ah S:8Ch
TL1 TH1	**Timer 1 registers** Used separately as two 8–bit Counters or in cascade as one 16–bit Counter. Counts an internal clock signal with frequency F_{OSC} /12 (Timer operation) or an external input (event Counter operation).	S:8Bh S:8Dh
TL2 TH2	**Timer 2 registers.** Used in cascade as one 16–bit Counter. Counts an internal clock signal with frequency F_{OSC} /12 (Timer operation) or an external input (event Counter operation)	S:CCh S:CDh
TCON	**Timer 0/1 Control register** Contains the run control bits, overflow flags, interrupt flags and interrupt type control bits for Timer 0 and Timer 1.	S:88h
T2CON	**Timer 2 Control register** Contains the receive clock, transmit clock and capture/reload bits used to configure Timer 2. Also contains the run control bit, Counter/Timer select bit, overflow flag, external flag and external enable for Timer 2.	S:C8h

Table 3.1. Timer/Counter SFRs

Mnemonic	Description	Address
TMOD	**Timer 0/1 Mode Control register** Contains the mode select bits, Counter/Timer select bits and external control gate bits for Timer 0 and Timer 1.	S:89h
T2MOD	**Timer 2 Mode Control Register.** Contains the Timer 2 output enable and down count enable bits.	S:C9h
RCAP2L RCAP2H	**Timer 2 Reload/Capture registers** Provide values to and receive values from the Timer registers (TL2,TH2).	S:CAh S:CBh

Table 3.2. describes the external signals referred to in this chapter.

Table 3.2. 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 registers 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

3.2. 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 3.10. or Figure 3.11.) 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.

3.3. Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 3.1. , Figure 3.3. ,Figure 3.3. and Figure 3.4. show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (See Figure 3.5.) and bits 0, 1, 4 and 5 of TCON register (See Figure 3.4.). 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.

3.3.1. 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 3.1.). The

upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments TH0 register.

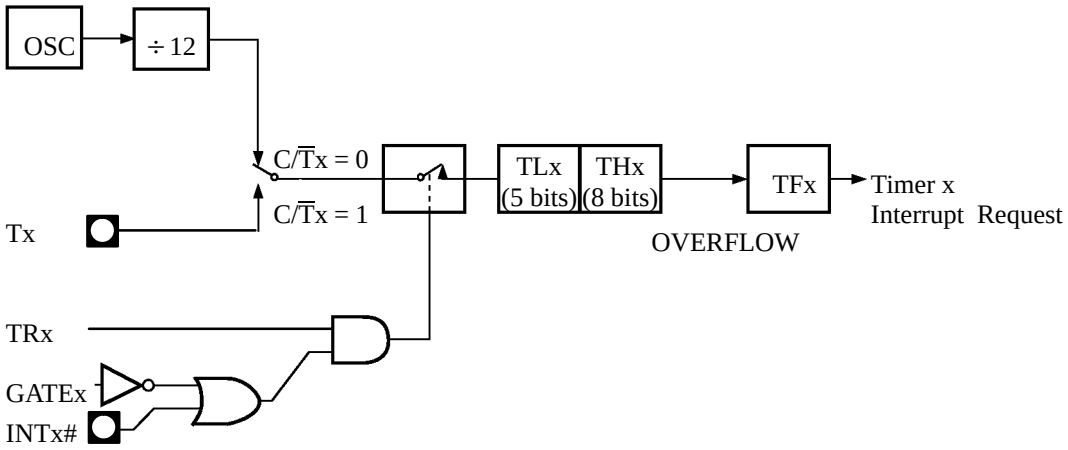


Figure 3.1. Timer/Counter x (x = 0 or 1) in Mode 0

3.3.2. 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 3.3.). The selected input increments TL0 register.

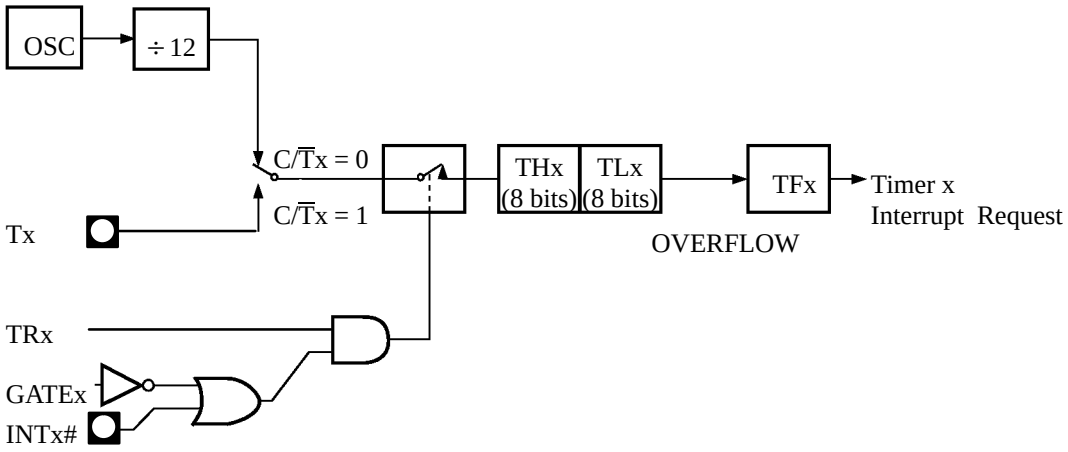


Figure 3.2. Timer/Counter x (x = 0 or 1) in Mode 1

3.3.3. 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 3.3.). 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.

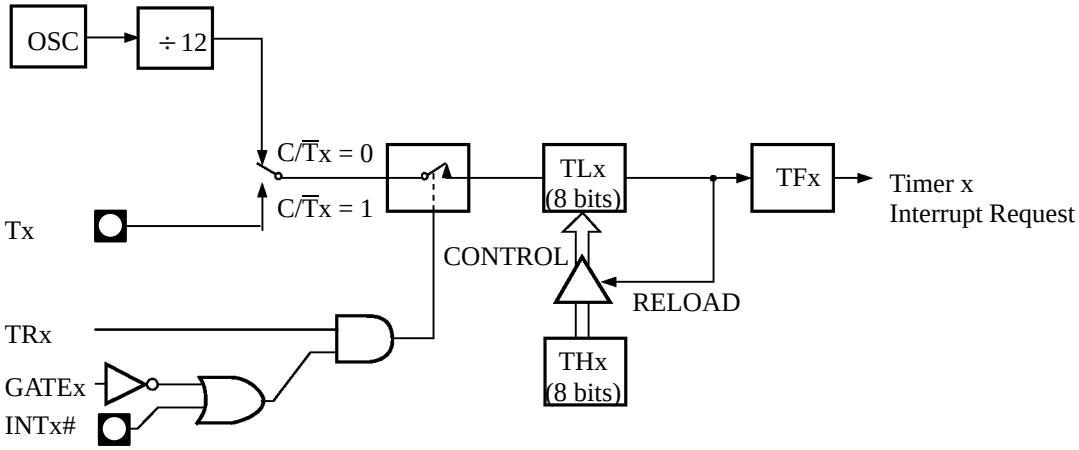


Figure 3.3. Timer/Counter x (x = 0 or 1) in Mode 2

3.3.4. 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 3.3.). 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.

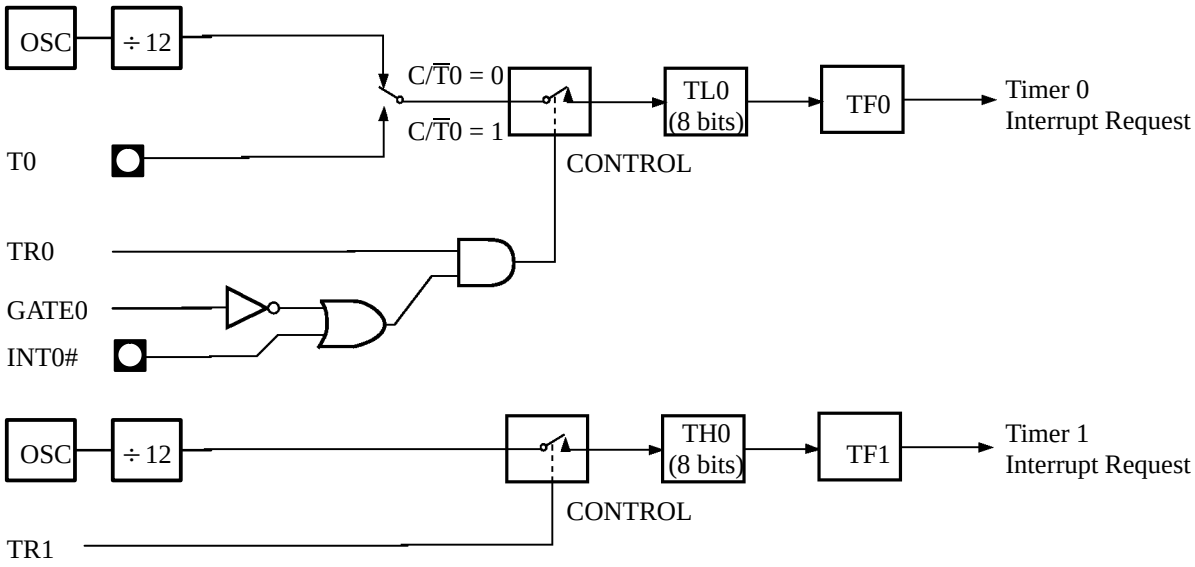


Figure 3.4. Timer/Counter 0 in Mode 3 : Two 8-bit Counters

3.4. Timer 1

Timer 1 functions as either a Timer or event Counter in three modes of operation. Figure 3.1. and Figure 3.3. 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 3.5.) and bits 2, 3, 6 and 7 of TCON register (See Figure 3.4.). 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 operation in modes 0, 1 and 2 is identical to Timer 0. 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.

3.4.1. 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 3.1.). The upper 3 bits of TL1 register are ignored. Prescaler overflow increments TH1 register.

3.4.2. 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 3.2.). The selected input increments TL1 register.

3.4.3. 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 3.3.). 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.

3.4.4. 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.

3.5. 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 3.10.) and T2CON register (See Figure 3.12.).
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 3.3. 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 TF2 allows TL2 to be incremented by the selected input.

The operating modes are described in the following paragraphs. Block diagrams in Figure 3.7. through Figure 3.8. show the Timer 2 configuration for each mode.

Table 3.3. Timer 2 Modes of Operation

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

3.5.1. 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 (See Figure 3.12.). At device reset, DCEN is cleared, so in the auto–reload mode, Timer 2 defaults to operation as an up Counter.

3.5.1.1. Up Counter Operation

When DCEN = 0, Timer 2 operates as an up Counter (Figure 3.6.). 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.

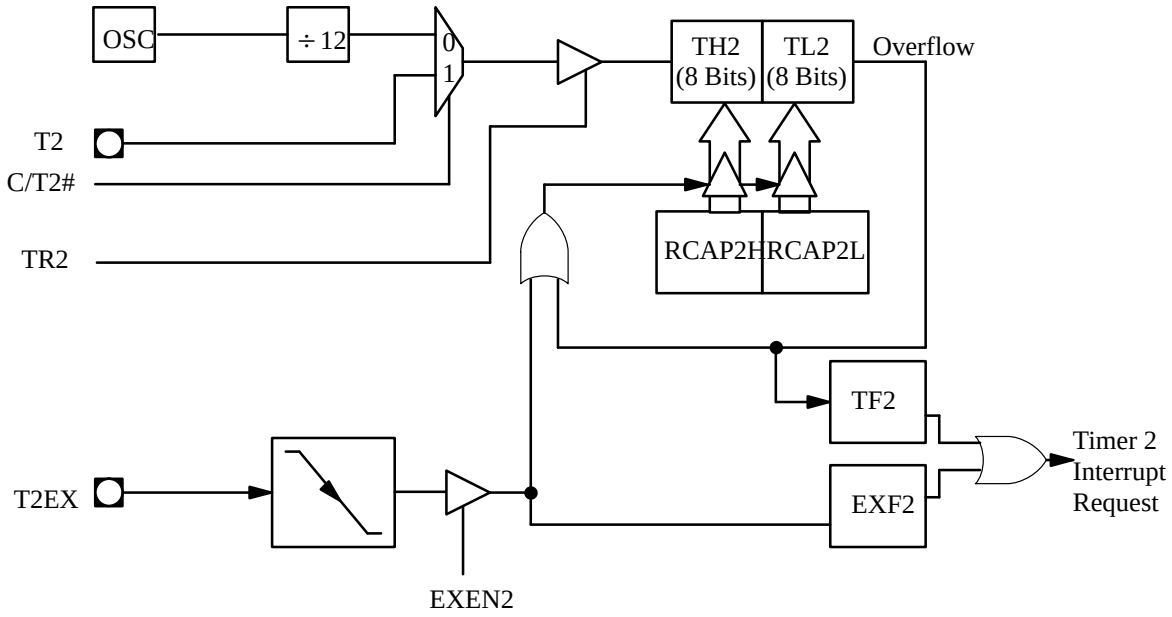


Figure 3.5. Timer 2: Auto Reload Mode Up Counter (DCEN = 0)

3.5.1.2. Up/Down Counter Operation

When DCEN = 1, Timer 2 operates as an up/down Counter (See Figure 3.8.). 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 changing 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.

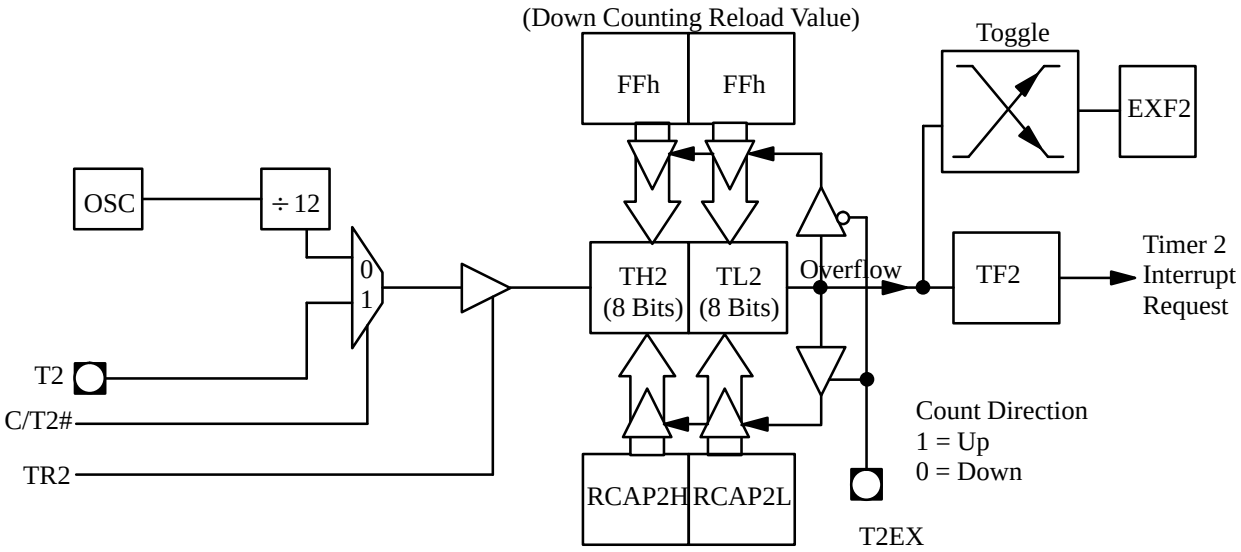


Figure 3.6. Timer 2: Auto Reload Mode Up/Down Counter (DCEN = 1)

3.5.2. Capture Mode

In the capture mode, Timer 2 functions as a 16–bit Timer or Counter (See Figure 3.5.). 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.

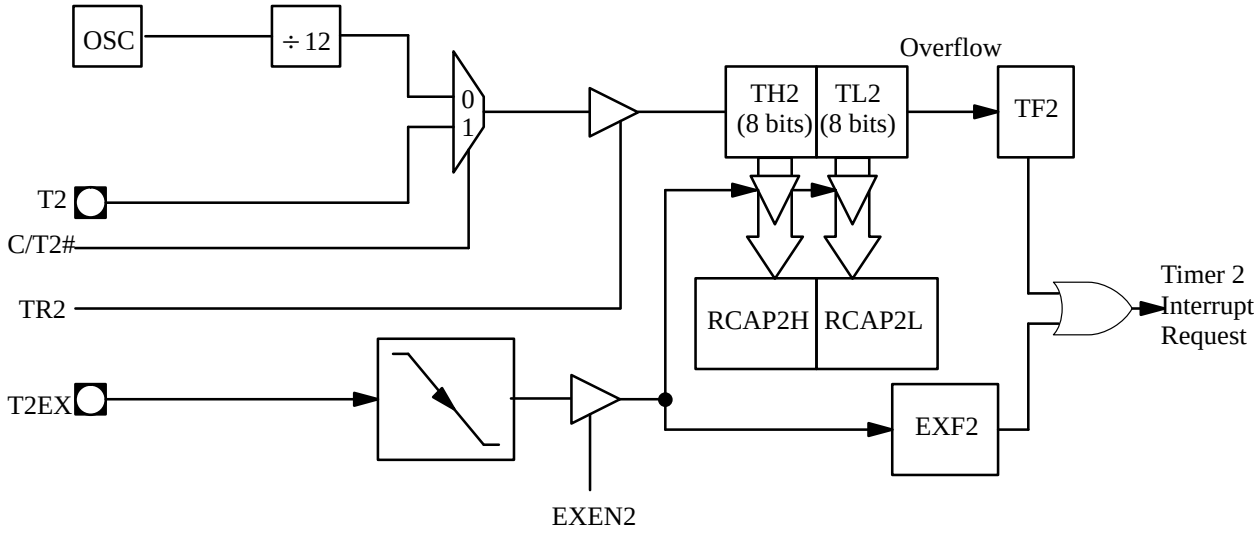


Figure 3.7. Timer 2: Capture Mode

3.5.3. 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 4.7.3.3. for more information on this mode.

3.5.4. Clock–Out Mode

In the clock–out mode, Timer 2 operates as a 50%–duty–cycle, programmable clock generator (See Figure 3.9.). The input clock increments TL0 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:

$$\text{Clock-Out Frequency} = \frac{F_{OSC}}{4 \times (65535 - \text{RCAP2H/RCAP2L})}$$

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.

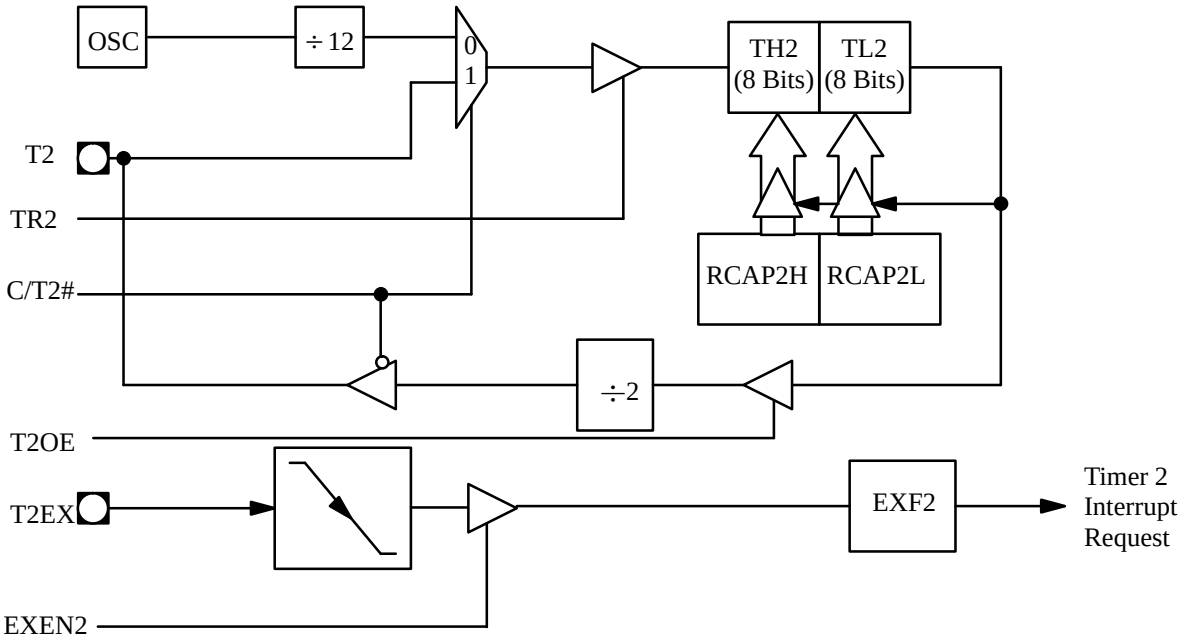


Figure 3.8. Timer 2: Clock Out Mode

3.6. Registers

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

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

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 0. Set to turn on Timer 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 out INT1# pin.
2	IT1	**Interrupt 1 Type Control bit** Clear to select low level active (level triggered) for external interrupt 1. 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 out INT0# pin.
0	IT0	**Interrupt 0 Type Control bit** Clear to select low level active (level triggered) for external interrupt 0. Set to select falling edge active (edge triggered) for external interrupt 0.

Reset value = 0000 0000b

Figure 3.9. TCON Register

TMOD (S:89h)
Timer/Counter Mode register

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

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 4	M11 M01	**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.
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 0	M10 M00	**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.

Reset value = 0000 0000b

Figure 3.10. TMOD Register

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

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

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 3.11. T2CON Register

T2MOD (S:C9h)
Timer/Counter 2 Control register

–	–	–	–	–	–	T2OE	DCEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
6	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
5	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
4	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
3	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
2	–	**Reserved** Must be set by software. Do not clear this bit. The value read from this bit is indeterminate.
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 3.12. T2MOD Register

Serial I/O Port

4.1. 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 4.1. and the Serial Port special function registers are described in Table 4.2. and detailed at the end of this chapter. Figure 4.1. is the Serial Port block diagram.

