TSC 80251A1

Extended 8–bit Microcontroller with Analog Interfaces

Datasheet – 1996

Table of Contents

Section III: Electrical and Mechanical Information

Section IV: Ordering Information

Section V: TEMIC Addresses

List of figures

Section I: Introduction to TSC80251A1

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: Pulse Measurement Unit

Chapter 6: Event and Waveform Controller

Chapter 7: 8–bit Analog to Digital Converter

Chapter 8: Power Monitoring and Management

Chapter 9: Interruption System

Section III: Electrical and Mechanical Information

Chapter 1: DC Characteristics

Chapter 2: AC Characteristics

Chapter 3: ADC Characteristics

Chapter 4: EPROM Programming

Chapter 5: Packages

List of tables

Section I: Introduction to TSC80251A1

Chapter 4: TSC80251A1 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 6: Event and Waveform Controller

Chapter 8: Power Monitoring and Management

Chapter 9: Interruption System

Section III: Electrical and Mechanical Information

Chapter 1: DC Characteristics

Chapter 2: AC Characteristics

Chapter 3: ADC Characteristics

Chapter 4: EPROM Programming

Chapter 5: Packages

General Introduction

Extended 8–bit Microcontroller with Analog Interfaces

The TSC80251A1 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 analog interface structures.

Three major peripheral blocks have been implemented to provide this facility to the designer:

- Analog to Digital Converter: 4 inputs at 8–bit resolution.
- Pulse Measurement Unit (PMU): 3 modules used to interface to smart analog sensors.
- Event and Waveform Controller (EWC): 5 programmable Counters e.g. for Pulse Width Modulation (PWM) or Compare/Capture functions.

1.1. Application focus

Typical applications for these products are CD–ROM, Card or Barcode readers, Monitors, Car Navigation Systems, Airbag and Brake Systems, as well as all kinds of Industrial Control and Measurement Equipment. With the high instruction throughput, the TSC80251A1 products are focussing 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 CPU–Performance is increased by up to 5 times, using existing 80C51 code without any modification.

Using the new C251 Instruction Set, the performance will be increased by up to 15 times, at the same clock rate.

This performance enhancement is based on the 16–bit instruction bus and additional internal 8 and 16–bit data busses. The 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, combined with 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.

All technical information in this document about core features are related to the core revision A (A–stepping). A new core revision, B/C (B–stepping) is presently in preparation.

Both versions are upward compatible, so that no problem will appear if an A–stepping product is replaced by a B–stepping one.

The major differences are some additional features in the configuration bytes and a modified emulator interface which will not affect existing application.

A new document will be released as soon as the first TSC80251A1 product will be available in revision C.

1.3. TSC80251A1 Products

The TSC80251A1 is available as a ROMless version (TSC80251A1) or with on–chip Mask Programmable ROM (TSC83251A1). The TSC87251A1 is an EPROM version or OTPROM (One Time Programmable) compatible with the Mask ROM version.

The standard production packages are 44 pins PLCC or TQFP.

The products can be delivered as 12 or 16 MHz versions at 5 Volts and in all major temperature ranges.

1.4. TSC80251A1 Documentation and Tools

The following documentation and Starter tools are available to allow the full evaluation of the TEMIC TSC80251A1 product range:

- “TSC80251A1 Microcontroller”

  Contains all information about the A1 derivatives (Block diagram, Memory mapping, Ports, Peripheral description, Electrical Mechanical and Ordering Information...).
- “TSC80251 Programmer‘s Guide”

  Contains all information for the programmer.

  (Architecture, Instruction Set, Programming, Development tools)
- “TSC80251 Design Guide”

  Contains a summary of available Application Notes for an easier usage of the TSC80251 and its major peripherals.
- “TSC80251A1 Starter Kit”

  This kit enables the TSC80251A1 to be evaluated by the designer.

  It contains the following:
  - C–Compiler (limited to 2 Kbytes of code)
  - Assembler
  - Linker
  - TSC80251A1 Simulator
  - Optionally TSC80251A1 Evaluation Board with ROM–Monitor

  Please visit our WWW for updated versions in ZIP format.
- “TSC80251A1 Development Tools”

  See chapter ”Development Tools” 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 TSC80251A1

Core Features

Based on the extended 8–bit C251 Architecture, the TSC80251A1 includes a complete set of new or improved C51 compatible peripherals as well as a 4 channels 8–bit A/D converter for communication with the analog environment.