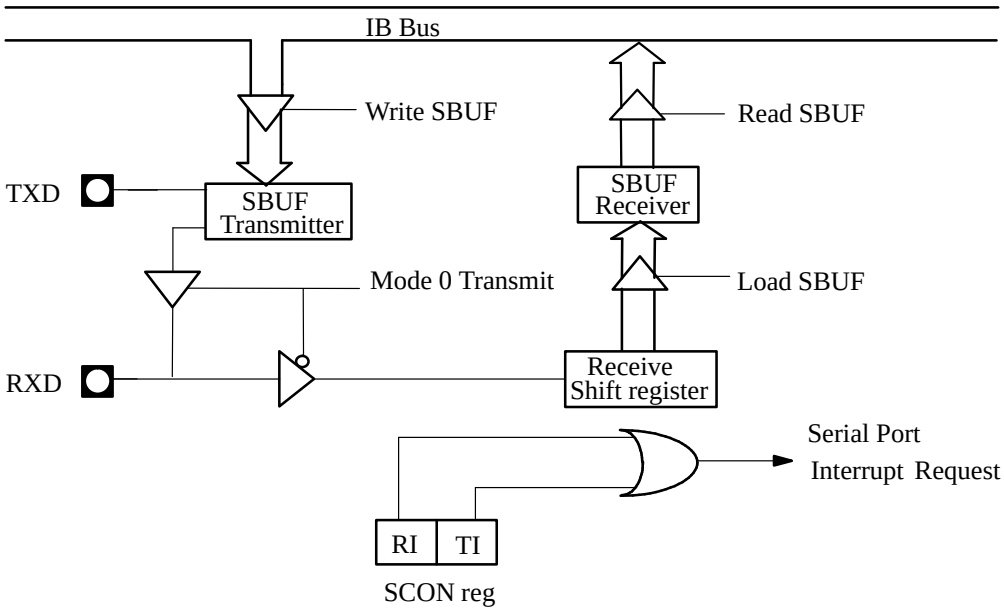


Figure 4.1. Serial Port Block Diagram

Table 4.1. 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 on the TXD pin and sends and receives messages on the RXD pin (See Figure 4.1.). 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. However, if software has not read the first byte by the time the second byte is received, the second byte will overwrite the first. The UART sets interrupt bits TI and RI on transmission and reception, respectively. These two bits share a single interrupt request and interrupt vector.

Table 4.2. Serial Port SFRs

Mnemonic	Description	Address
BDRCON	**Baud Rate Control register** Enables and configures the internal Baud Rate register.	S:9Bh
BRL	**Baud Rate Reload register** Contains the auto–reload value of the Baud Rate Generator.	S:9Ah
SADDR	**Serial Address register** Defines the individual address for a slave device connected on the serial lines.	S:A9h
SADEN	**Serial Address Enable register** Specifies the mask byte that is used to define the given address for a slave device.	S:B9h
SBUF	**Serial Buffer** Two separate registers comprise the SBUF register. Writing to SBUF loads the transmit buffer and reading SBUF accesses the receive buffer.	S:99h
SCON	**Serial Port Control register** Selects the Serial Port operating mode. SCON enables and disables the receiver, framing bit error detection, overrun error detection, multiprocessor communication, automatic address recognition and the Serial Port interrupt bits.	S:98h

4.2. Modes of Operation

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

4.2.1. 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 4.2. shows the timing for transmission and reception in mode 0.

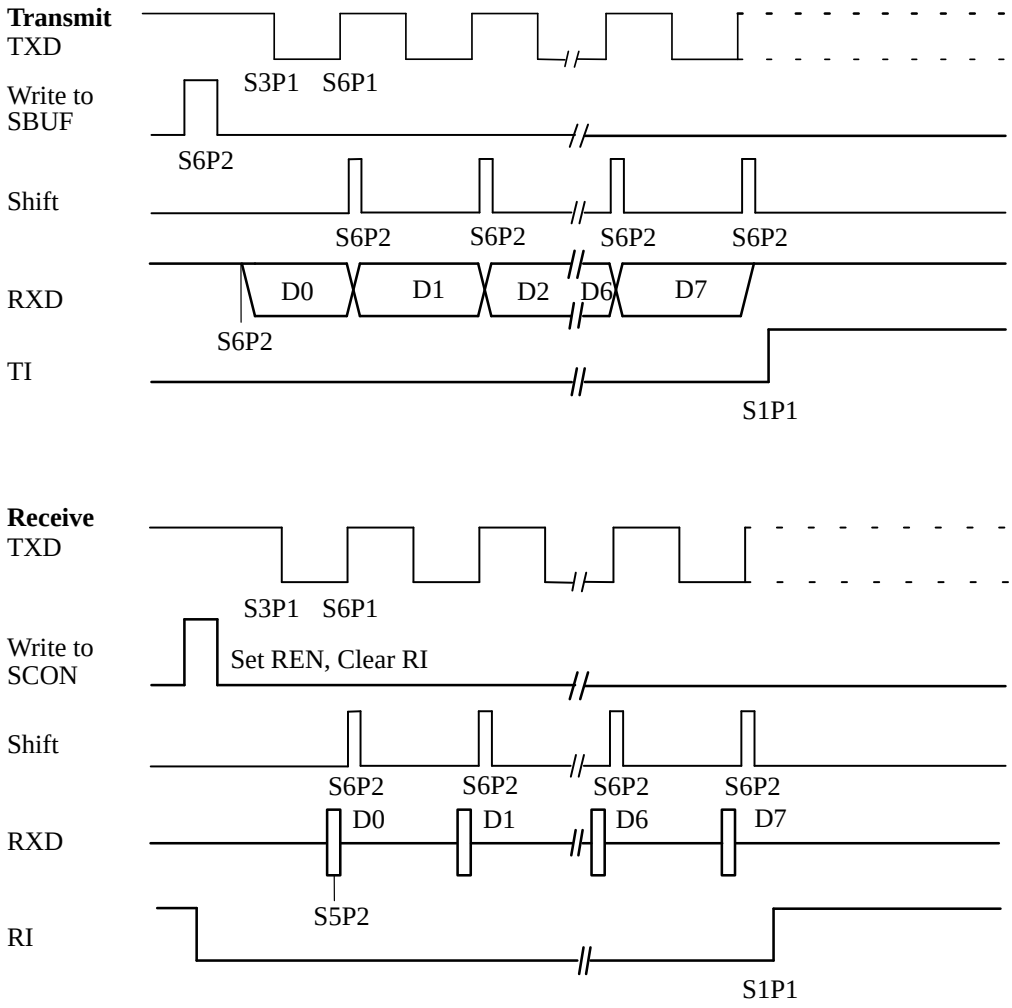


Figure 4.2. Mode 0 Timings

4.2.1.1. Transmission (Mode 0)

Follow these steps to begin a transmission:

© Write to SCON register clearing bits SM0, SM1 and REN.

© 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.

4.2.1.2. 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 4.2.). In the second peripheral cycle clock–signal pulse, and 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 LSB (D7) is shifted into the shift register, and hardware asserts RI bit to indicate acompleted reception. Software can then read the received byte from SBUF register.

4.2.2. 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. The data frame (See Figure 4.3.) 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 “Baud Rate Generator” paragraph).

© Modes 2 and 3

Modes 2 and 3 are full–duplex, asynchronous modes. The data frame (See Figure 4.3.) 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.

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.

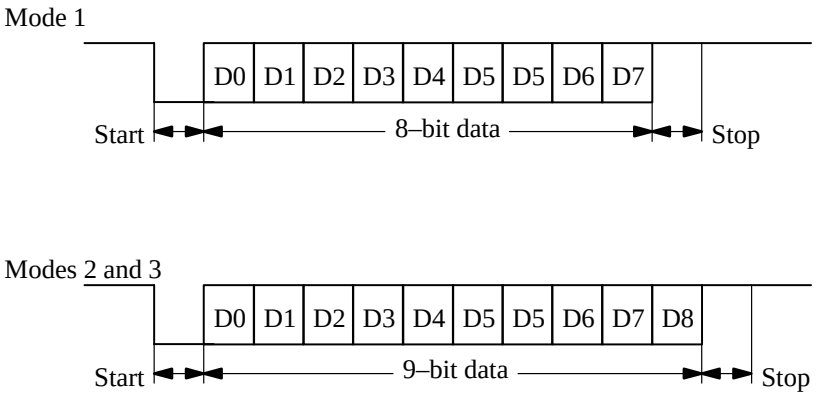


Figure 4.3. Data Frames (Modes 1, 2 and 3)

4.2.2.1. 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 and clear REN bit. 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.

4.2.2.2. 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.

4.3. 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.

4.4. 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 data received and not read by the CPU is overwritten by a new one. Figure 4.4. shows an example of Overrun Error.

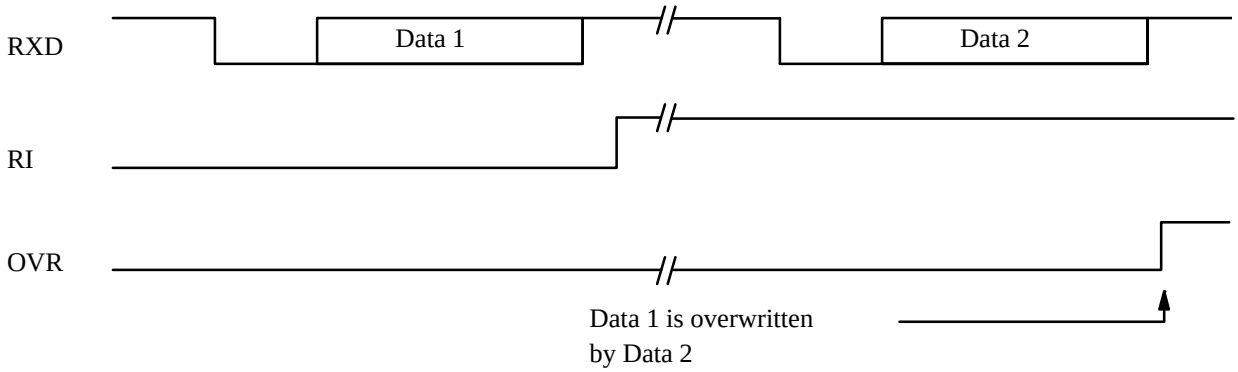


Figure 4.4. Overrun Error (Modes 1, 2 and 3)

In this example Data 1 is received and RI is set. Then a Data 2 is sent before the CPU has read the first one. Data 1 is overwritten by Data 2 and the Overrun Error bit (OVR) is set in SCON register to indicate the error.

4.5. 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 TSC80251G1 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.

4.6. 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).

4.6.1. 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 mores 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`).

4.6.2. 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 can send and address FBh.

4.6.3. 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.

4.7. Baud Rates

The Baud Rate Control register (BDRCON, see Figure 4.9.) is added to the TSC80251G1 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.

4.7.1. 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 4.5.

© 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 = Fosc/[4×(256–BRL)]
When SPD = 0, the Slow mode is selected: Baud_Rate = Fosc/[24×(256–BRL)].

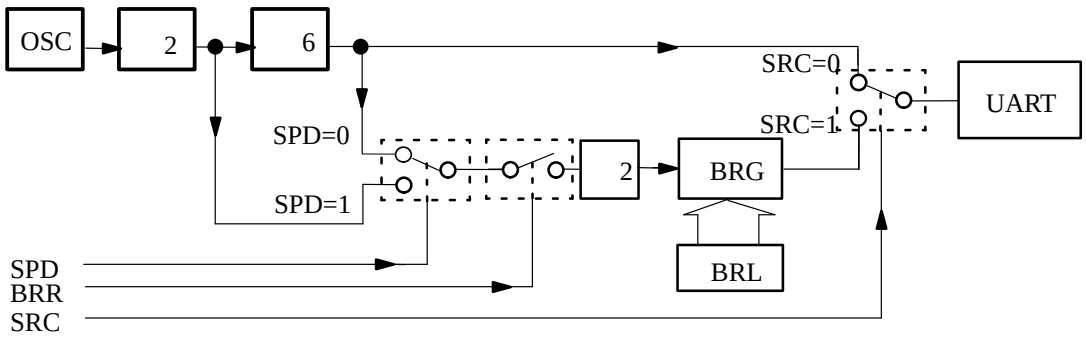


Figure 4.5. Clock Transmission Sources in Mode 0

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.

4.7.2. 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:

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

The configuration is shown in Figure 4.6.

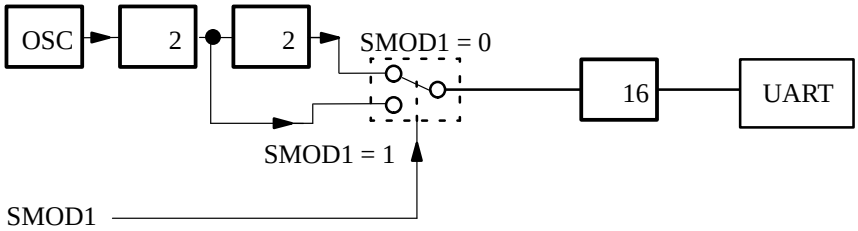


Figure 4.6. UART in Mode 2

4.7.3. 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.

4.7.3.1. Baud Rate Selection

The Baud Rate Generator for transmit and receive clocks can be selected separately via the BDRCON register (See Figure 4.11.).

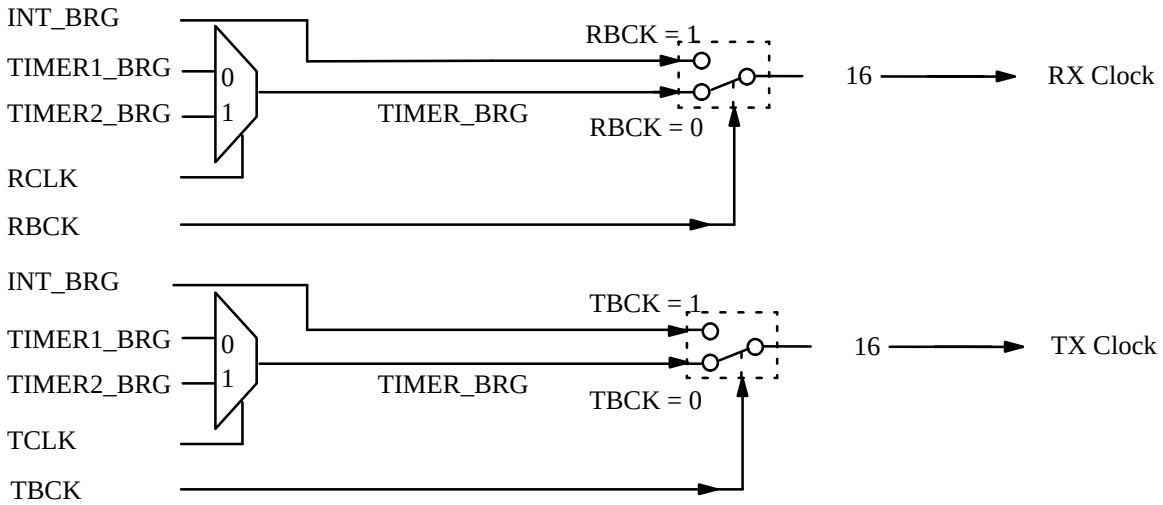


Figure 4.7. Baud Rate Generator Selection

4.7.3.2. 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:

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

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

The configuration is shown in Figure 4.8.

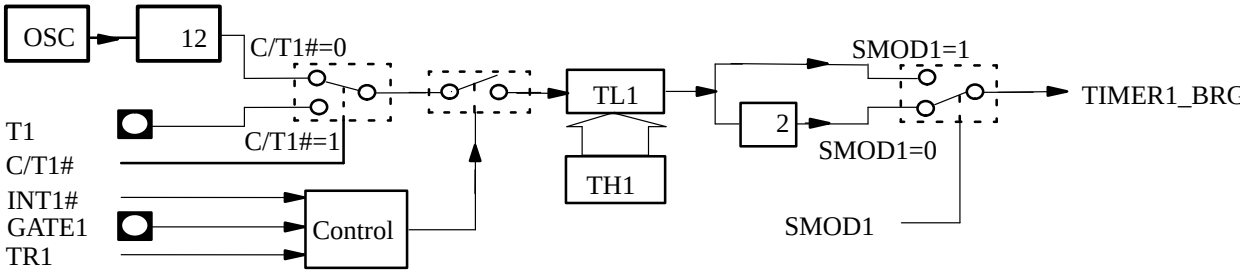


Figure 4.8. Timer 1 as Baud Rate Generator in Modes 1 and 3

Table 4.3. 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
150	0	64	0	0	48	0.16

Table 4.4. Timer 1 Generated Baud Rates at 16 MHz

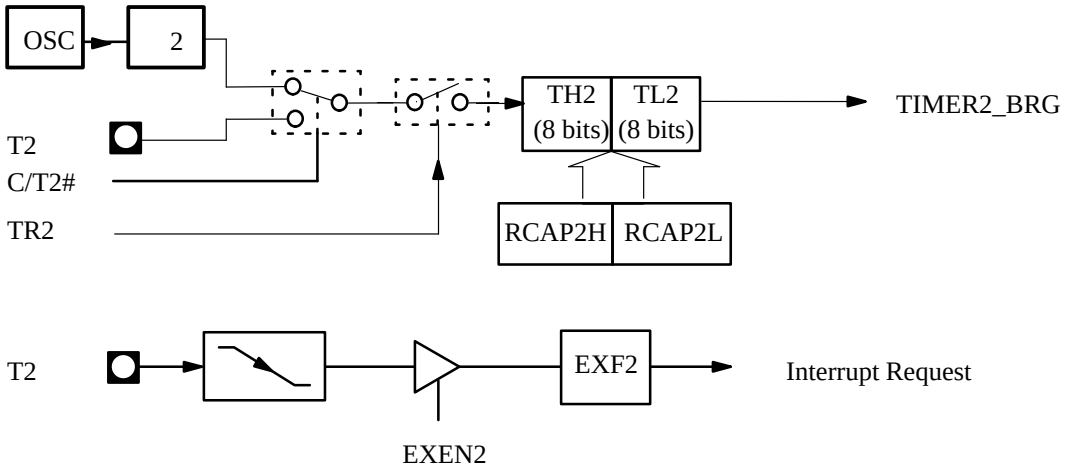
Baud Rates	F_{OSC} = 14.7456 MHz			F_{OSC} = 16 MHz		
	SMOD1	TH1	Error (%)	SMOD1	TH1	Error (%)
57600	–	–	–	–	–	–
38400	0	255	0	–	–	–
28800	–	–	0	–	–	–
19200	0	254	0	–	–	–
9600	0	252	0	1	247	3.55
4800	0	248	0	1	239	2.12
2400	0	240	0	1	221	0.79
1200	0	224	0	1	187	0.64
600	0	192	0	1	117	0.08
300	0	128	0	0	117	0.08
150	0	0	0	–	–	–

4.7.3.3. 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). You can use the T2EX pin 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 clear in T2CON register).



Note availability of additional external interrupt.

Figure 4.9. 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:

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

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

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 4.5. 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	6552	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
150	63232	0	63036	0

Table 4.6. 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
150	62464	0	62203	0.01

4.7.3.4. 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

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

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

© SPD = 0 (Default Mode)

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

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

The configuration is shown in the Figure 4.10.

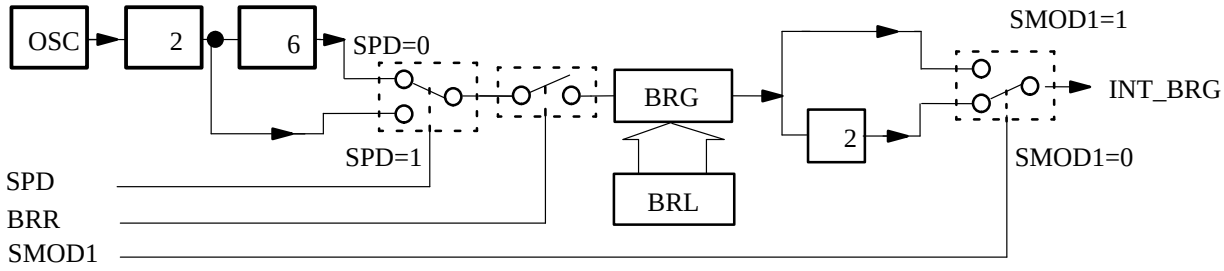


Figure 4.10. Internal Baud Rate Generator in Modes 1 and 3

Table 4.7. Internal Baud Rate Generator at 12 MHz

Baud Rate	F_{OSC} = 11.0592 MHz				F_{OSC} = 12 MHz			
	SPD	SMOD	BRL	Error (%)	SPD	SMOD	BRL	Error (%)
115200	1	0	253	0	–	–	–	–
57600	0	1	255	0	–	–	–	–
38400	1	0	247	0	1	0	246	2.34
28800	0	0	255	0	1	0	243	0.16
19200	0	1	253	0	1	0	236	2.34
9600	0	0	253	0	1	0	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
150	0	0	64	0	0	0	48	0.16

Table 4.8. Internal Baud Rate Generator at 16 MHz

Baud Rates	F_{OSC} = 14.7456 MHz				F_{OSC} = 16 MHz			
	SPD	SMOD	BRL	Error (%)	SPD	SMOD	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
150	–	–	–	–	–	–	–	–

4.8. Registers

BDRCON (S:9Bh)
Baud Rate Control register

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

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 Baud Rate Set to start the Baud Rate
3	TBCK	**Transmission Baud Rate Generator Selection bit** Clear to select Timer 1 for the Baud Rate Generator Set to select Internal Baud Rate Generator
2	RBCK	**Reception Baud Rate Generator Selection bit** Clear to select Timer 1 for the 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 when SRC = 0 Set to select the FAST Baud Rate Generator when SRC = 1
0	SRC	**Baud Rate Source select bit in Mode 0** Clear to select $F_{OSC}/12$ as the Baud Rate Generator (fixed transmission clock). Set to select the internal Baud Rate Generator.

Reset value = XXX0 0000b

Figure 4.11. BDRCON Register

BRL (S:9Ah)
Baud Rate Reload register

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

Reset value = 0000 0000b

Figure 4.12. BRL Register

SADDR (S:A9h)
Serial Address register

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

Reset value = 0000 0000b

Figure 4.13. SADDR Register

SADEN (S:B9h)
Serial Address Enable register

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

Reset value = 0000 0000b

Figure 4.14. SADEN Register

SBUF (S:99h)
Serial Buffer register

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

Reset value = XXXX XXXXb

Figure 4.15. SBUF Register

SCON (S:98h)
Serial Control register

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

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, clear SMOD0 bit in PCON register. Software writes to bits SM1 and SMO to select the Serial Port operating mode. SMO SM1 Mode Description Baud Rate 0 0 0 Shift register $F_{OSC}/12$ or variable if SRC bit BDRCON register 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 enable transmission. Set to enable reception.
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 4.16. SCON Register

Event and Waveform Controller

5.1. 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 TSC80251G1 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.

© Each module 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)

5.2. PCA Mode

5.2.1. Timers/Counters

Figure 5.1. 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.