The key features of the new C251 Architecture are:

- 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 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)
- Up to a factor 3 of code size reduction (when a C for 80C51 program is recompiled in C251 language)
- Reduction of RFI and power consumption (reduced operating frequency)
- Complete System Development Support
  - Compatible with existing tools
  - New tools available: Compiler, Assembler, Debugger, ICE
- Efficient C language support

Product Features

- 1 Kbyte of internal RAM
- TSC83251A1: 24 Kbytes of on-chip masked ROM
- TSC87251A1: 24 Kbytes of internal programmable ROM (OTP or UV erasable in window package)
- TSC80251A1: ROMless version
- External memory space (Code/Data): 256 Kbytes
- Four 8–bit parallel I/O Ports (Ports 0, 1, 2 and 3 of the standard 80C51)
- Two 16–bit Timers/Counters (Timers 0 and 1 of the standard 80C51)
- Serial I/O Port : full duplex UART (80C51 compatible)
- Three PMU: Pulse Measurement Unit for smart analog interface

  For each of the three modules:
  - 8–bit prescaler
  - 8–bit Timer for period and width measurements (duty cycle)
  - The measurement can start either on the rising or on the falling edge
  - One interrupt
  - Only one port line is used
- EWC: Event and Waveform Controller
  - High-speed output
  - Compare/Capture inputs
  - PWM: Pulse Width Modulator
  - Watchdog Timer capabilities
  - Compatible with PCA: Programmable Counter Array (5 x 16–bit modules)
- 8–bit Analog to Digital Converter
  - 4 channels
  - Conversion time: 600 clock periods (37.5 µs at 16 MHz)
- Power Management
  - Power–On reset (integrated on the chip)
  - Power–Off flag (cold and warm resets)
  - Power-Fail detector
  - Power consumption reduction
    - Software programmable system clock
    - Idle and Power–Down modes
- Power Supply: 5V ± 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) and TQFP44(**)

*Please contact your sales office for availability of speed options
** Please contact your sales office for TQFP availability