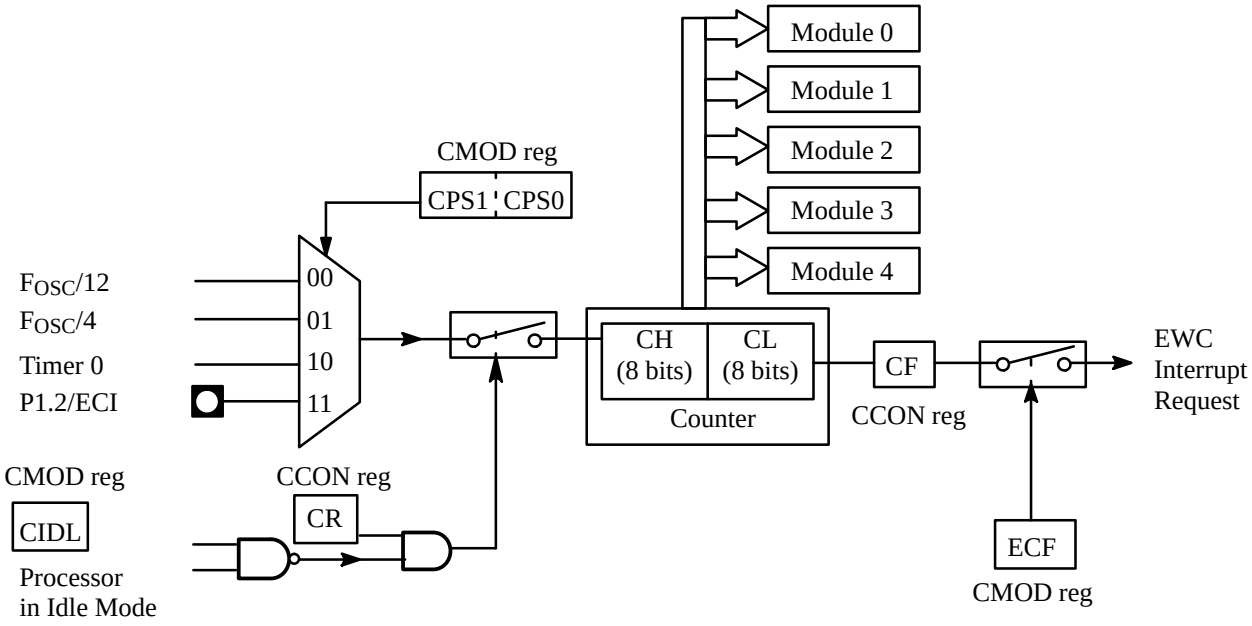


Figure 5.1. EWC Counter in PCA Mode

CPS1 and CPS0 bits in CMOD register select one of four signals as the input to the Counter (See Figure 5.1.):

© 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 5.1.) 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.

5.2.2. Compare/Capture Modules

Each Compare/Capture module is made up of a Compare/Capture register pair (CHx/CLx; x = 0, 1, 2, 3 or 4), 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. NO TAGprovides bit definition and Table 5.1. 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 5.1. PCA Module Modes (x = 0, 1, 2, 3, 4)

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 = indetermined.
3. For the PCA WDT mode, set also WDTE bit in CMOD register to enable the reset output signal (Module 4 only).

5.2.2.1. 16-bit Capture Mode

The Capture mode (See Figure 5.2.) 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 5.1. 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.

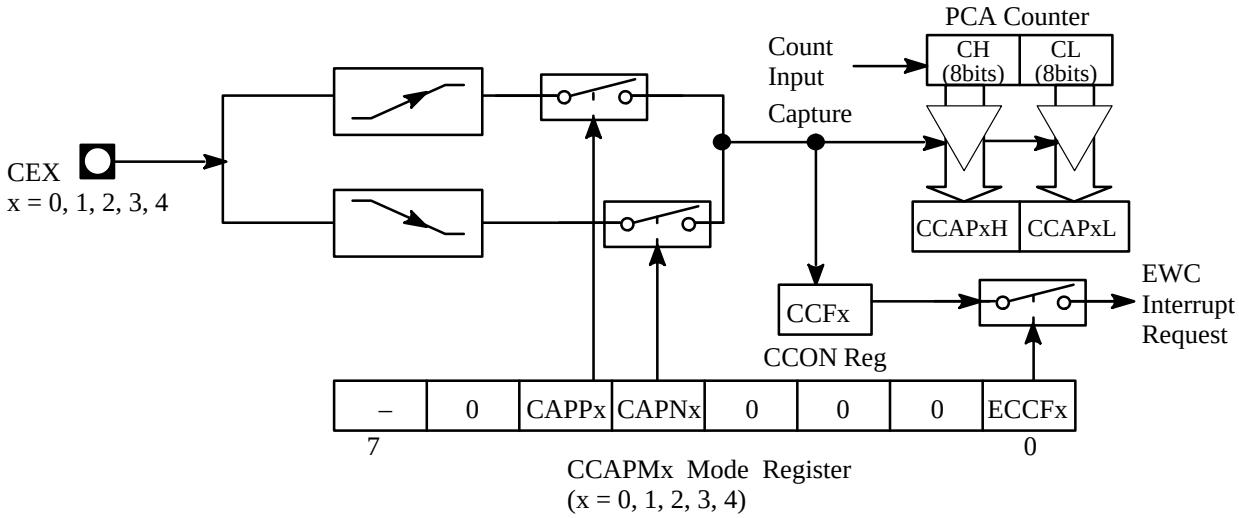


Figure 5.2. PCA 16–bit Capture Mode

5.2.2.2. 16–bit Software Timer Mode

To program a Compare/Capture module for the 16–bit software Timer mode (See Figure 5.3.), set the ECOMx and MATx bits in the module's CCAPMx register. Table 5.1. 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).

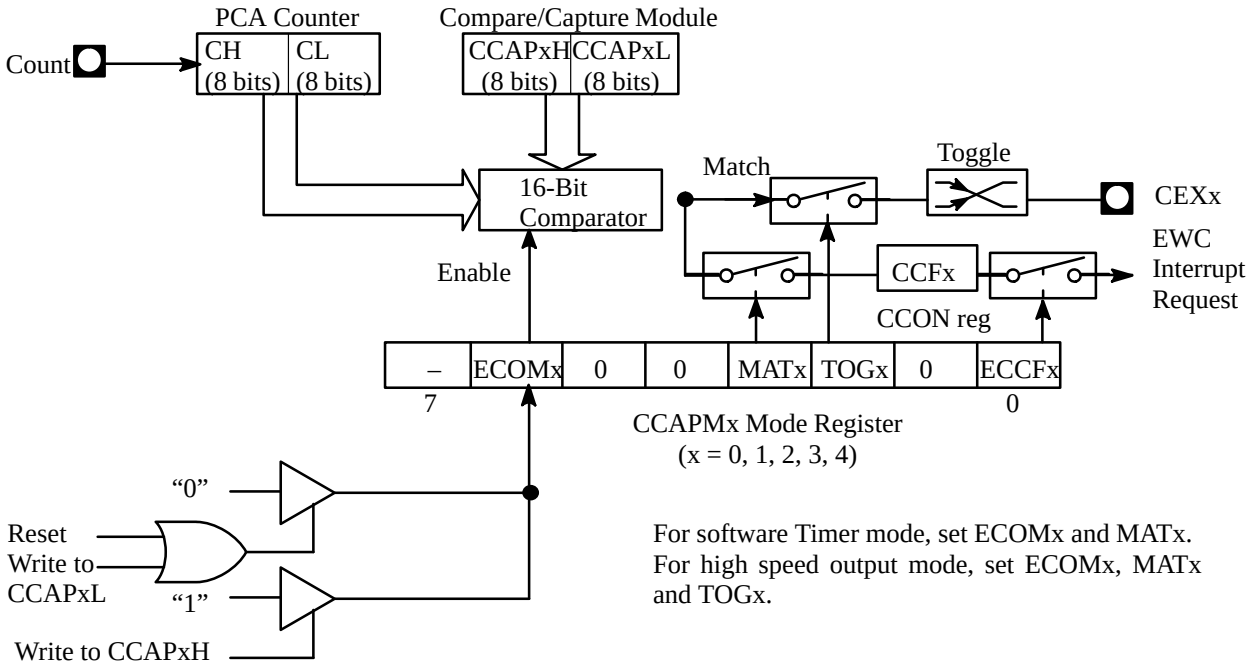


Figure 5.3. PCA Software Timer and High–Speed Output Modes

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.

5.2.2.3. High-Speed Output Mode

The high–speed output mode (See Figure 5.3.) 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 5.1. 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 (CCFx in CCON 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.

5.2.2.4. 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.

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 5.4.). When not programmed as a WDT, it can be used in the other modes.

To program module 4 for the PCA WDT mode:

- Set ECOM4 and MAT4 bits in CCAPM4 register and WDTE bit in CMOD register. Table 5.1. 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 (See Figure 5.13.).
- 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 Counter bit (CR in CCON register) to start the PCA WDT.
- 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.

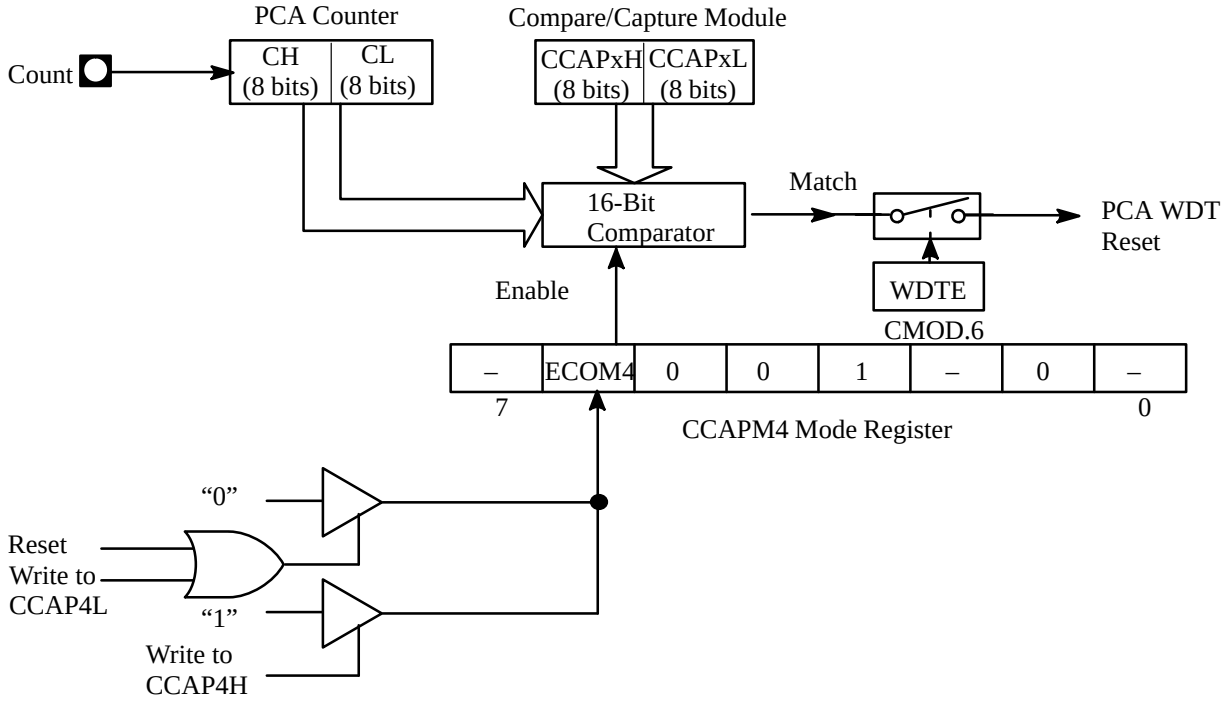


Figure 5.4. PCA Watchdog Timer Mode

5.2.2.5. 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 5.6.). 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 5.6. , 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 5.1. 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 Counter 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} = 16 MHz, this is 15.6 KHz.

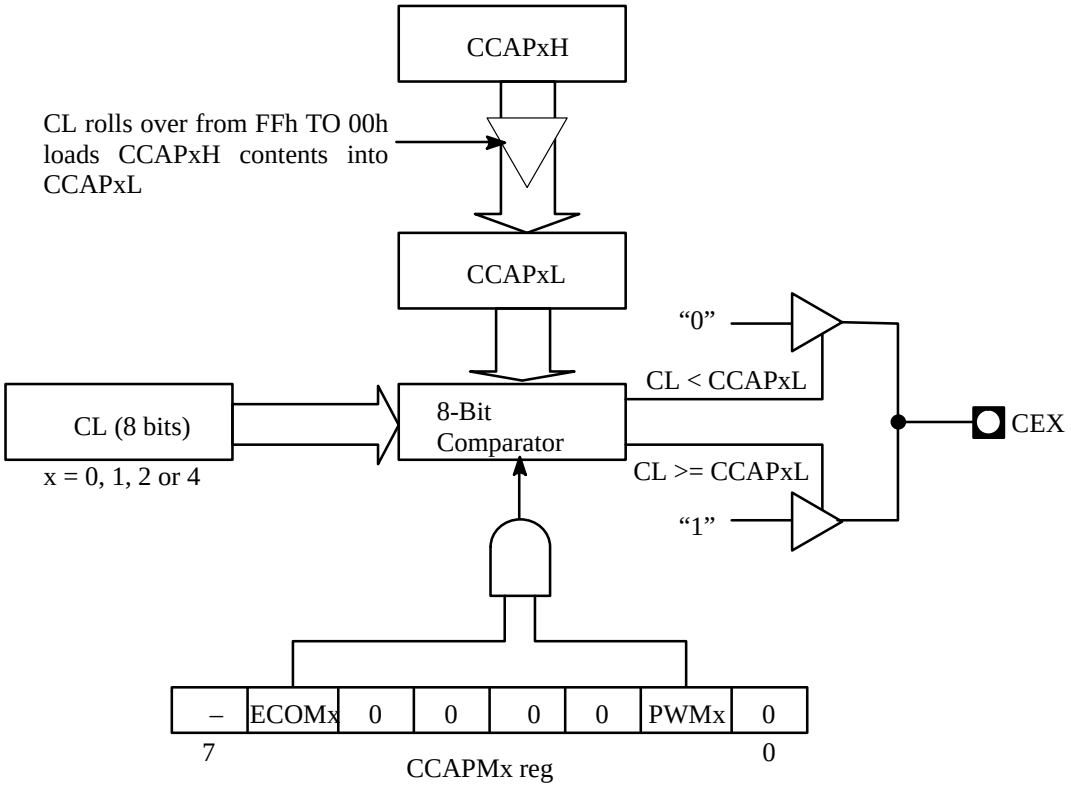


Figure 5.5. PWM Mode

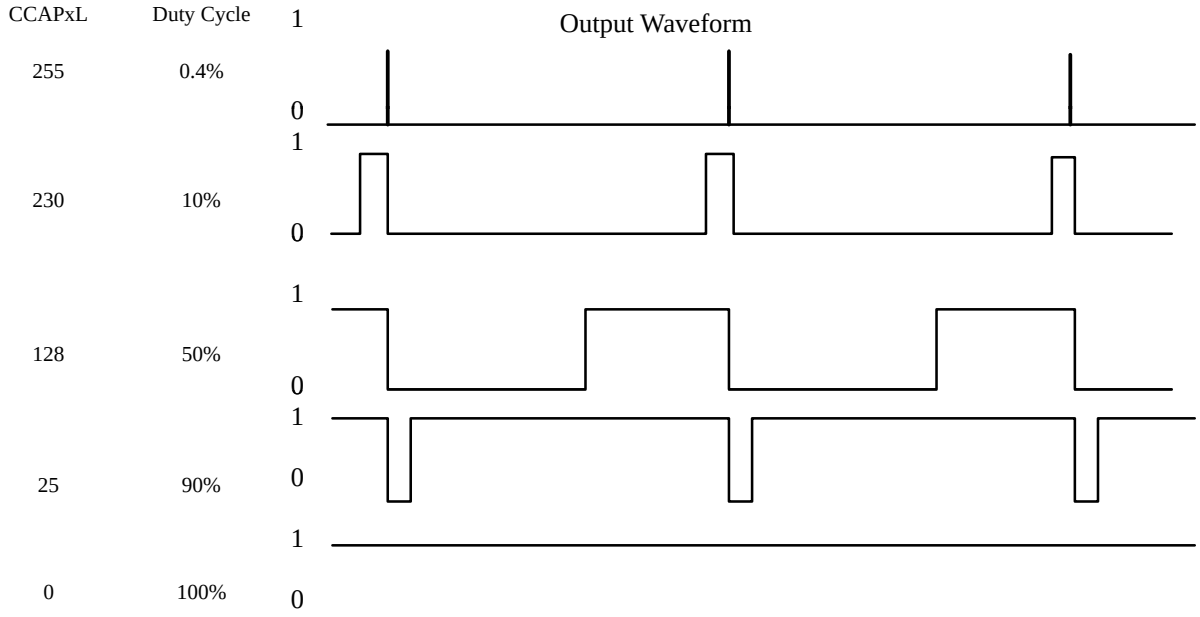


Figure 5.6. PWM Variable Duty Cycle

5.3. Registers

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

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

Reset Value = 0000 0000b

Figure 5.7. EWC CCAPxH Registers (x = 0, 1, 2, 3, 4)

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

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

Reset Value = 0000 0000b

Figure 5.8. EWC CCAPxL Registers (x = 0, 1, 2, 3, 4)

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

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

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 = 1000 0000b

Figure 5.9. EWC CCAPMx Registers (x = 0, 1, 2, 3, 4)

CCON (S:D8h)
Timer/Counter Control register

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

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 ECCF 4 bit in CCAPM 4 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 ECCF 3 bit in CCAPM 3 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 ECCF 2 bit in CCAPM 2 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 ECCF 1 bit in CCAPM 1 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 ECCF 0 bit in CCAPM 0 register is set. Must be cleared by software.

Reset Value = 0010 0000b

Figure 5.10. EWC CCON Register

CH (S:F9h)
Timer/Counter High register

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

Reset Value = 0000 0000b

Figure 5.11. EWC CH Register

CL (S:E9h)
Timer/Counter Low register

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

Reset Value = 0000 0000b

Figure 5.12. EWC CL Register

CMOD (S:D9h)
Counter Mode register

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

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, Fosc/12 0 1 Internal Clock, Fosc/4 1 0 Timer 0 overflow 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/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 = 0011 1000b

Figure 5.13. EWC CMOD Register

SSLC / Inter–Integrated Circuit (I^2C) Interface

6.1. Introduction

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

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

When I^2C interface is selected, SPI is no longer available. This section describes the I^2C interface. The I^2C bus is a bi–directional two–wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (I^2C) control. The bus is made of two lines: oneSerial Clock (SCL) and one Serial Data (SDA) that carry information between the ICs connected to them. The serial data transfer is limited to 100kbit/s in basic mode. Various communication configuration can be designed using this bus; however, the TSC80251G1 implements only the two basic Master transfer modes without multimaster capability. Figure 6.1. shows a typical I^2C bus configuration.

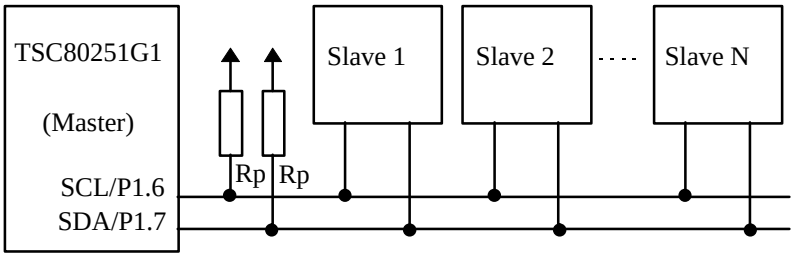


Figure 6.1. Typical I^2C Bus Configuration using the TSC80251G1

The CPU interfaces to the I^2C logic via the following three 8–bit Special Function Registers (SFR): the Synchronous Serial Control register (SSCON, see Figure 6.6.), the Synchronous Serial Data register (SSDAT, see Figure 6.8.) and the Synchronous Serial Control and Status register (SSCS, see Figure 6.9.).

SSCON is used to enable the I^2C interface, to program the bit rate (See Table 6.1.), to acknowledge or not a received data, to send a START or a STOP condition on the I^2C bus, and to acknowledge a SSLC interrupt. An hardware reset disables the I^2C interface.

In write mode, SSCS is used to select the I^2C interface and to select the bit rate source. In read mode, SSCS contains a status code which reflects the status of the I^2C logic and the I^2C bus. The three least significant bits are always zero. The five most significant bits contains the status code. There are 11 possible status code. When SSCS contains F8h, no relevant state information is available and no SSLC interrupt is requested. A valid status code is available in SSCS one machine cycle after the Synchronous Serial Interrupt flag (SSI) is set by hardware and is still present one machine cycle after SSI has been reset by software. Table 6.2. to Table 6.4. give the status for the master modes and miscellaneous states.

SSDAT contains a serial data byte 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^2C logic is in a defined

state and SSI 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.

Figure 6.2. shows how a data transfer is accomplished on the I^2C bus.

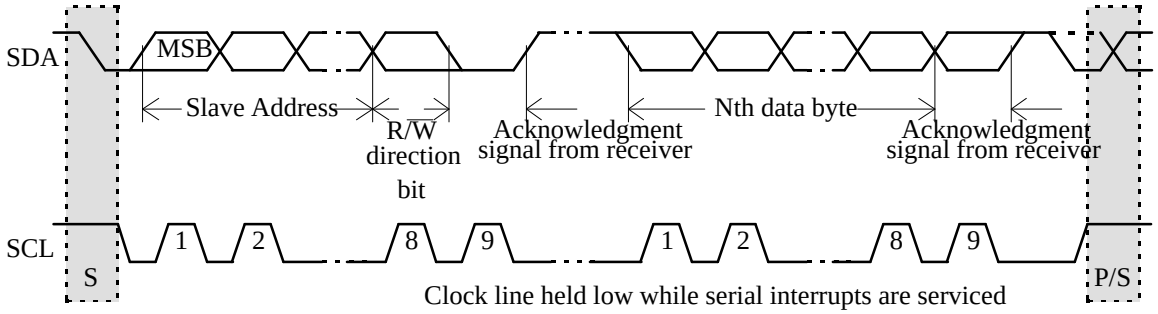


Figure 6.2. Complete data transfer on I^2C bus

The two operating modes are:

- Master Transmitter
- Master Receiver

Data transfer in each mode of operation are shown in Figure 6.3. and Figure 6.4. These figures contain the following abbreviations:

A	Acknowledge bit (low level on SDA)
$\overline{A}$	Not acknowledge bit (high level on SDA)
Data	8–bit data byte
P	Stop condition
MR	Master Reveive
MT	Master Transmit
S	Start condition
SLA	Slave Address
R	Read bit (high level on SDA)
W	Write bit (low level on SDA)

In Figure 6.3. and Figure 6.4. , circles are used to indicate when SSI is set. The numbers in the circles show the status code held in SSCS. At each point, a proper 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 SSI is cleared by software.

When the SSLC 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 6.2. and Table 6.3.

6.1.1. Interface and Bit rate source selection

Before the I^2C interface can be enabled, SSCS must be initialised as follows:

SSBRS	–	–	–	–	–	–	SSM0D
Bit Rate Source	–	–	–	–	–	–	0

SSMOD selects the I^2C interface. The bit rate can be derived from a programmable bit rate generator or from the internal bit rate controller. The Synchronous Serial Bit Rate Selection bit (SSBRS) selects the programmable or predefined bit rates (See Figure 6.9.). When in the programmable bit rate generator configuration, the bit rate depends on the content of the Synchronous Serial Bit Rate register (SSBR, see Figure 6.5.). It is given by the following formula:

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

2

6.1.2. Master transmitter mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 6.3.). Before the master transmitter mode can be entered, SSCON must be initialized as follows:

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

SSCR0, SSCR1 and SSCR2 select one predefined serial bit rate if the programmable bit rate generator is not used. SSPE must be set to enable I^2C interface. SSSTA, SSSTO and SSI must be cleared.

The master transmitter mode may now be entered by setting the SSSTA bit. The I^2C logic will now test the I^2C bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SSI) 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 bit in SSCON) must then be reset 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 two status code in SSCS are possible: 18h and 20h. The appropriate action to be taken for each of these status code is detailed in Table 6.2. 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 6.2. After a repeated START condition (state 10h) I^2C logic may switch to the master receiver mode by loading SSDAT with SLA+R.

6.1.3. Master receiver mode

In the master receiver mode, a number of data bytes are received from a slave transmitter (See Figure 6.4.). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the service routine must load SSDAT with the 7–bit slave address and the data direction bit (SLA+R). **Then SSI must be cleared in SSCON 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 two status code in SSCS are possible: 40h, and 48h. The appropriate action to be taken for each of these status code is detailed in Table 6.3. 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 6.3. After a repeated START condition (state 10h) I^2C logic may switch to the master transmitter mode by loading SSDAT with SLA+W.

The I^2C logic interfaces to the external I^2C bus via two port pins: P1.6/SCL and P1.7/SDA. To allow proper operation of the I^2C interface, the output latches of P1.6 and P1.7 must be set to logic 1 (See section 2.6).

Table 6.1. Serial Clock Rates

SSBRS	SSCR2	SSCR1	SSCR0	Bit frequency (kHz)		F_{OSC} divided by
				F_{OSC}= 12MHz	F_{OSC}= 16MHz	
0	0	0	0	47	62.5	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	960
0	1	0	1	100	–	120
0	1	1	0	–	–	60
0	1	1	1	0.5 < . < 62.5	0.67 < . < 83	96 · (256 – reload value Timer 1) (reload value range: 0–254 in mode 2)
1	0	X	X	11.6< · <100 min SSBR value: 27	15.5< · <100 min SSBR value: 37	4 · (SSBR value + 3)
1	1	X	X	–	–	Reserved

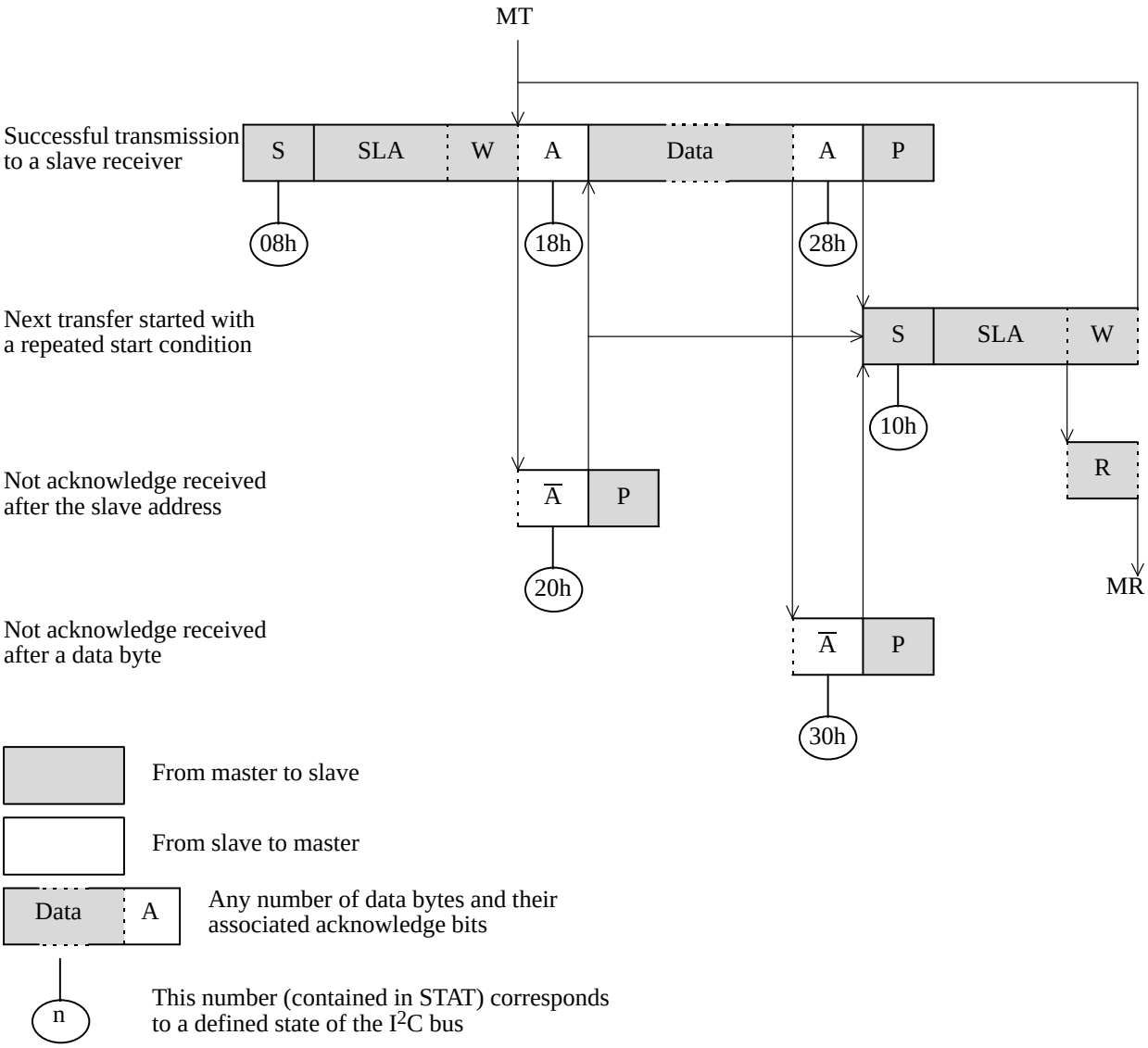


Figure 6.3. Format and States in the Master Transmitter Mode

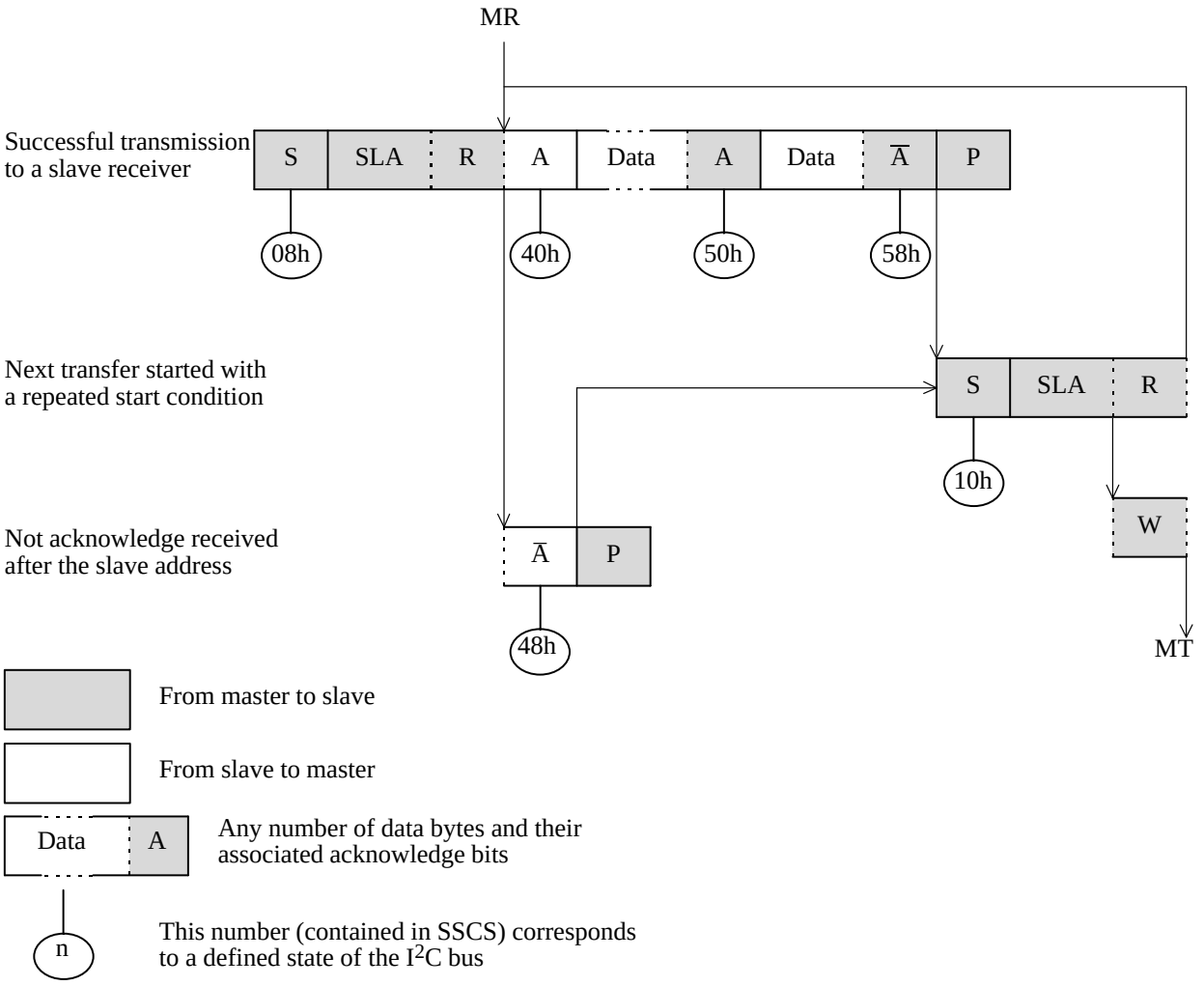


Figure 6.4. Format and state in the master receiver mode

Table 6.2. Status for Master Transmitter Mode

Status Code (SSCS)	Status of the I^2C bus and I^2C hardware	Application software response					Next action taken I^2C hardware
		To/From SSDAT	To SSCON				
			SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted.	Write SLA+W	X	0	0	X	SLA+W will be transmitted then ACK will be transmitted
10h	A repeated START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted then ACK will be transmitted
		Write SLA+R	X	0	0	X	SLA+R will be transmitted then logic will switched 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 then ACK will be 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.
20h	SLA+W has been transmitted; NOT ACK has been received.	Write data byte	0	0	0	X	Data byte will be transmitted then ACK will be 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.
28h	Data byte has been transmitted; ACK has been received.	Write data byte	0	0	0	X	Data byte will be transmitted then ACK will be 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.
30h	Data byte has been transmitted; NOT ACK has been received.	Write data byte	0	0	0	X	Data byte will be transmitted then ACK will be 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.

Table 6.3. Status for Master Receiver Mode.

Status Code (SSCS)	Status of the I^2C bus and I^2C hardware	Application software response: To/From SSDAT	To SSCON: SSSTA	SSSTO	SSI	SSAA	Next action taken I^2C hardware
08h	A START condition has been transmitted.	Write SLA+R	X	0	0	X	SLA+R will be transmitted then ACK will be transmitted
10h	A repeated START condition has been transmitted	Write SLA+R	X	0	0	X	SLA+R will be transmitted then ACK will be transmitted
		Write SLA+W	X	0	0	X	SLA+W will be transmitted then logic will switched to master transmitter mode.
40h	SLA+R has been transmitted; ACK has been received.	No SSDAT action	0	0	0	0	Data byte will be received then NOT ACK will be returned.
		No SSDAT action	0	0	0	1	Data byte will be received then 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 then NOT ACK will be returned.
		Read data byte	0	0	0	1	Data byte will be received then 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 6.4. Status for Miscellaneous States

Status Code (SSCS)	Status of the I^2C bus and I^2C hardware	Application software response: To/From SSDAT	To SSCON: SSSTA	SSSTO	SSI	SSAA	Next action taken I^2C hardware
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. State 00h can also occur when interference causes I^2C logic to enter an undefined state.	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.

6.2. 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= $F_{OSC}/(4\cdot(SSBR\ value+3))$.

Reset value = 00h

Figure 6.5. SSBR register

SSCON (S:93h)
Synchronous Serial Control register (read/write)

SSCR2	SSPE	SSSTA	SSST0	SSI	SSAA	SSCR1	SSCR0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	SSCR2	**Synchronous Serial Control Rate bit 2** See Table 6.1.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Set to enable the I^2C 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** 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, in receiver mode, to force a not acknowledge (high level on SDA). Set, in receiver mode, to force an acknowledge (low level on SDA). This bit has no effect when in transmitter mode.
1	SSCR1	**Synchronous Serial Control Rate bit 1** See Table 6.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** See Table 6.1.

Reset value = 0000 0000b

Figure 6.6. SSCON register

SSCS (S:94h) read
Synchronous Serial Control and Status register

SSSC4	SSSC3	SSSC2	SSSC1	SSSC0	0	0	0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	SSSC4	**Synchronous Serial Status Code bit 4** See Table 6.2. to Table 6.4.
6	SSSC3	**Synchronous Serial Status Code bit 3** See Table 6.2. to Table 6.4.
5	SSSC2	**Synchronous Serial Status Code bit 2** See Table 6.2. to Table 6.4.
4	SSSC1	**Synchronous Serial Status Code bit 1** See Table 6.2. to Table 6.4.
3	SSSC0	**Synchronous Serial Status Code bit 0** See Table 6.2. to Table 6.4.
2	0	**Always zero**
1	0	**Always zero**
0	0	**Always zero**

Reset value = F8h

Figure 6.7. SSCS register: read mode

SSDAT (S:95h)
Synchronous Serial Data register (read/write)

SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	SD7	**Address bit 7 or Data bit 7.**
6	SD6	**Address bit 6 or Data bit 6.**
5	SD5	**Address bit 5 or Data bit 5.**
4	SD4	**Address bit 4 or Data bit 4.**
3	SD3	**Address bit 3 or Data bit 3.**
2	SD2	**Address bit 2 or Data bit 2.**
1	SD1	**Address bit 1 or Data bit 1.**
0	SD0	**Address bit 0 (R/$\overline{W}$) or Data bit 0.**

Reset value = 00h

Figure 6.8. SSDAT register

SSCS (S:94h) write
Synchronous Serial Control and Status register

SSBRS	–	–	–	–	–	–	SSM0D
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Clock Source Selection bit (see Table 6.1.)** Clear to select bit rate controlled by SSCR0 to SSCR2. Set to select external 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	**Synchronous Serial Mode selection bit** Clear to select I^2C interface.

Reset value = 0XXX XXX0b

Figure 6.9. SSCS register: write mode

SSLC / Synchronous Peripheral Interface (μwire/SPI)

7.1. Introduction

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

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

When I^2C is selected, I^2C Interface is no longer available. This chapter describes the SPI. This synchronous interface allows several SPI microcontrollers or μWire and SPI–type peripherals to be interconnected on a bus.

Figure 7.1. shows a typical SPI bus configuration using one TSC80251G1 master and many slaves. 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 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 TSC80251G1 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.

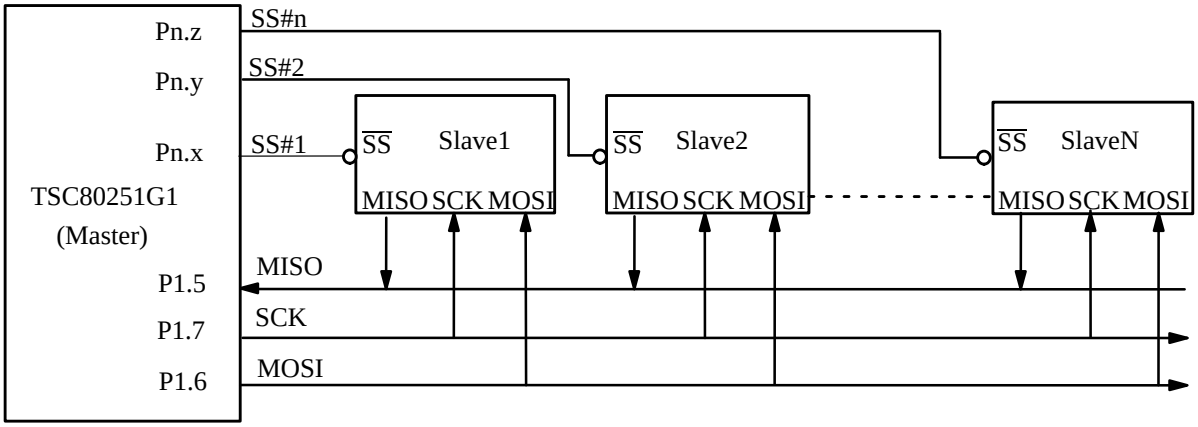


Figure 7.1. Typical SPI Bus Configuration using TSC80251G1

7.2. Description

Figure 7.2. highlights the dedicated harware. MISO is connected to the input of an 8–bit shift register (Serial Synchronous Data, SSDAT) the output of which is connected to MOSI output.

A bit rate generator provides the clock of the shift register and SCK accordingly, depending on the chosen transmission clock policy (See Interface Configuration below). Additional logic is also included to provide control and status information.

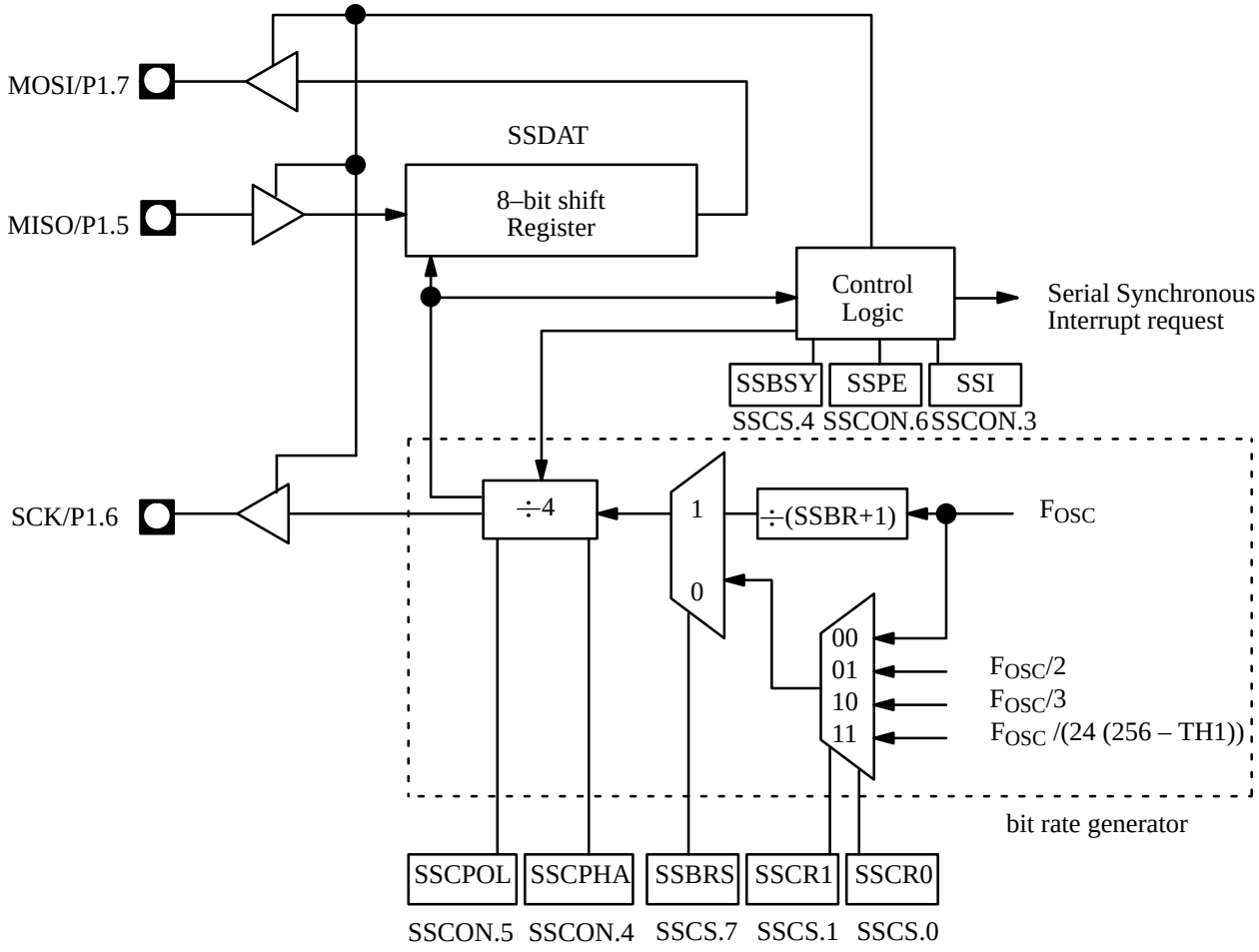


Figure 7.2. Wire and Serial Peripheral Interface

The SPI configuration is made through three registers:

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

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

- SSDAT
- SSCS
- SSCON

7.2.1. Interface Configuration

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

The Synchronous Serial Mode bit (SMOD in SSCS) must be set first to select the μwire/SPI mode. Then the rest of the configuration may be done.

The Synchronous Serial Bit Rate Select bit (SSBRS in SSCS) allows to choose between two bit rate generation modes, see Table 7.1. and Figure 7.2. . When SSBRS is set, the bit rate is programmed according to the Synchronous Serial Bit Rate register 8–bit value (SSBR). When SSBRS is cleared, the bit rate is selected according to the Synchronous Serial Control Rate bits (SSCR1 and SSCR0 in SSCON).

Table 7.1. Serial Clock Rates

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

The Synchronous Serial Polarity bit (SSPOL in SSCON) defines the default SCK line level in idle state (Note: when the peripheral is disabled, the default state is one). Then the Synchronous Serial Phase bit (SSPHA in SSCON) defines the edges on which the MISO input and the MOSI respectively output are sampled and shifted out respectively(See Figure 7.3.).