Block Diagram

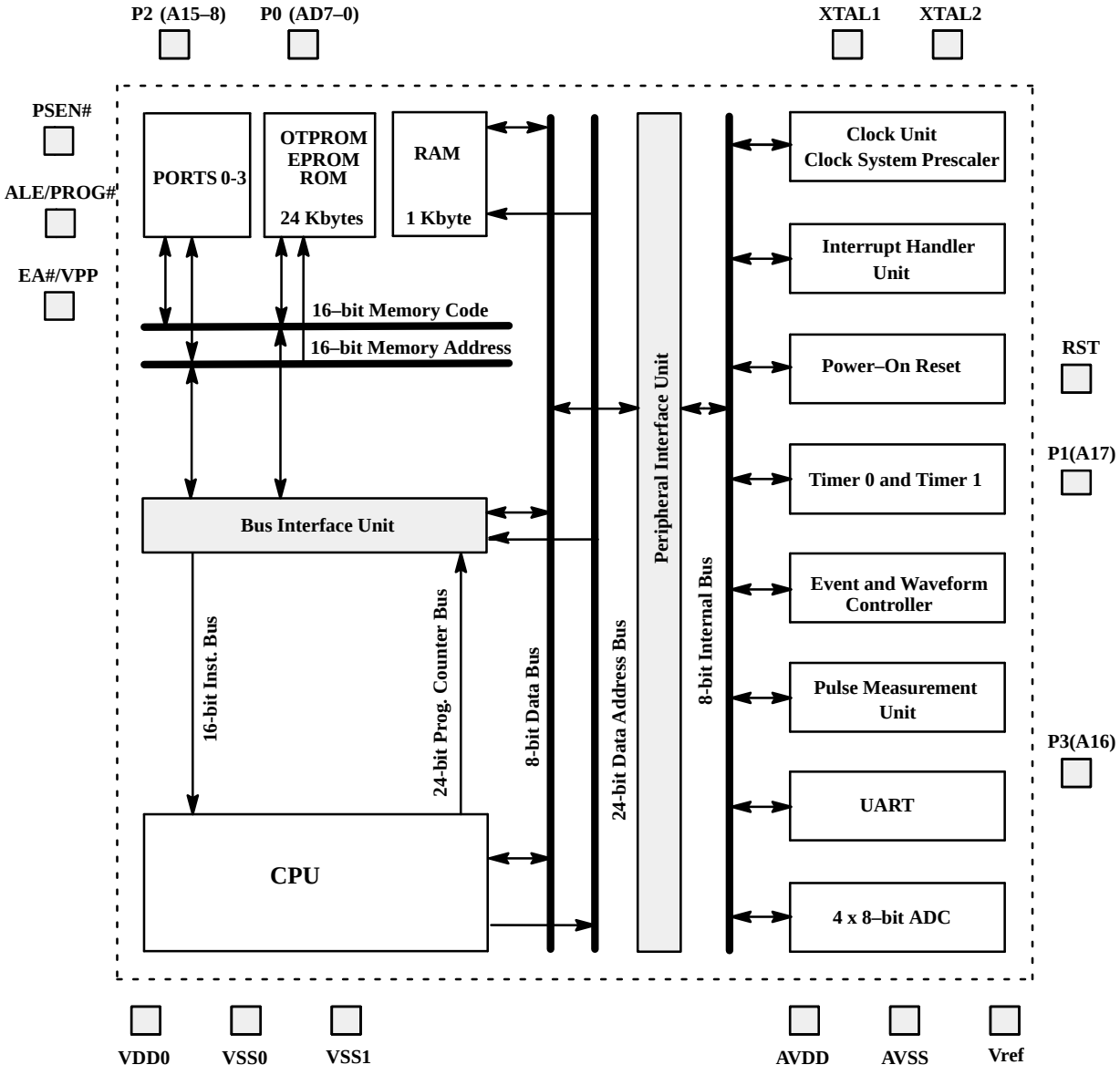


Figure 3.1. TSC80251A1 block diagram

Pin Description

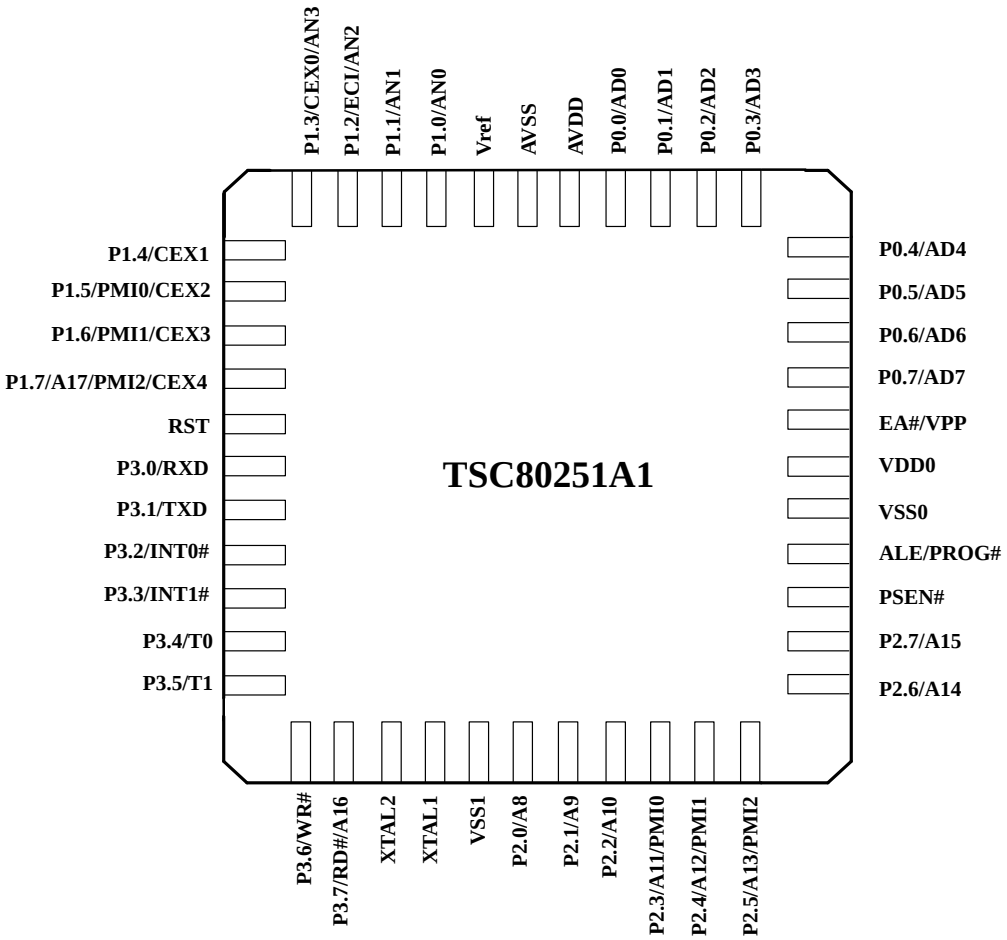


Figure 4.1. TSC80251A1 pin description

Table 4.1. TSC80251A1 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 AN0 : Analog Input 0, P1.1 AN1 : Analog input 1, P1.2 ECI : EWC External Clock input. AN2 : Analog input 2, P1.3 CEX0 : EWC module 0 Capture input/PWM output. AN3 : Analog input 3, P1.4 CEX1 : EWC module 1 Capture input/PWM output, P1.5 PMI0 : Pulse Measurement input 0, CEX2 : EWC module 2 Capture input/PWM output. P1.6 EAD6 : External Address line 6, PMI1 : Pulse Measurement input 1, CEX3 : EWC module 3 Capture input/PWM output. P1.7 A17 : Address line for the 256–Kbyte memory space depending on the byte CONFIG0 (See Table 1.2.), PMI2 : Pulse Measurement input 2, CEX4 : EWC module 4 Capture input/PWM output.
P2.0:7	I/O	**Port 2** This is an 8–bit bidirectional I/O port with internal pull-ups. 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 on the byte CONFIG0.

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 configuration byte CONFIG0.
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.
Vref	I	**Voltage reference for the Analog to Digital Converter**
VSS0	GND	**Digital Ground**
VDD0	PWR	**Digital Supply Voltage**