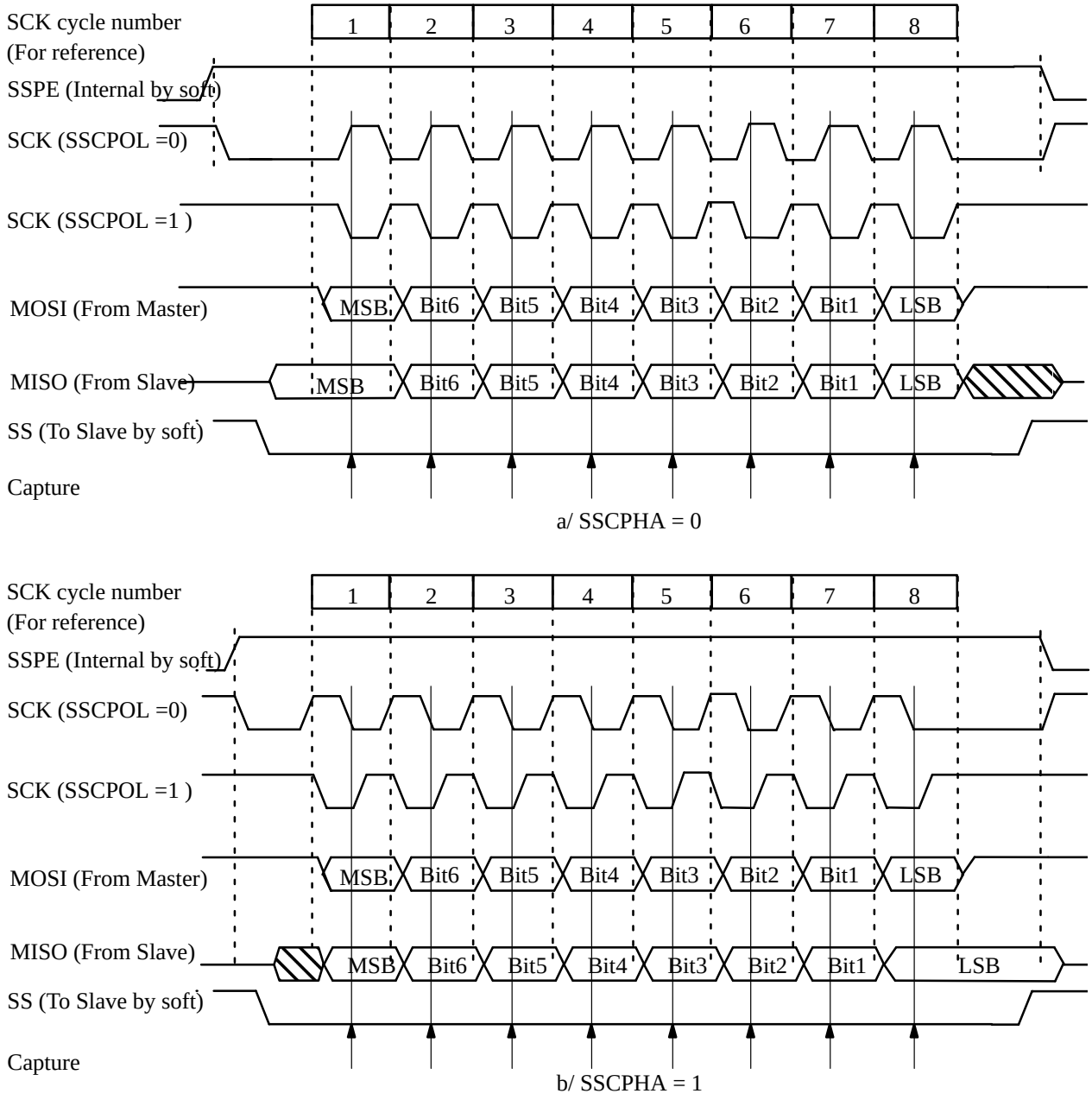


Figure 7.3. Data Transmission (a/ SSCPHA = 0, b/ SSCPHA = 1)

When the peripheral 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 SSPHA and SSPOL bits once the SSPE bit has been set.

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

7.2.2. Data Exchange

There are two possible policies to exchange data:
- polling
- interrupts

To exchange data in polling mode:

1. Ensure the SSLC interrupts are disabled (See IS section).
2. Check SSBSY is not set in SSCS

Loop:

3. Assert SS# (as required)
4. Clear the Synchronous Serial Interrupt bit (SSI) in SSCON
5. Write the transmit data to SSDAT (or jump to next step if there is no data to transmit)
6. Set SSBSY to start exchange
7. Read SSBSY in SSCS until it has been cleared by hardware
8. Read the received data in SSDAT (or jump to next step if there is no data to receive)
9. Deassert SS# (as required)
10. Jump to step 3 if there are more data to transmit or receive, otherwise exit

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.

To exchange data in interrupt mode:

Main program:

1. Check SSBSY is not set in SSCS
2. Clear SSI
3. Ensure the SSLC interrupt service routine is well initialized and enable the SSLC interrupts (See IS section)
4. Assert SS# (as required)
5. Write the transmit data in SSDAT (or jump to next step if there is no data to transmit)
6. Set SSBSY to start exchange
7. Wait

Interrupt service routine:

1. Deassert SS# (as required)
2. Read the received data in SSDAT (or jump to next step if there is no data to receive)
3. Write the transmit data to SSDAT (or jump to next step if there is no data to transmit)
4. Assert SS# (as required)
5. Clear SSI
6. Set SSBSY to continue (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. Then it may be executed at a high priority level preventing burst activity on the SPI bus.

When the SPI is configured, 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.

7.3. Registers

SSBR (S:92h)
Synchronous Serial Bit Rate register (8–bit)

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

Bit Number	Bit Mnemonic	Description
7–0		Bit rate is given by the formula: Br = F_{OSC}/(4.(SSBR value + 1))

Reset Value = 00h

Figure 7.4. SSBR Register

SSCON (S:93h)
Synchronous Serial Control register (read/write)

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

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	SSPE	**Synchronous Serial Peripheral Enable bit** Clear to disable the µWire/SPI interface. Set to enable the µWire/SPI interface.
5	SSCPOL	**Synchronous Serial Clock Polarity bit** Clear to have the clock output set to 0 in iddle state. Set to have the clock output set to 1 in iddle state. Note: when the peripheral is disabled, the clock output is pulled high not to conflict with other functions on the port.
4	SSCPHA	**Synchronous Serial Clock Phase bit** Clear to have the data sampled when the clock leave the iddle state (see SSCPOL) Set to have the data sampled when the clock return to the iddle 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	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	SSCR1	**Synchronous Serial Control Rate bit 1** See Table 7.1.
0	SSCR0	**Synchronous Serial Control Rate bit 0** See Table 7.1.

Reset Value = X000 0000b

Figure 7.5. SSCON Register

SSCS (S:94h)
Synchronous Serial Control and Status register (read/write)

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

Bit Number	Bit Mnemonic	Description
7	SSBRS	**Synchronous Serial Bit Rate Selection bit** Clear to select the bit rate controlled by SSCR1 and SSCR0 (See Table 7.1.). Set to select the programmable bit rate generator (See Table 7.1.).
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	SSBSY	**Synchronous Serial Busy bit** 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	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	SSMOD	**Synchronous Serial Mode selection bit** Set to select the µwire/SPI mode.

Reset Value = 0XXX XXX0b

Figure 7.6. SSCS Register: write mode

SSDAT (S:95h)
Synchronous Serial Data register (8–bit read/write)

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

Reset Value = 0000 0000b

Figure 7.7. SSDAT Register

Hardware Watchdog Timer

8.1. Introduction

The TSC80251G1 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 malfunctions. The WDT described in this chapter is not associated with the PCA Watchdog Timer (See “Event and Waveform Controller” chapter), which may be disabled in software and is less reliable.

8.2. Description

The WDT is a 14–bit counter that counts peripheral cycles, i.e. the system clock divided by twelve ($F_{OSC}/12$).

The Hardware Watchdog Timer register (WDTRST, see Figure 8.1.) provides control access to the WDT. Two operations control the WDT:

- Chip reset clears and disables the WDT.
- Writing a specific two–byte sequence 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. With F_{OSC} = 16 Mhz, a peripheral cycle is 750 ns and the WDT overflows in 750×16384 = 12.288 ms.

The 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.

8.3. 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.

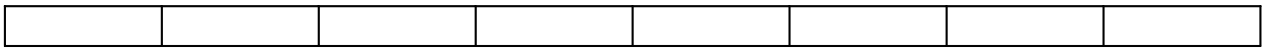
8.4. Hardware WDT during Idle and Power–Down Modes

Operation of the WDT during power reduction modes deserves special attention.
The WDT continues to count while the TSC80251G1 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 TSC80251G1 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# or INT1#. 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.

8.5. Register

WDTRST (S:A6h)
Hardware Watchdog Timer Reset register (8–bit write–only)



7	6	5	4	3	2	1	0

Reset Value = 1111 1111b

Figure 8.1. WDTRST Register

Power Monitoring and Management

9.1. Introduction

The power monitoring and management can be used to supervise the Power Supply (VDD) and to start up properly when the TSC80251G1 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.

9.2. 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 9.1.), 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 is immediately enabled.

When the power supply rises above V_{RET} and as long as it stays below V_{RST+}, the Power–On Flag (POF, see paragraph 9.4.) and the Reset Detection control bit (RSTD, see Figure 9.6.) 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 occured. 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 9.1. Pin Conditions in Special Operating Modes

Mode	Program Memory	ALE pin	PSEN# pin	Port 0 pin	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

9.3. Power–Fail Reset

The Power–Fail detector is enabled by RSTD bit in POWM register. Then the power supply is continuously monitored and an internal reset is generated if VDD goes below V_{RST-} for at least 2xPFILTxTOSC (Note: this internal reset is not propagated outside). The Power Filter register (PFILT, see Figure 9.5.) must be programmed by the user with an 8–bit value depending of the time constant he wants.

If the power supply rises again over V_{RST+}, the Power–On Flag (POF, see paragraph 9.4.) is set and the internal reset completes after 64 oscillator clock periods.

If RSTD is cleared, the power supply monitoring is disabled. This is particularly usefull to save power in Power–Down mode. Then VDD may be reduced to V_{RET} without generating a reset or losing data. In this case, the circuit behavior is unpredictable unless an external reset is applied when VDD goes below V_{RET}.

9.4. Power–Off Flag

When the power is turned off or fails, the data retention is not guaranteed. A Power–Off Flag (POF, see Figure 9.4.) 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.

9.5. 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 9.1. 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:

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

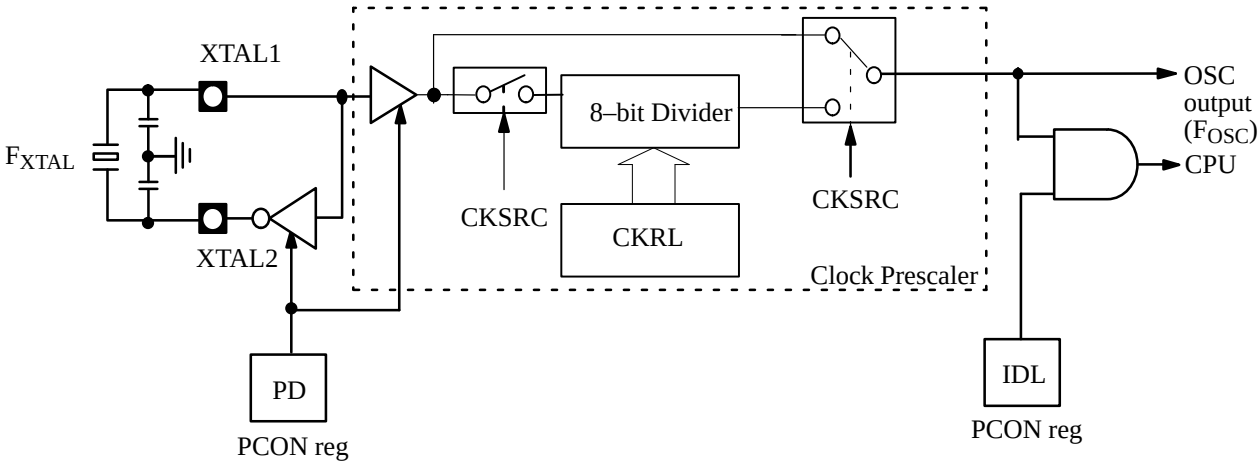


Figure 9.1. Block Diagram of the On–Chip Oscillator

In all this document, the on–chip oscillator is used to be symbolized by Figure 8.7. Please notice that all the peripherals share the same clock. Special care should be taken when changing it to prevent any peripheral operating failure.

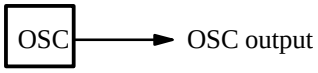


Figure 9.2. Symbolic of the On–Chip Oscillator

9.6. 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 9.1.). 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. 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 9.1.).
- 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 9.1.).

9.6.1. Entering Idle Mode

To enter Idle mode, set IDL bit in PCON register. The TSC80251G1 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 TSC80251G1 enters Power–Down mode. Then it does not go in Idle mode when exiting Power–Down mode.

9.6.2. 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 TSC80251G1 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.

9.6.3. 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 9.7.3.). 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.

Notes:
- The external interrupt used to recover from Idle mode must be configured as level sensitive 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.
- When RPD bit in PCON register is set, it is still possible to exit Idle mode using an enabled interrupt (See paragraph 9.6.2.).

9.7. Power–Down Mode

The Power–Down mode places the TSC80251G1 in a very low power state. Power–Down mode stops the oscillator and freezes all clock at known states (See Figure 9.1.). 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 9.1.).
- 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 9.1.).

Note:
VDD may be reduced to as low as 2 V during Power–Down mode to further reduce power dissipation. Take care, however, that VDD is not reduced until Power–Down mode is invoked.

9.7.1. Entering Power–Down Mode

To enter Power–Down mode, set PD bit in PCON register. The TSC80251G1 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.

9.7.2. 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.
  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.

Note:
To enable an external interrupt, set EX0 and/or EX1 bit(s) in IE register. The external interrupt used to exit Power–Down mode must be configured as level sensitive and must be assigned the highest priority. In addition, the duration of the interrupt must be of sufficient length to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted.

- 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 TSC80251G1 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 the Power–Down mode should not write to a Port pin or to the external RAM.

9.7.3. 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 9.6.3.). 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 recover from Power–Down mode must be configured as level sensitive 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.
- When RPD bit in PCON register is set, it is still possible to exit Power–Down mode using an enabled external interrupt (See paragraph 9.7.2.).

9.8. Registers

CKRL (S:8Eh)
Clock Reload register

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

Reset Value = 00h

Figure 9.3. CKRL Register

PCON (S:87h)
Power Configuration register

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

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 3 V to indicate that the Power Supply has been off or VDD had fallen below 3 V and that on–chip volatile memory is indeterminated. 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

Figure 9.4. PCON Register

PFILT (S:86h)
Power Filter register

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

Reset Value = XXh

Figure 9.5. PFILT Register

POWM (S:8Fh)
Power Management register

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

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: $F_{OSC} = \frac{F_{XTAL}}{2(CKRL + 1)}$ The CPU frequency is $F_{OSC}/2$.
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 0XXX

Figure 9.6. POWM Register

Interrupt System

10.1. Introduction

The TSC80251G1, 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 TSC80251G1 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 TSC80251G1, 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 10.1. 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**	–

Table 10.2. Interrupt System SFRs

Mnemonic	Description	Address
IE0	**Interrupt Enable register** Used to enable and disable the nine lowest programmable interrupts. The reset value of this register is zero (interrupts disabled).	S:A8h
IE1	**Interrupt Enable register** Used to enable and disable the nine highest programmable interrupts. The reset value of this register is zero (interrupts disabled).	S:B1h
IPH0	**Interrupt Priority High register 0** Establishes relative four–level priority for the nine lowest programmable interrupts. Used in conjunction with IPL0.	S:B7h
IPH1	**Interrupt Priority High register 1** Establishes relative four–level priority for the nine highest programmable interrupts. Used in conjunction with IPL1.	S:B3h
IPL0	**Interrupt Priority Low register 0** Establishes relative four–level priority for the nine lowest programmable interrupts. Used in conjunction with IPH0.	S:B8h
IPL1	**Interrupt Priority Low register 1** Establishes relative four–level priority for the nine lowest programmable interrupts. Used in conjunction with IPH1.	S:B2h

The TSC80251G1 has one software interrupt: TRAP, NMI and nine peripheral 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 10.3. and Figure 10.4.).
Four 8–bit registers are used to establish the priority level of the nine sources: IPL0, IPH0, IPL1 and IPH1 (See Figure 10.5. to Figure 10.8.).

10.2. Interrupt System Priorities

Each of the nine interrupt sources on the TSC80251G1 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 10.6. and Figure 10.8.) and one in the Interrupt Priority Low registers (IPL0 or IPL1, see Figure 10.5. and Figure 10.7.) This provides each interrupt source four possible priority levels select bits (See Table 10.3.).

Table 10.3. Level of Priority

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 10.4. *).*

Table 10.4. 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	H
Keyboard	10	FF:0043h	S
Reserved	–	FF:004Bh	–
Reserved	–	FF:0053h	–
Reserved	–	FF:005Bh	–
Reserved	–	FF:0063h	S
SSLC	15	FF:006Bh	
Reserved	16 Lowest Priority	FF:0073h	–

10.3. External Interrupts

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 bit **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 bit. External hardware must deassert INTn# before the service routine completes, or an additional interrupt is requested. External interrupt pins must be deasserted for at least four state times prior to a request.

External interrupt pins are sampled once every four state times (a frame length of 500 ns at 16 MHz). 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.

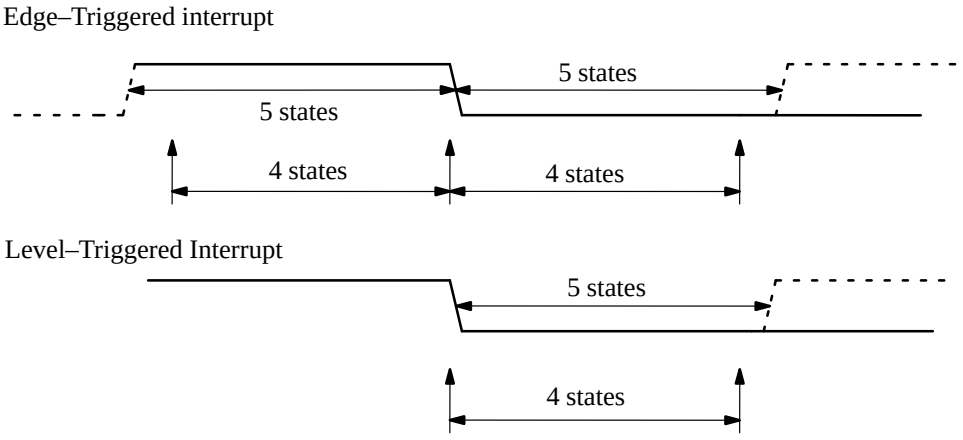


Figure 10.1. Minimum Pulse Timings.

10.4. Keyboard Interface

Port 1 as 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. Port lines are sampled once every state, then a level must 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 10.11.).
- The detection of the programmed level sets the corresponding flag in P1F register (See Figure 10.9.).

- If the corresponding Port 1 Interrupt Enable bit in P1IE register (See Figure 10.10.) is set, the flag setting generates the Keyboard interrupt request.
  But it must be enabled by the KBIE bit in IE1 register (See Figure 10.4.).
- 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.

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.

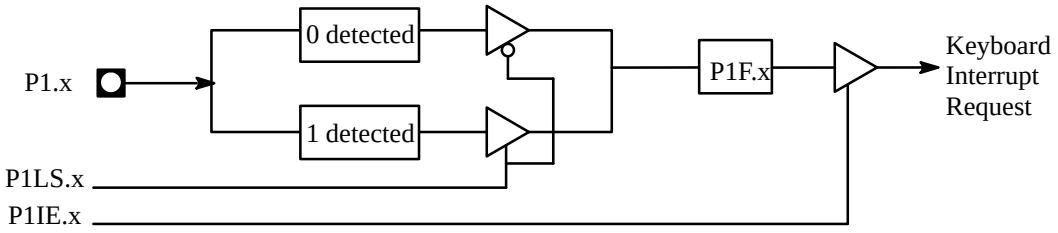


Figure 10.2. Keyboard Interface Interrupt Structure

10.5. Registers

IE0 (S:A8h)
Interrupt Enable 0 register

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

Bit Number	Bit Mnemonic	Description
7	EA	**Global Interrupt Enable bit** Clear to disable all interrupts that are individually disabled by bits 6:0 in IE0 register and bits 5 and 0 in IE1 register, except the TRAP register which is always enabled. Set to enable all interrupts that are individually enabled by bits 6:0 in IE0 register and bits 5 and 0 in IE1 register.
6	EC	**Enable Counter Interrupt bit** Clear to disable EWC interrupt. Set to enable EWC interrupt.
5	ET2	**Enable Timer 2 Interrupt bit** Clear to disable Timer 2 overflow interrupt. Set to enable Timer 2 overflow interrupt.
4	ES	**Enable Serial Port Interrupt bit** Clear to disable Serial Port interrupt. Set to enable Serial Port interrupt.

Bit Number	Bit Mnemonic	Description
3	ET1	**Enable Timer 1 Interrupt bit** Clear to disable Timer 1 overflow interrupt. Set to enable Timer 1 overflow interrupt.
2	EX1	**Enable External 1 Interrupt bit** Clear to disable external interrupt 1. Set to enable external interrupt 1.
1	ET0	**Enable Timer 0 Interrupt bit** Clear to disable Timer 0 overflow interrupt. Set to enable Timer 0 overflow interrupt.
0	EX0	**Enable External 0 Interrupt bit** Clear to disable External interrupt 0. Set to enable External interrupt 0.

Reset Value = 0000 0000b

Figure 10.3. IE0 Register

IE1 (S:B1h)
Interrupt Enable 1 register

–	–	SSIE	–	–	–	–	KBIE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

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 10.4. IE1 Register

IPH0 (S:B7h)
Interrupt Priority High 0 register

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

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit
6	IPHC	**EWC 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	**Timer 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 0000b

Figure 10.5. IPH0 Register

IPH1 (S:B1h)
Interrupt Priority High 1 register

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

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** IPHSSLC IPLSSLC 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

Figure 10.6. IPH1 Register

IPL0 (S:B8h)
Interrupt Priority Low 0 register

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

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	IPLC	**EWC 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 IPHADC 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 10.7. IPL0 Register

IPL1 (S:B2h)
Interrupt Priority Low 1 register

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

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 IPHSSLC 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 IPHPMU for priority level.

Reset Value = XX0X XXX0b

Figure 10.8. IPL1 Register

P1F (S:9Eh)
Port 1 Flag register

P1F.7	P1F.6	P1F.5	P1F.4	P1F.3	P1F.2	P1F.1	P1F.0
7	**6**	**5**	**4**	**3**	**2**	**1**	**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 10.9. P1F Register

P1IE (S:9Dh)
Port 1 Interrupt Enable register

P1IE.7	P1IE.6	P1IE.5	P1IE.4	P1IE.3	P1IE.2	P1IE.1	P1IE.0
7	**6**	**5**	**4**	**3**	**2**	**1**	**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 10.10. P1IE Register

P1LS (S:C9h)
Port 1 Level Selection register

7	**6**	**5**	**4**	**3**	**2**	**1**	**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 10.11. PILS Register

Section III

Electrical and Mechanical Information

DC Characteristics

Table 1.1. Absolute Maximum Ratings

● Ambient Temperature Under Bias	
Commercial	0 to +70°C
Industrial	–40 to +85°C
Automotive	0 to +125°C
● Storage Temperature	–65 to +150°C
● Voltage on EA#/VPP Pin to VSS	0 to +13.0 V
● Voltage on any other Pin to VSS	–0.5 to +6.5 V
● I_{OL} per I/O Pin	15 mA
● Power Dissipation	1.5 W

Note:
Stresses at or above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

Table 1.2. DC Characteristics

Parameter values applied to all devices unless otherwise indicated.

Commercial	**Industrial**	**Automotive**
TA = 0 to 70°C VSS = 0 V VDD = 5 V ± 10 %	TA = –40 to +85°C VSS = 0 V VDD = 5 V ± 10 %	TA = –40 to +125°C VSS = 0 V VDD = 5 V ± 10 %

Symbol	Parameter	Min	Typical (4)	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (except EA#)	–0.5		0.2VDD - 0.1	V	
V_{IL1}	Input Low Voltage (EA#)	0		0.2VDD - 0.3	V	
V_{IH}	Input high Voltage (except XTAL1, RST)	0.2VDD + 0.9		VDD + 0.5	V	
V_{IH1}	Input high Voltage (XTAL1)	0.7 VDD		VDD + 0.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3)			0.3 0.45 1.0	V	I_{OL} = 100 µA I_{OL} = 1.6 mA I_{OL} = 3.5 mA (1, 2)
V_{RST}+	Reset threshold on		3.7		V	
V_{RST}–	Reset threshold off		3.3		V	

Symbol	Parameter	Min	Typical (4)	Max	Units	Test Conditions
V_{RET}	VDD data retention limit		2		V	
V_{OL1}	Output Low Voltage (Ports 0, ALE, PSEN#)			0.3 0.45 1.0	V	I_{OL} = 200 µA I_{OL} = 3.2 mA I_{OL} = 7.0 mA (1, 2)
V_{OH}	Output high Voltage (Ports 1, 2, 3, ALE, PSEN#)	VDD –0.3 VDD –0.7 VDD –1.5			V	I_{OH} = –10 µA I_{OH} = –30 µA I_{OH} = –60 µA (3)
V_{OH1}	Output high Voltage (Port 0 in External Address)	VDD –0.3 VDD –0.7 VDD –1.5			V	I_{OH} = –200 µA I_{OH} = –3.2 mA I_{OH} = –7.0 mA
V_{OH2}	Output high Voltage (Port 2 in External Address during Page Mode)	VDD –0.3 VDD –0.7 VDD –1.5			V	I_{OH} = –200 µA I_{OH} = –3.2 mA I_{OH} = –7.0 mA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			- 50 - 75	µA	V_{IN} = 0.45 V Automotive range
I_{LI}	Input Leakage Current (Port 0)			± 10	µA	0.45<V_{IN}<VDD
I_{TL}	Logical 1-to-0 Transition Current (Ports 1, 2, 3)			- 650	µA	VIN = 2.0 V
R_{RST}	RST Pull–Down Resistor	40		225	kΩ	
C_{IO}	Pin Capacitance		10		pF	F_{OSC} = 16 MHz T_A = 25°C
I_{PD}	Powerdown Current		20		µA	
I_{DL}	Idle Mode Current		15		mA	F_{OSC} = 16 MHz
			10		mA	F_{OSC} = 12 MHz

Symbol	Parameter	Min	Typical (4)	Max	Units	Test Conditions
I_{DD}	Operating Current		50		mA	F_{OSC} = 16 MHz
			40		mA	F_{OSC} = 12 MHz

Notes:
1. Under steady–state (non–transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: . . . . . 10 mA
Maximum I_{OL} per 8–bit port: Port 0 . . . . . 26 mA
Ports 1-3 . . . . 15 mA
Maximum Total I_{OL} for all: Output Pins . . 71 mA
If I_{OL} exceeds the test conditions, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses above 0.4 V on the low–level outputs of ALE and Ports 1, 2, and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins change from high to low. In applications where capacitive loading exceeds 100 pF, the noise pulses on these signals may exceed 0.8 V. It may be desirable to qualify ALE or other signals with a Schmitt Trigger or CMOS–level input logic.
3. Capacitive loading on Ports 0 and 2 causes the V_{OH} on ALE and PSEN# to drop below the specification when the address lines are stabilizing.
4. Typical values are obtained using VDD = 5 V and T_A = 25°C with no guarantee.
They are not tested and there is not guarantee on these values.

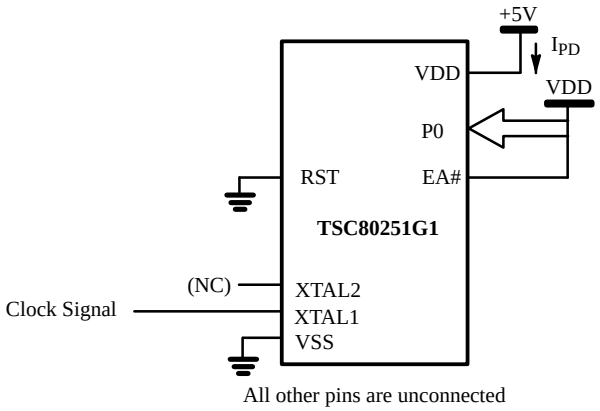


Figure 1.1. I_{PD} Test Condition, Power–Down Mode

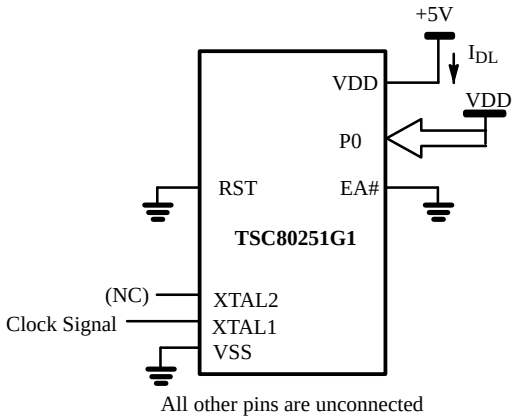


Figure 1.2. I_{DL} Test Condition, Idle Mode

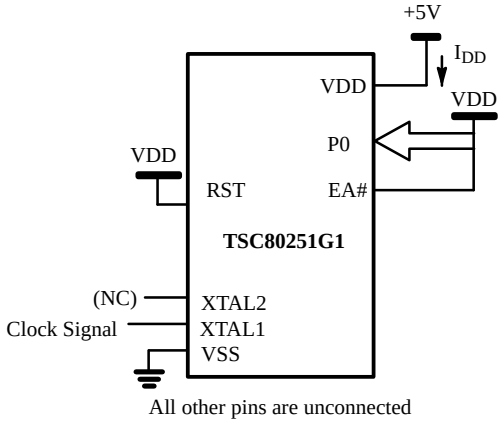


Figure 1.3. I_{DD} Test Condition, Active Mode

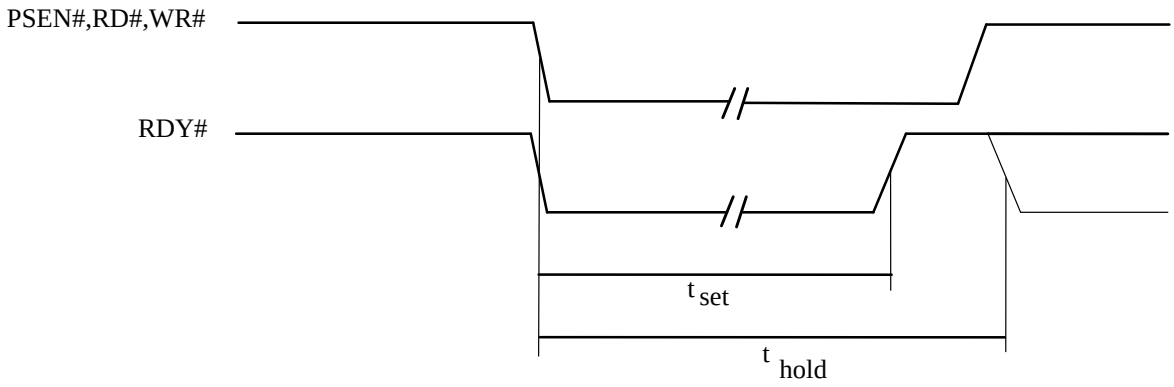


Figure 1.4. Wait Timings

Table 1.3. Electrical Parameters

Parameter	Definition	Min	Max	Units
t_{clcl}	Clock period	62,50		ns
t_{set}	Ready valid after strobe (RD#, WR# or PSEN#) low		$(2N+1).t_{clcl} - 15$	ns
t_{hold}	Ready hold after strobe low	$(2N+1).t_{clcl} + 2$		ns

AC Characteristics

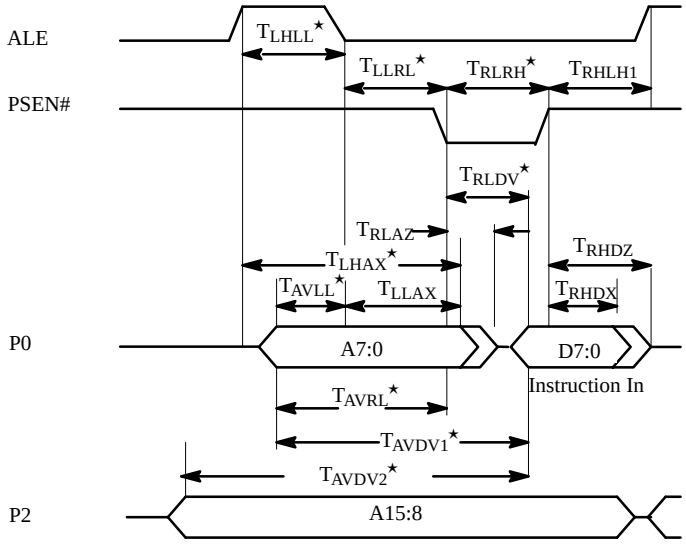
Table 2.1. AC Characteristics (Capacitive Loading = 50 pF)

Symbol	Parameter	12 MHz		16 MHz		F_{OSC}		Units
		Min	Max	Min	Max	Min	Max	
T_{OSC}	$1/F_{OSC}$	83		63				ns
T_{LHLL}	ALE Pulse Width	73		53		T_{OSC} -10		ns (2)
T_{AVLL}	Address Valid to ALE Low	63		43		T_{OSC} - 20		ns (2)
T_{LLAX}	Address hold after ALE Low	63		43		T_{OSC} - 20		ns
T_{RLRH} (1)	RD# or PSEN# Pulse Width	65		45		T_{OSC} - 18		ns (3)
T_{WLWH}	WR# Pulse Width	65		45		T_{OSC} - 18		ns (3)
T_{LLRL} (1)	ALE Low to RD# or PSEN# Low	73		53		T_{OSC} - 10		ns
T_{RHRL}	ALE High to RD# or PSEN# High	73		53		T_{OSC} - 10		ns
T_{LHAX}	ALE high to Address hold	147		105		$2T_{OSC}$ - 20		ns (2)
T_{RLDV} (1)	RD# or PSEN# Low to Valid Data/Instruction.		33		13	T_{OSC} - 50		ns (3)
T_{RHDX} (1)	Data/Instruct. hold After RD# or PSEN# high	0		0		0		ns
T_{RLAZ} (1)	RD#/PSEN# Low to Address Float		2		2		2	ns
T_{RHDZ} (1)	Data/Instruct. Float After RD# or PSEN# high		63		43		T_{OSC} - 20	ns
T_{RHLH1} (1)	RD#/PSEN# high to ALE high (Instruction)	68		48		T_{OSC} - 15		ns (1)
T_{RHLH2} (1)	RD#/PSEN# high to ALE high (Data)	235		173		$3T_{OSC}$ - 15		ns (1)
T_{WHLH}	WR# high to ALE high	235		173		$3T_{OSC}$ - 15		ns
T_{AVDV1}	Address (P0) Valid to Valid Data/Instruction In		190		128		$3T_{OSC}$ - 60	ns (2, 3, 4)
T_{AVDV2}	Address (P2) Valid to Valid Data/Instruction In		273		190		$4T_{OSC}$ - 60	ns (2, 3, 4,)
T_{AVDV3}	Address (P0) Valid to Valid Instruction In		128		88		$2T_{OSC}$ - 38	ns

3

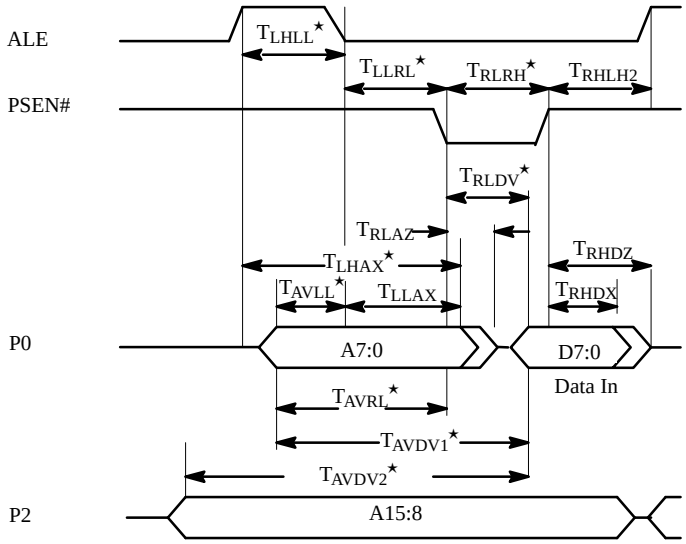
Symbol	Parameter	12 MHz		16 MHz		F_{OSC}		Units
		Min	Max	Min	Max	Min	Max	
T_{AVRL}	Address Valid to RD#/PSEN# Low	143		101		$2T_{OSC}$ - 24		ns (2)
T_{AVWL1}	Address (P0) Valid to WR# Low	143		101		$2T_{OSC}$ - 24		ns (2)
T_{AVWL2}	Address (P2) Valid to WR# Low	220		158		$3T_{OSC}$ - 30		ns (2)
T_{WHQX}	Data hold after WR# high	63		43		T_{OSC} - 20		ns
T_{QVWH}	Data Valid to WR# high	58		38		T_{OSC} - 25		ns (3)
T_{WHAX}	WR# high to Address hold	147		105		$2T_{OSC}$ - 20		ns
T_{XLXL}	Serial Port Clock Cycle Time	1000		750		12 T_{OSC}		ns
T_{QVSH}	Output Data Setup to Clock Rising Edge	870		620		12 T_{OSC} - 133		ns
T_{XHQX}	Output Data hold after Clock Rising Edge	720		510		10 T_{OSC} - 117		ns
T_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		0		ns
T_{XHDV}	Clock Rising Edge to Input Data Valid		700		500		10 T_{OSC} - 133	ns

Notes :
1. Specifications for PSEN# are identical to those for RD#.
2. If a wait state is added by extending ALE, add $2T_{OSC.}$
3. If a wait state is added by extending RD#/PSEN#/WR#, add $2T_{OSC}$.
4. If wait states are added as described in both Note 2 and Note 3, add a total of $4T_{OSC}$.



★ The value of this parameter depends on wait states. See the table of AC characteristics.

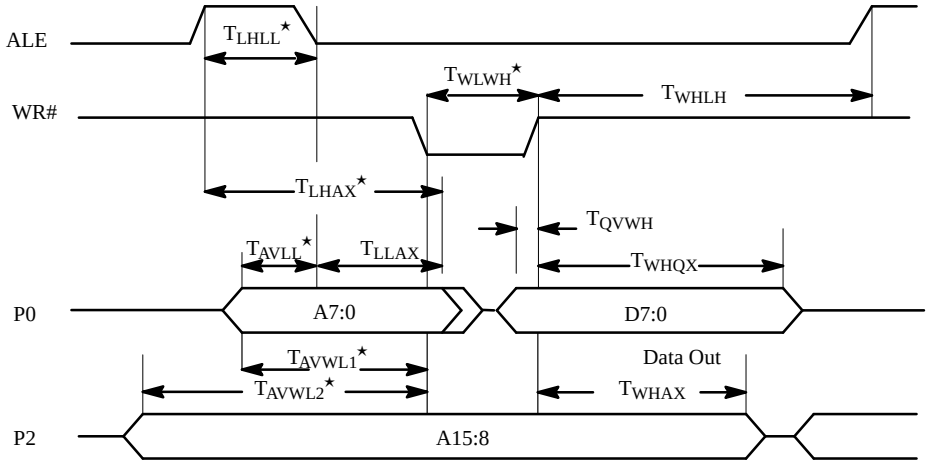
Figure 2.1. External Instruction Bus Cycle in Non–Page Mode



★ The value of this parameter depends on wait states. See the table of AC characteristics.

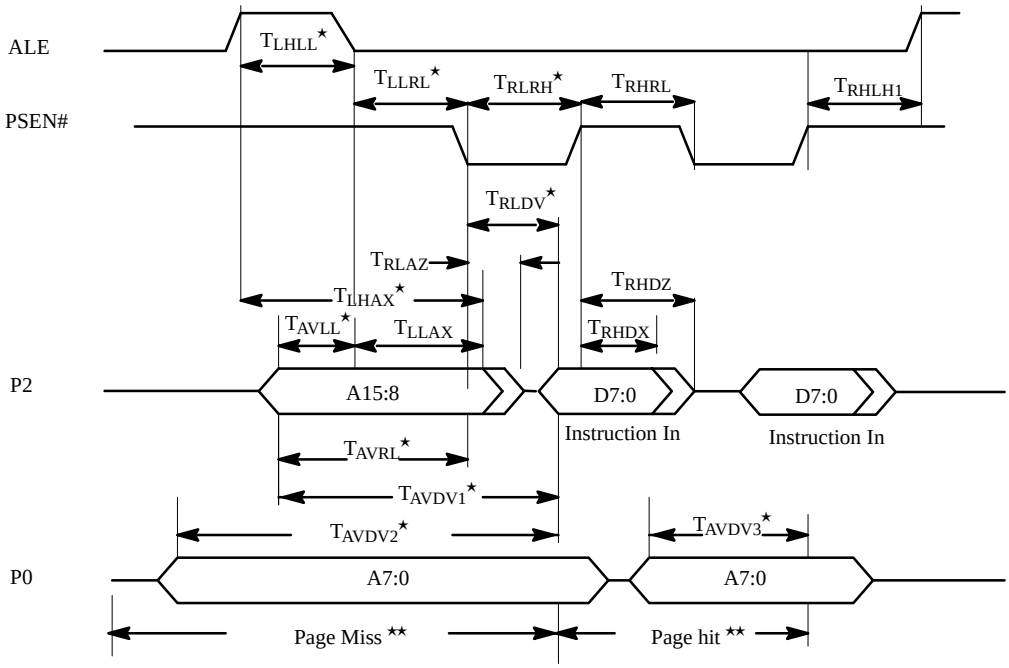
Figure 2.2. External Data Read Cycle in Non–Page Mode

3



★ The value of this parameter depends on wait states. See the table of AC characteristics.

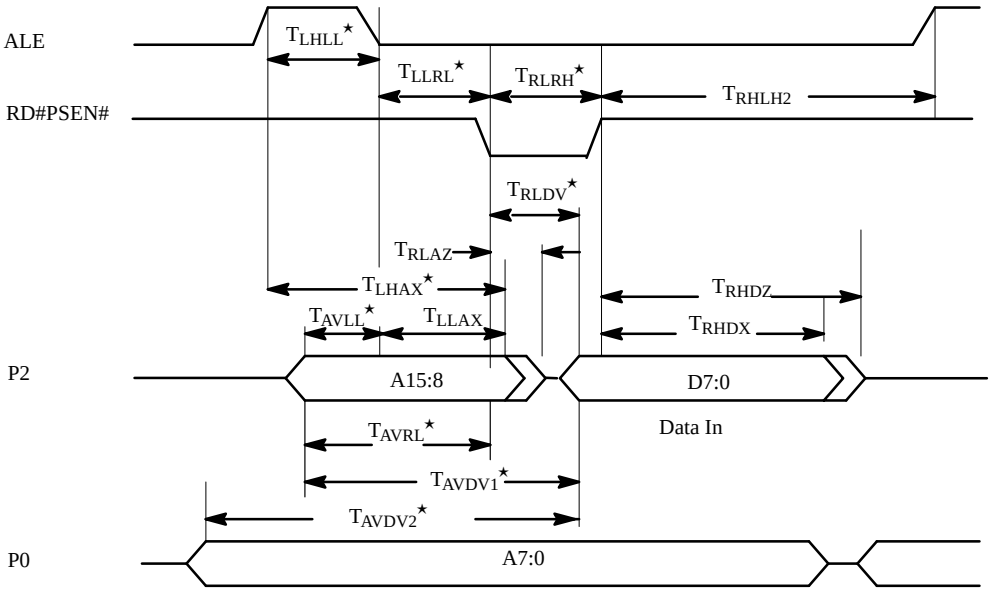
Figure 2.3. External Write Data Bus Cycle in Non–Page Mode



★ The value of this parameter depends on wait states. See the table of AC characteristics.

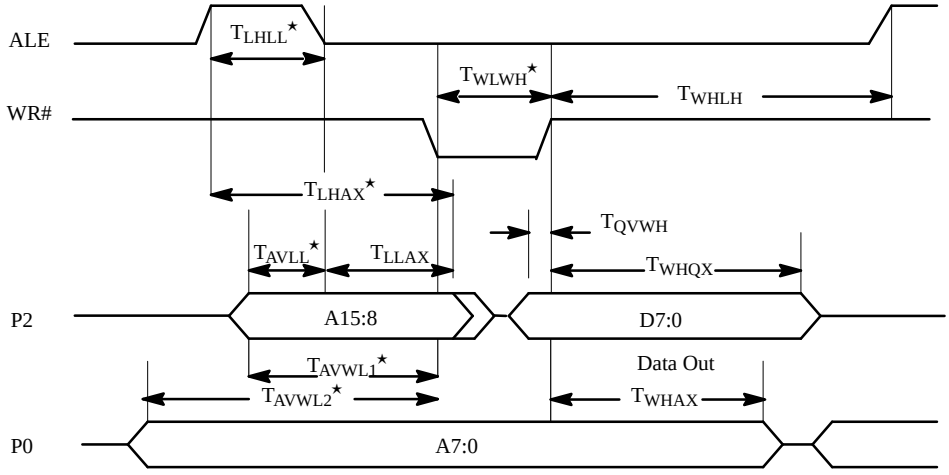
★★ A page hit (i.e., a code fetch to the same 256-byte "page" as the previous code fetch) requires one state ($2T_{OSC}$); a page miss requires two states ($4T_{OSC}$).

Figure 2.4. External Instruction Bus Cycle in Page Mode



★ The value of this parameter depends on wait states. See the table of AC characteristics.

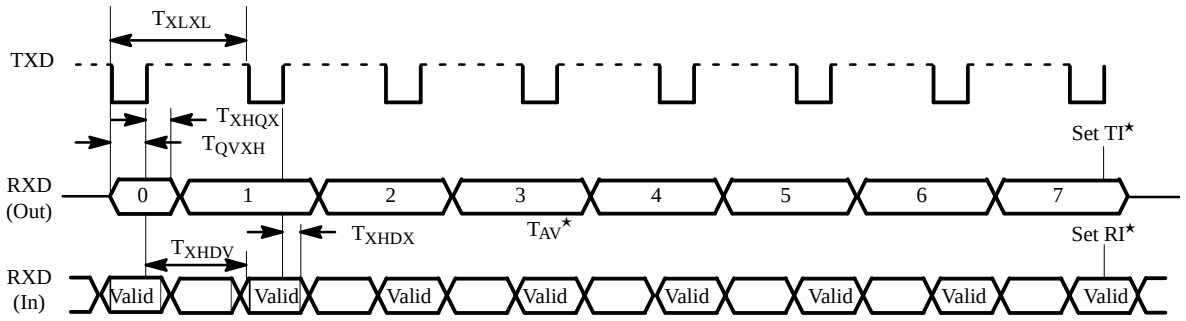
Figure 2.5. External Read Data Bus Cycle in Page Mode



★ The value of this parameter depends on wait states. See the table of AC characteristics.

Figure 2.6. External Write Data Bus Cycle in Page Mode

3



* TI and RI are set during S1P1 of the peripheral cycle following the shift of the eight bit.

Figure 2.7. Serial Port Waveform – Shift Register Mode

Notation for timing parameters name

A = Address	D = Data	E = Enable	G = PROG#	H = high	L = Low
Q = Data out	S = Supply (VPP)	V = Valid	X = No Longer Valid		Z = Floating

IC Interface AC/DC Characteristics

3.1. AC Characteristics

3.1.1. SSLC (IC) Interface Timing

Symbol	Parameter	Input	Output
$T_{HD;STA}$	Start condition hold time	14 T_{CLCL}	> 4.0s (1)
T_{LOW}	SCL low time	16 T_{CLCL}	> 4.7s (1)
T_{HIGH}	SCL high time	14 T_{CLCL}	> 4.0s (1)
T_{RC}	SCL rise time	1s	– (2)
T_{FC}	SCL fall time	0.3s	< 0.3s (3)
$T_{SU;DAT1}$	Data set–up time	250ns	> 20 T_{CLCL} – T_{RD}
$T_{SU;DAT2}$	SDA set–up time (before repeated START condition)	250ns	> 1s (1)
$T_{SU;DAT3}$	SDA set–up time (before STOP condition)	250ns	> 8 T_{CLCL}
$T_{HD;DAT}$	Data hold time	0ns	> 8 T_{CLCL} – T_{FC}
$T_{SU;STA}$	Repeated START set–up time	14 T_{CLCL}	> 4.7s (1)
$T_{SU;STO}$	STOP condition set–up time	14 T_{CLCL}	> 4.0s (1)
T_{BUF}	Bus free time	14 T_{CLCL}	> 4.7s (1)
T_{RD}	SDA rise time	1s	– (2)
T_{FD}	SDA fall time	0.3s	< 0.3s (3)

1. At 100 kbit/s. At other bit–rates this value is inversely proportional to the bit–rate of 100 kbit/s.
2. Determined by the external bus–line capacitance and the external bus–line pull–up resistor, this must be < 1s.
3. Spikes on the SDA and SCL lines with a duration of less than 3 T_{CLCL} will be filtered out. Maximum capacitance on bus–lines SDA and SCL = 400pF.
4. T_{CLCL} = 1/fosc = one oscillator clock period at pin XTAL1. For 83ns < T_{CLCL} < 285ns (12MHz > fosc > 3.5MHz) the SSLC interface meets the IC bus specification for bit–rate up to 100 kbit/s

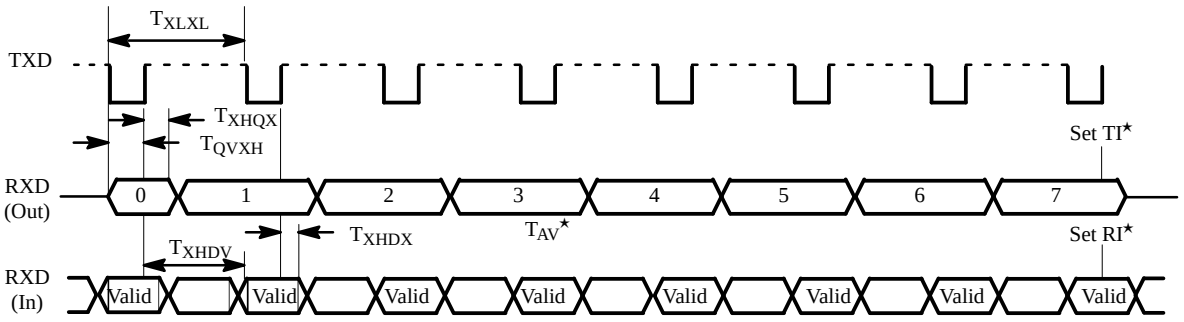
3.2. DC Characteristics

Symbol	Parameter	Min	Max	Unit	Test Conditions
Inputs					
VIL1	Input Low Voltage, SCL, SDA(5)	–0.5	0.3 Vcc	V	
VIH2	Input High Voltage, SCL, SDA(5)	0.7 Vcc	Vcc + 0.5	V	
Outputs					
VOL	Output Low Voltage, ports 1, 2, 3, SCL, SDA, PWM0–7		0.3 0.45 1.0	V V V	IOL = 100µA IOL = 1.6mA(4) IOL = 3.5mA

Note:

5. The input threshold voltage of SCL and SDA (SIO1) meets the IC specification, so an input voltage below 0.3.VCC will be recognised as a logic 0 while an input voltage above 0.7.VCC will be recognised as a logic 1.

3.2.1. SSLC (IC) Timing Waveforms



* TI and RI are set during S1P1 of the peripheral cycle following the shift of the eight bit.

Figure 3.1.

EPROM Programming

4.1. Programming Modes

The TSC87251G1 derivatives in Window CQPJ are erasable by UV which set all the EPROM memory cells to one and allows a reprogrammation. The other TSC87251G1 derivatives are one time programmable as an EPROM cell cannot be reset once programmed to 0. Table 4.1. shows the hardware setup needed to program the TSC87251G1 EPROM areas:

- The chip has to be maintained under reset and the PSEN# has to be to forced to 0 until the completion of the Programming sequence.
- The Programming address are applied on Ports 1 and 3 which are respectively the upper and lower address lines.
- The Programming data are applied on Port 2.
- The EPROM Programming is done by applying VPP on the EA# pin and by generating 5 pulses on ALE/PROG# pin for the on–chip code memory and 25 for the Configuration bytes.

Table 4.1. EPROM Programming Configuration

EPROM Mode	RST	EA#	PSEN#	ALE	P0	P2	P1(Upper)P3(Lower)	Notes
On–chip code memory	1	VPP	0	5 Pulses	68h	Data	0000h-5FFFh	1
Configuration bytes	1	VPP	0	25 Pulses	69h	Data	0080h-0081h	1

Notes:
1. The ALE/PROG# pulse waveform is shown in Figure 4.2.

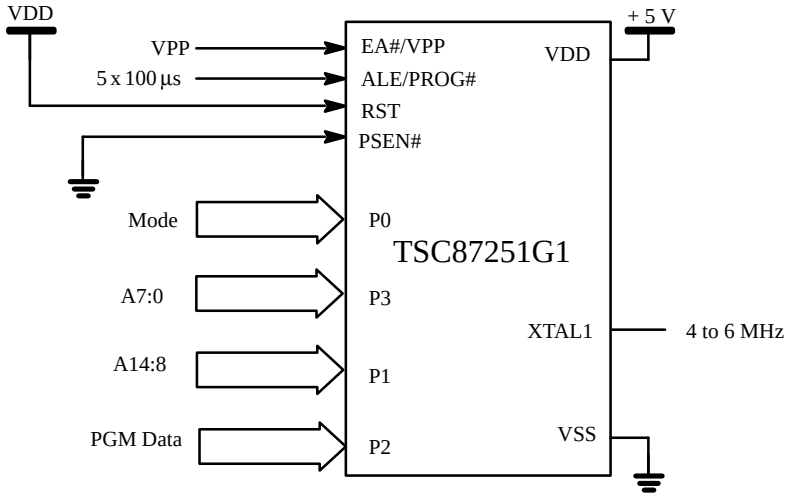
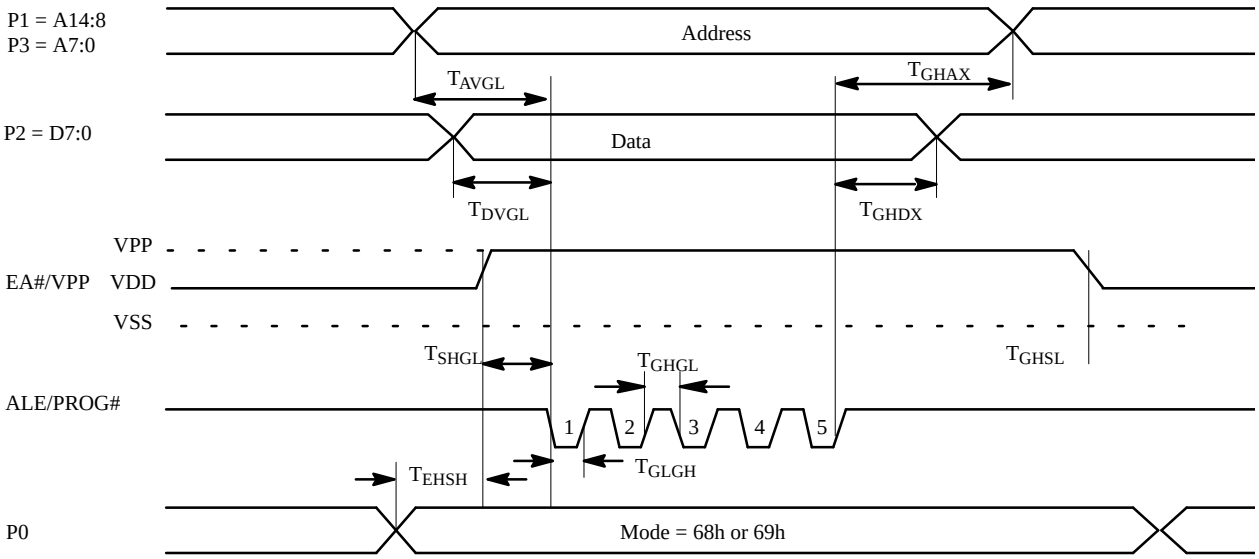


Figure 4.1. Setup for EPROM Programming



Note:
The timing is the same for both Programming Modes excepted the number of Programming pulses. Only 5 Programming pulses are shown here.

Figure 4.2. Timings for EPROM Programming

4.2. Verify Algorithm

Figure 4.3. show the setup needed to verify the TSC80251G1 EPROM areas. Table 4.2. shows the Configuration needed to verify the on-chip code memory and Configuration bytes. The 15 addresses must be connected to the Ports 3 and 1. ALE/PROG# and PSEN# are driven low while Port 0 receives the Configuration.

Figure 4.4. shows the timings to apply in orded to execute the EPROM verify Mode.

- Port 0 drives the verify Mode (28h for Programming Mode).
- The address to access is driven on Port 1 and Port 3 while the PSEN# and ALE are driven low. The data is driven on Port 2, 48 clock periods after the address is stable.

Table 4.2. EPROM Verifying Configuration

Verify EPROM	RST	EA#	PSEN#	ALE	P0	P2	P1(Upper) P3(Lower)
On–chip code memory	1	1	0	1	28h	Data	0000h-5FFFh
Configuration bytes	1	1	0	1	29h	Data	0080h-0083h

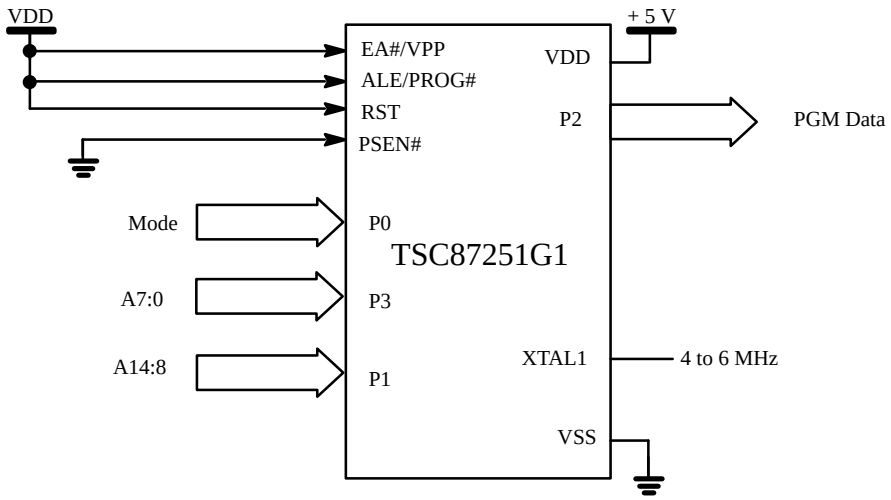


Figure 4.3. Setup for EPROM Verification

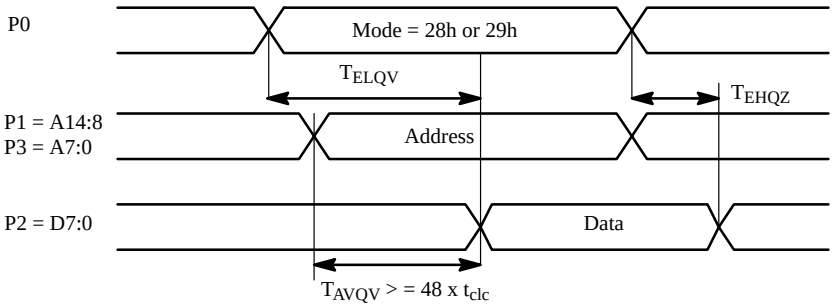


Figure 4.4. Timings for EPROM Verification

Table 4.3. EPROM Programming & Verification Characteristics
(TA = 21 to 27°C ; VCC = 5V +/– 0.25V ; VSS= 0)

Symbol	Parameter	Min	Max	Units
VPP	Programming Supply Voltage	12,75	13	V
IPP	Programming Supply Current		75	mA
T_{OSC}	Oscillator Frequency	167	250	ns
T_{AVGL}	Address Setup to PROG# low	$48T_{OSC}$		
T_{GHAX}	Address Hold after PROG# low	$48T_{OSC}$		
T_{DVGL}	Data Setup to PROG# low	$48T_{OSC}$		
T_{GHDX}	Data Hold after PROG#	$48T_{OSC}$		
T_{EHSH}	ENABLE High to VPP	$48T_{OSC}$		
T_{SHGL}	VPP Setup to PROG# low	10		µs
T_{GHSL}	VPP Hold after PROG#	10		µs
T_{GLGH}	PROG# Width	90	110	µs
T_{AVQV}	Address to Data Valid		$48T_{OSC}$	
T_{ELQV}	ENABLE low to Data Valid		$48T_{OSC}$	
T_{EHQZ}	Data Float after ENABLE	0	$48T_{OSC}$	
T_{GHGL}	PROG high to PROG# low	10		µs

Packages

5.1. List of Packages

All available packages are described in this chapter.

- PDIL 40
- PLCC 44
- CQPJ 44 with Window
- PQFP 44
- VQFP 44 (10×10)
- VQFP 44 (14×14)

5.2. PDIL 40

5.2.1. Mechanical Outline

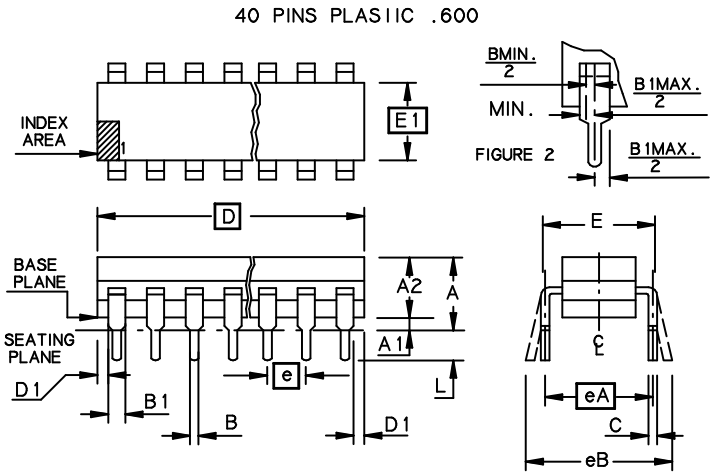


Figure 5.1. Plastic Dual In Line

Table 5.1. PDIL Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	–	5.08	–	.200
A1	0.38	–	.015	–
A2	3.18	4.95	.125	.195
B	0.36	0.56	.014	.022
B1	0.76	1.78	.030	.070
C	0.20	0.38	.008	.015
D	50.29	53.21	1.980	2.095
E	15.24	15.87	.600	.625
E1	12.32	14.73	.485	.580
e	2.54 B.S.C.		.100 B.S.C.	
eA	15.24 B.S.C.		.600 B.S.C.	
eB	–	17.78	–	.700
L	2.93	3.81	.115	.150
D1	0.13	–	.005	–

5.2.2. Pin Assignment

Table 5.2. PDIL Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.0/T2	21	P2.0/A8
2	P1.1/T2EX	22	P2.1/A9
3	P1.2/ECI	23	P2.2/A10
4	P1.3/CEX0	24	P2.3/A11
5	P1.4/CEX1	25	P2.4/A12
6	P1.5/CEX2/MISO	26	P2.5/A13
7	P1.6/CEX3/SCL/SCK	27	P2.6/A14
8	P1.7/A17/CEX4/SDA/MOSI	28	P2.7/A15
9	RST	29	PSEN#
10	P3.0/RXD	30	ALE/PROG#
11	P3.1/TXD	31	EA#VPP
12	P3.2/INT0#	32	P0.7/AD7
13	P3.3/INT13	33	P0.6/AD6
14	P3.4/T0	34	P0.5/AD5
15	P3.5/T1	35	P0.4/AD4
16	P3.6/WR#	36	P0.3/AD3
17	P3.7/A16/RD#	37	P0.2/AD2
18	XTAT2	38	P0.1/AD1
19	XTAL1	39	P0.0/AD0
20	VSS	40	VDD

5.3. PLCC 44

5.3.1. Mechanical Outline

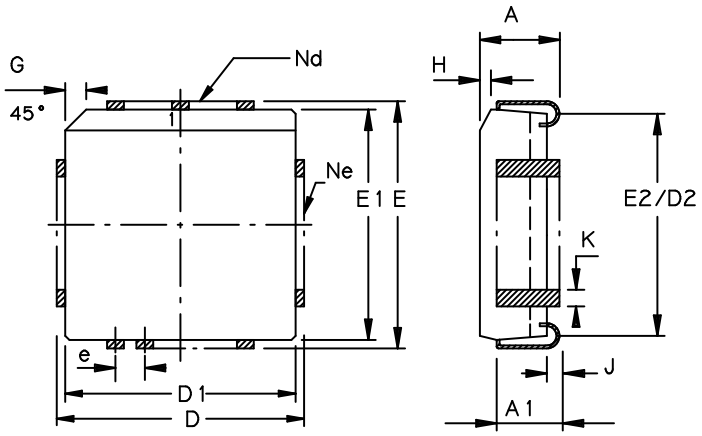


Figure 5.2. Plastic Lead Chip Carrier

Table 5.3. PLCC Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	4.20	4.57	.165	.180
A1	2.29	3.04	.090	.120
D	17.40	17.65	.685	.695
D1	16.44	16.66	.647	.656
D2	14.99	16.00	.590	.630
E	17.40	17.65	.685	.695
E1	16.44	16.66	.647	.656
E2	14.99	16.00	.590	.630
e	1.27 BSC		.050 BSC	
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	–	.020	–
K	0.33	0.53	.013	.021
Nd	11		11	
Ne	11		11	

5.3.2. Pin Assignment

Table 5.4. PLCC Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	VSS1	23	VSS2
2	P1.0/T2	24	P2.0/A8
3	P1.1/T3EX	25	P2.1/A9
4	P1.2/ECI	26	P2.2/A10
5	P1.3/CEX0	27	P2.3/A11
6	P1.4/CEX1	28	P2.4/A12
7	P1.5/CEX2/MISO	29	P2.5/A13
8	P1.6/CEX3/SCL/SCK	30	P2.6/A14
9	P1.7/A17/CEX4/SDA/MOSI	31	P2.7/A15
10	RST	32	PSEN#
11	P3.0/RXD	33	ALE/PROG#
12	WAIT#	34	NMI
13	P3.1/TXD	35	EA#/VPP
14	P3.2/INT0#	36	P0.7/AD7
15	P3.3/INT1#	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR#	40	P0.3/AD3
19	P3.7/A16/RD#	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VDD

5.4. CQPJ 44 with Window

5.4.1. Mechanical Outline

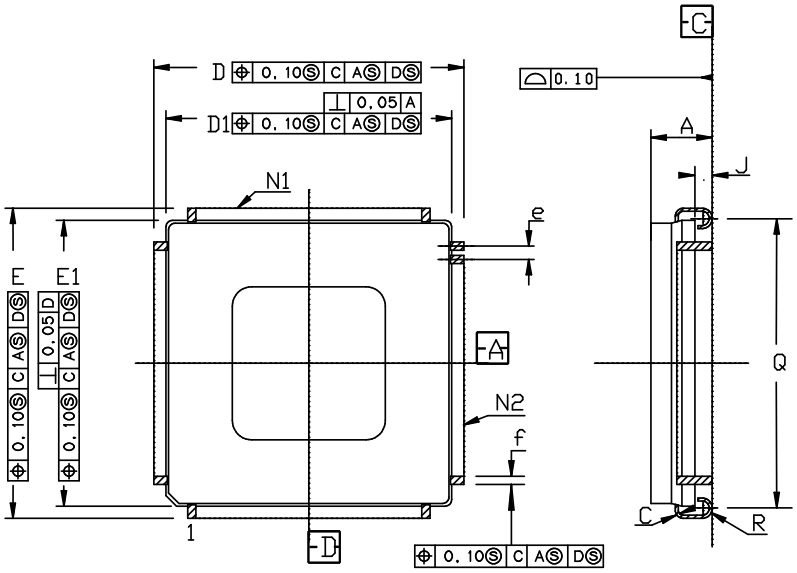


Figure 5.3. Ceramic Quad Pack J

Table 5.5. CQPJ Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	–	4.90	–	.193
C	0.15	0.25	.006	.010
D – E	17.40	17.55	.685	.691
D1 – E1	16.36	16.66	.644	.656
e	1.27 TYP		.050 TYP	
f	0.43	0.53	.017	.021
J	0.86	1.12	.034	.044
Q	15.49	16.00	.610	.630
R	0.86 TYP		.034 TYP	
N1	11		11	
N2	11		11	

5.4.2. Pin Assignment

Table 5.6. CQPJ Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.5/CEX2/MISO	23	P2.5/A13
2	P1.6/CEX3/SCL/SCK	24	P2.6/A14
3	P1.7/A17/CEX4/SDA/MOSI	25	P2.7/A15
4	RST	26	PSEN#
5	P3.0/RXD	27	ALE/PROG#
6	WAIT#	28	NMI
7	P3.1/TXD	29	EA#/VPP
8	P3.2/INT0#	30	P0.7/AD7
9	P3.3/INT1#	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR#	34	P0.3/AD3
13	P3.7/A16/RD#	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	VSS2	39	VSS1
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T3EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

5.5. PQFP 44

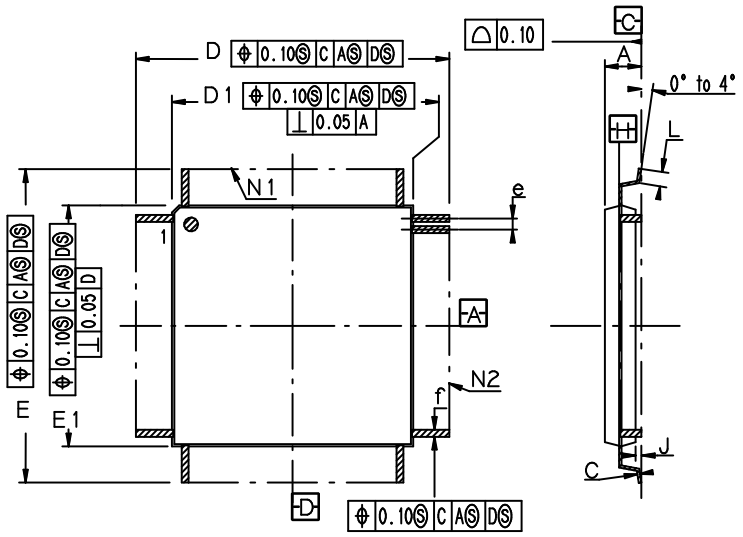


Figure 5.4. Plastic Quad Flat Pack

Table 5.7. PQFP Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	2.00	2.40	.079	.094
C	0.10	0.20	.004	.008
D	13.65	14.15	.537	.557
D1	9.90	10.10	.390	.398
E	13.65	14.15	.537	.557
E1	9.90	10.10	.390	.398
e	0.80 B.S.C.		.0315 B.S.C.	
f	0.20	0.40	.008	.016
J	0.00	0.30	.000	.012
L	0.65	0.95	.025	.037
N1	11		11	
N2	11		11	

5.5.1. Pin Assignment

Table 5.8. PQFP Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.5/CEX2/MISO	23	P2.5/A13
2	P1.6/CEX3/SCL/SCK	24	P2.6/A14
3	P1.7/A17/CEX4/SDA/MOSI	25	P2.7/A15
4	RST	26	PSEN#
5	P3.0/RXD	27	ALE/PROG#
6	WAIT#	28	NMI
7	P3.1/TXD	29	EA#/VPP
8	P3.2/INT0#	30	P0.7/AD7
9	P3.3/INT1#	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR#	34	P0.3/AD3
13	P3.7/A16/RD#	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	VSS2	39	VSS1
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T3EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

5.6. VQFP 44 (10×10)

5.6.1. Mechanical Outline

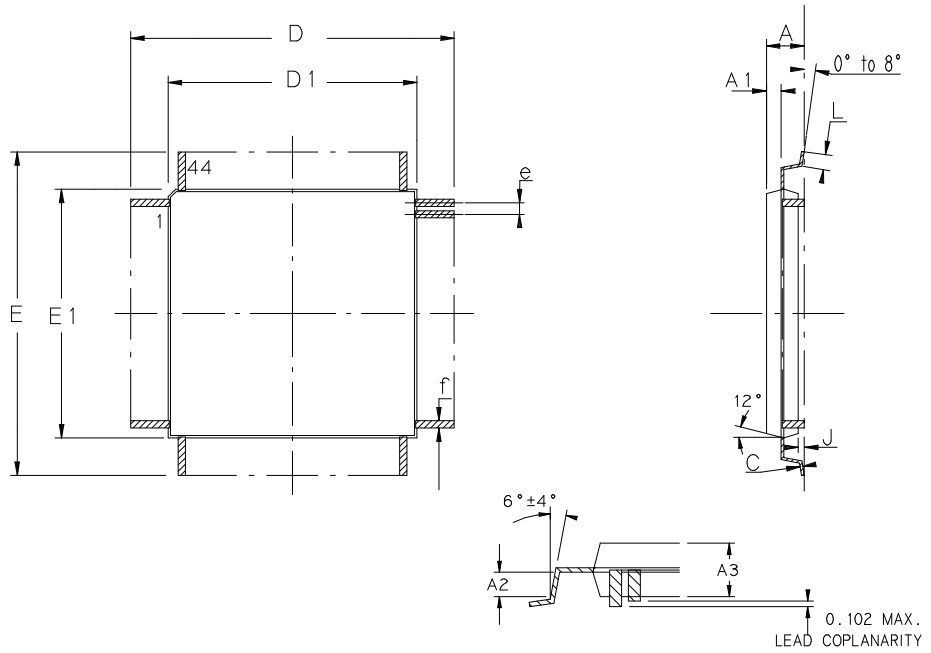


Figure 5.5. Shrink Quad Flat Pack (Plastic)

Table 5.9. VQFP Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	–	1.60	–	.063
A1	0.64 REF		.025 REF	
A2	0.64 REF		.025REF	
A3	1.35	1.45	.053	.057
D	11.90	12.10	.468	.476
D1	9.90	10.10	.390	.398
E	11.90	12.10	.468	.476
E1	9.90	10.10	.390	.398
J	0.05	–	.002	6
L	0.45	0.75	.018	.030
e	0.80 BSC		.0315 BSC	
f	0.35 BSC		.014 BSC	

5.6.2. Pin Assignment

Table 5.10. VQFP Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.5/CEX2/MISO	23	P2.5/A13
2	P1.6/CEX3/SCL/SCK	24	P2.6/A14
3	P1.7/A17/CEX4/SDA/MOSI	25	P2.7/A15
4	RST	26	PSEN#
5	P3.0/RXD	27	ALE/PROG#
6	WAIT#	28	NMI
7	P3.1/TXD	29	EA#/VPP
8	P3.2/INT0#	30	P0.7/AD7
9	P3.3/INT1#	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR#	34	P0.3/AD3
13	P3.7/A16/RD#	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	VSS2	39	VSS1
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T3EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

5.7. VQFP 44 (14×14)

5.7.1. Mechanical Outline

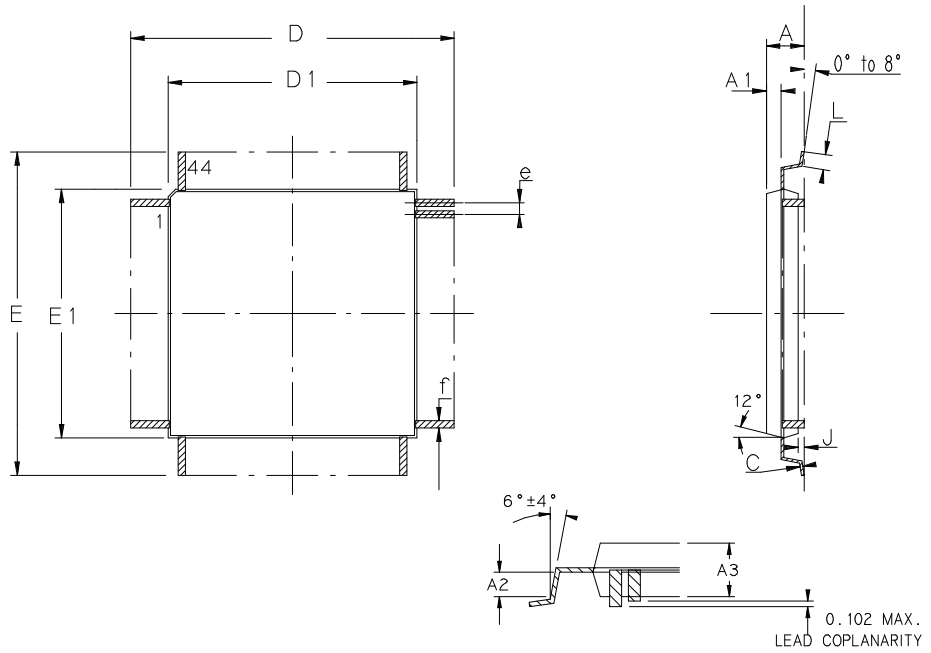


Figure 5.6. Shrink Quad Flat Pack

Table 5.11. VQFP Package Size

	MM		INCH	
	Min	**Max**	**Min**	**Max**
A	–	1.60	–	.063
A1	0.64 REF		.025 REF	
A2	0.64 REF		.025REF	
A3	1.35	1.45	.053 BSC	
D	16.00 BSC		.63 BSC	
D1	14.00 BSC		.55 BSC	
E	16.00 BSC		.63 BSC	
E1	14.00 BSC		.55 BSC	
J	0.05	0.15	.002	6
L	0.45	0.75	.018	.030
e	1.00 BSC		.0394 BSC	
f	0.35	0.5	.0014	.0197

5.7.2. Pin Assignment

Table 5.12. VQFP Pin Assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.5/CEX2/MISO	23	P2.5/A13
2	P1.6/CEX3/SCL/SCK	24	P2.6/A14
3	P1.7/A17/CEX4/SDA/MOSI	25	P2.7/A15
4	RST	26	PSEN#
5	P3.0/RXD	27	ALE/PROG#
6	WAIT#	28	NMI
7	P3.1/TXD	29	EA#/VPP
8	P3.2/INT0#	30	P0.7/AD7
9	P3.3/INT1#	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR#	34	P0.3/AD3
13	P3.7/A16/RD#	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	VSS2	39	VSS1
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T3EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

Section IV

Application Notes

How to Plug a TSC80251G1 Step A in a C51 Board ?

1.1. Introduction

This application note assumes that the user is familiar with C51 microcontrollers.

The purpose of this application note is to address the software and the hardware design considerations when migrating from C51 microcontrollers to the first general purpose TEMIC C251 microcontroller, the TSC80251G1 Step A.

Without changing the code, the average performance will increase by a factor from 2 to 5 (See "Speed Increase" part). If the new instruction set is used, the performance may increase up to a factor of 15.

To plug a TSC80251G1 into a C51 socket, the user has to consider three points:

- the pin–to–pin replacement,
- the programming of the two configuration bytes,
- the speed increase.

Caution:

If the chip is a TSC80251G1–B (See Ordering Information in Product Design Guide), the microcontroller is already programmed to be C51 compatible and the user is not concerned with programming the configuration bytes.

1.2. Pin–to–Pin Replacement

The C51 microcontrollers have 4 NC (Not Connected) pins which are used on TSC80251G1 microcontrollers.
In Step A, these pins must not be floating, so please connect VSS1, VSS2, NMI (respectively pins #1, #23 and #34 in PLCC44) to the Ground and WAIT# (pin #12 in PLCC44) to VCC.
However, the TSC80251G1 Step C will provide a pin–to–pin replacement of C51 microcontrollers without these constraints.

1.3. Configuration Bytes

The TSC80251G1 Step A provides a variety of features and operating modes by programming two configuration bytes CONFIG0 and CONFIG1 (See Figure 1.1. and Figure 1.2.). These bytes are read from specific registers (See Table 1.1.) during the reset of the chip.

Table 1.1. Configuration Bytes Location

Microcontrollers	Location of registers (⋆)	Programming
TSC87251G1 EPROM	On–chip EPROM	by user or ask factory
TSC87251G1 OTPROM	On–chip OTPROM	by user or ask factory
TSC83251G1 MaskROM	Metal Mask	in factory
TSC80251G1 ROMless	Metal Mask	in factory

(⋆) In Step C, user configuration bytes (UCONFIG0 and UCONFIG1) are used instead of CONFIG0 and CONFIG1. In all cases (EPROM, OTPROM, MaskROM, ROMless), the user is able to program these bytes. He must do it unless it has been ordered to factory (EPROM, OTPROM, MaskROM).

CONFIG0 (80h)
Configuration byte 0

–	–	WSA	XALE	RD1	RD0	PAGE	SRC
7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to CONFIG0.
6	–	**Reserved** Set this bit when writing to CONFIG0.
5	WSA	**Wait State A bit** ⇒ Clear to generate one wait state for memory regions 00:, FE: and FF:. Set for no wait states for regions 00:, FE: and FF:.
4	XALE	**Extend ALE bit** Clear to extend the time of the ALE pules from T_{OSC} to 3.$T_{OSC.}$ ⇒ Set to keep the time of the ALE pulse to T_{OSC}.
3	RD1	**Memory Signal Selection bits** Codes specify a 17–bit or 16–bit external address bus and address ranges for RD#, WR# and PSEN# signals. RD1 RD0 RD# P1.7 PSEN# Range 0 0 A16 A17 PSEN# is the read signal for both external data and program address spaces (256 Kbytes). 0 1 A16 I/O pin PSEN# is the read signal for both external data and program address spaces (128 Kbytes). 1 0 I/O pin I/O pin PSEN# is the read signal for both external data and program address spaces (64 Kbytes). ⇒ 1 1 RD# I/O pin PSEN# is the read signal for the external program address spaces (64 Kbytes) and RD# is the read signal for the external data address space (64 Kbytes).
2	RD0	
1	PAGE	**Page Mode Select bit** Clear for page mode with A15:8/D7:0 on Port 2 and A7:0 on Port 0. ⇒ Set for non–page mode 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.

Note:
The arrow (⇒) shows the recommended configuration for C51 compatibility.

Figure 1.1. Configuration Byte 0

4

CONFIG1 (81h)
Configuration Byte 1

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

Bit Number	Bit Mnemonic	Description
7	–	**Reserved** Set this bit when writing to CONFIG1.
6	–	**Reserved** Set this bit when writing to CONFIG1.
5	–	**Reserved** Set this bit when writing to CONFIG1.
4	INTR	**Interrupt Mode bit** ⇒ Clear so that the interrupts push 2 bytes onto the stack (the 2 lower bytes of the PC register). Set so that the interrupts push 4 bytes onto the stack (the 3 bytes of the PC register and the PSW1 register).
3	WSB	**Wait State B** ⇒ Clear to generate one wait state for memory region 01:. Set for no wait states for region 01:.
2	–	**Reserved** Set this bit when writing to CONFIG1.
1	–	**Reserved** Set this bit when writing to CONFIG1.
0	EMAP	**EPROM 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)

Note:
The arrow (⇒) shows the recommended configuration for C51 compatibility.

Figure 1.2. Configuration Byte 1

The default configuration for OTPROM/EPROM microcontrollers is CONFIG0 = 1111 1111b and CONFIG1 = 1111 1111b.

The configuration recommended for C51 compatibility (hardware and software) is CONFIG0 = 1101 1110b and CONFIG1 = 1110 0111b.
In details:

- SRC = 0 for binary mode.
- PAGE = 1 for non–page mode.
- INTR = 0 to push 2 bytes onto the stack.

- RD1 = 1 and RD0 = 1 for memory signals.
- XALE = 0 to keep the ALE pulse width (0 wait state).
- WSA = 0 and WSB = 0 to add 1 wait state for all external memory regions.
- EMAP = 1.

The first four points are mandatory to follow for C51 compatibility while the others are optional. In fact, the internal code memory of the TSC80251G1 is faster than most of the external memory and peripheral device available. When interfacing the TSC80251G1 with slower devices (used on C51 board), a wait state can be used to extend the external system bus cycle. The TSC80251G1 Step A can be configured to generate 0 or 1 wait state for ALE, PSEN#, RD# and WR# signals. With 0 wait state, the pulse of these signals is 1 oscillator period. With 1 wait state, the pulse widths are extended to 3 oscillator periods.

The user may need to check the timing specifications for the TSC80251G1 and the external devices to ensure that the TSC80251G1 will be able to interface successfully with external devices.

1.4. Speed Increase

The TSC80251G1 uses the new C251 microcontroller's core that is different from the traditional C51's one. It is designed based on a pipelined architecture and a register–based machine. Therefore, the execution time decreases and the user must consider changing the the timing loops or sequences of C51 code running on TSC80251G1 when he relies on C51 execution time.

For example, a delay time is used to provide a 13 µs delay in a 12 MHz 80C51 application. The code and the taken time are shown on Listing 1.1.

```
        ; Internal or External Code Execution          Taken time in number of cycles

              MOV   R0,#06h                            ; 1
LOOP    :     DJNZ  R0,LOOP                            ; 2

        ; 1 cycle = 12 oscillator periods
        ; Total oscillator periods = [ 1 + ( 6 x 2 ) ] x 12 = 156
        ; Total time taken in the timing loop = 156 x 83.33 ns = 13 µs
```

Listing 1.1. Loop Executed at 12 MHz on 80C51

In the same application, the 80C51 microcontroller is replaced by a 12 MHz TSC80251G1 programmed using the recommended configuration for C51 compatibility in binary mode. No modification is made to the code shown on Listing 1.1. But to simplify timing loop calculation, the code is executed in internal memory. The code and the taken time is shown on Listing 1.2.

```
        ; Internal Binary Code Execution          Taken time in number of states

              MOV   R0,#06h                       ; 1
LOOP:         DJNZ  R0, LOOP                      ; 5 and 2 for exiting the loop

        ; 1 state =  2 oscillator periods
        ; Total oscillator clock periods = [ 1 + ( 5 x 5 + 2 ) ] x 2 = 56
        ; Total time taken in the timing loop = 56 x 83.33 ns = 4.66 µs
```

Listing 1.2. Loop Executed at 12 MHz on TSC80251G1 Without Code Modification

To maintain the same delay time, some changes in the assembly code are needed. The new code and the taken time are shown on Listing 1.3.

```
        ; Internal Binary Code Execution          Taken time in number of states

              MOV   R0,#0Fh                       ; 1
LOOP:         DJNZ  R0, LOOP                      ; 5 and 2 for exiting the loop

        ; 1 state =  2 oscillator periods
        ; Total oscillator clock periods = [ 1 + ( 15 x 5 + 2 ) ] x 2 = 156
        ; Total time taken in the timing loop = 156 x 83.33 ns = 13 µs
```

Listing 1.3. Loop executed at 12 MHz on TSC80251G1 With Code Modification

Calculating execution time for TSC80251G1 is not as easy as for 80C51. The execution time depends on whether the code is fetched: from internal or external memory. When bytes are fetched in external memory in non–page mode and with one wait state, each fetch takes 6 oscillator periods (3 states) and the corresponding speed increase factor is 2. When no wait states are added, this factor becomes 3. When bytes are fetched in external memory in page mode, each fetch requires only 2 oscillator periods (1 state) and this factor becomes 6.
It is the same factor for internal code execution, which is the fastest case. But the average speed increase factor is 5 since all instructions do not take the same execution time.
The user may note that, on Listing 2, the execution time is only divided by a factor of 2.7 because DJNZ is one of the slowest instruction.

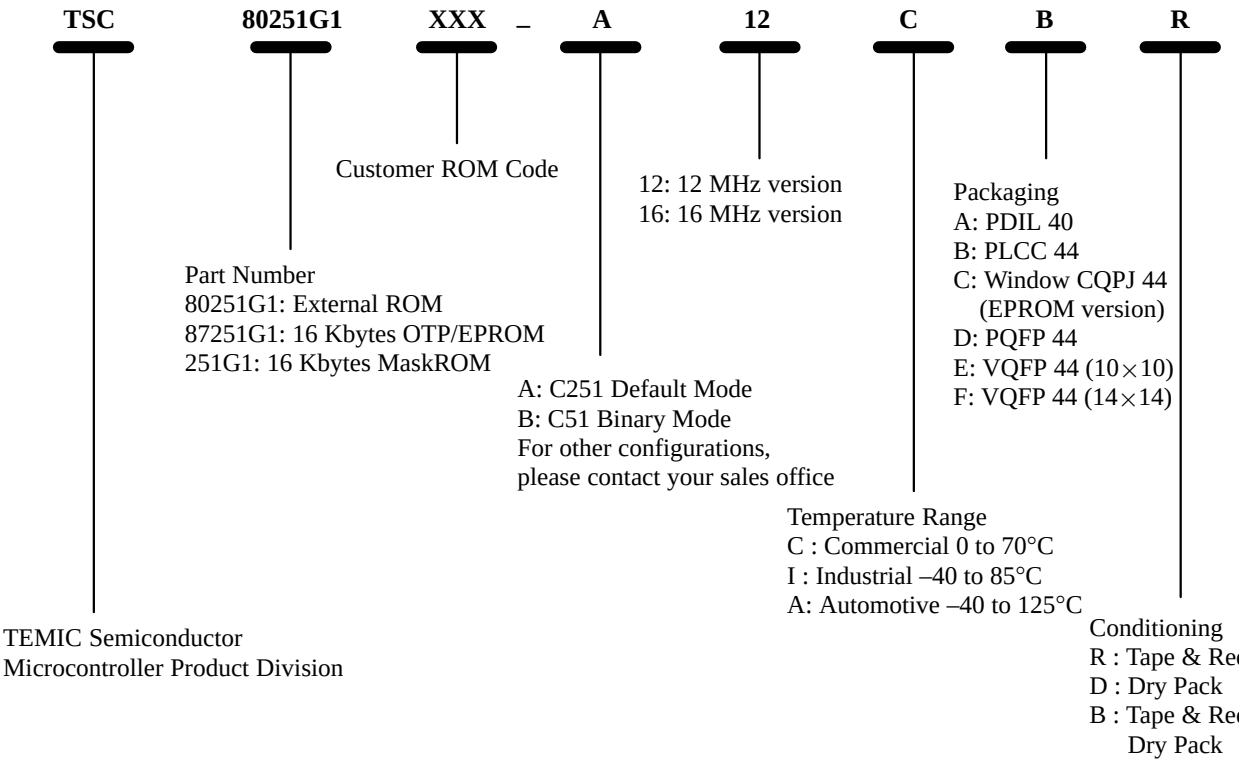
Caution:

To execute code in page–mode, an address latch must be connected to Port 2 instead of Port 0 and the hardware of 80C51 applications must be changed.
To use existing 80C51 applications without changing your hardware, please execute code in non–page mode.

Section V

Ordering Information

Ordering Information



Examples

Part Number	Description
TSC80251G1–A16CBR	ROMless, C251 Default Mode, 16 MHz, PLCC 44, 0 to 70°C, Tape and Reel
TSC80251G1–B16CBR	ROMless, C51 Binary Mode, 16 MHz, PLCC 44, 0 to 70°C, Tape and Reel
TSC87251G1–12CB	OTP, 12 MHz, PLCC 44, 0 to 70°C
TSC87251G1–12CC	EPROM, 12 MHz, CQPJ 44, 0 to 70°C

Development Tools

Part Number	Description
TSC80251G1–SKA	Software Starter Kit Keil
TSC80251G1–SKB	Software Starter Kit Tasking
TSC80251G1–EKA	Evaluation Kit Keil
TSC80251G1–EKB	Evaluation Kit Tasking

Product Marking :

TEMIC
Customer P/N
Temic P/N
Ⓜ© Intel'95
YYWW Lot Number

5