VSS1	GND	**Digital Ground**
AVSS	GND	**Analog Ground**
AVDD	PWR	**Analog Supply Voltage**
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 VDD0. 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 TSC80251A1 derivatives are detailed in this section.

1.2. Configuration

The TSC80251A1 derivatives provide design flexibility by configuring certain 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 bytes stored on the stack by an interrupt.

The choice of internal program/code or external memory access is made through the External Access pin (EA#, see paragraph 1.3.2.). The internal memories of the TSC80251A1 derivatives are detailed in paragraph 1.3. “Memory Mapping”.

The choice of external memory interface is detailed in this section:

- Page Mode and Wait States
- External Memory Signals

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 discusses the choice of external 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 oscillator to reduce the power consumption (See “Power Monitoring and Management” chapter) and the speed of the external cycles is then reduced accordingly.

TSC80251A1 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.

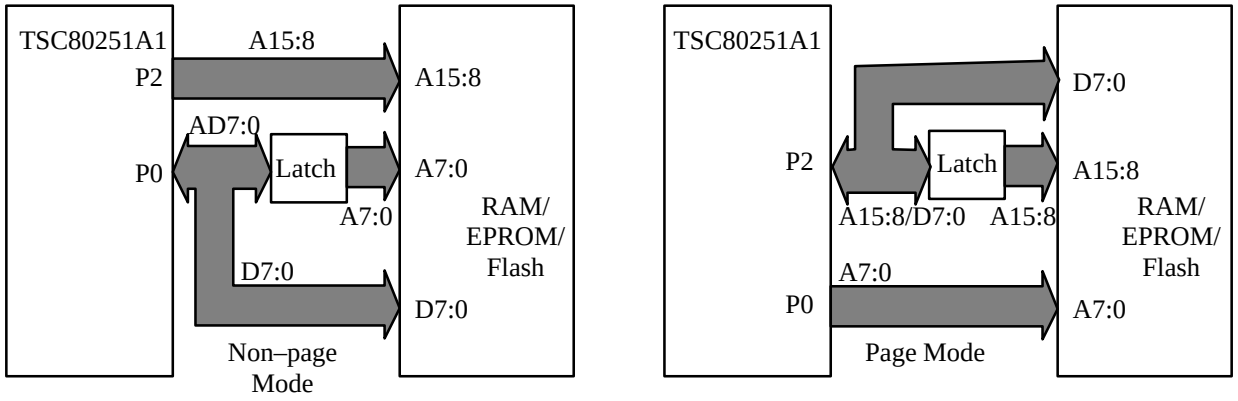


Figure 1.1. Bus structure in non–page mode and page mode

The 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.

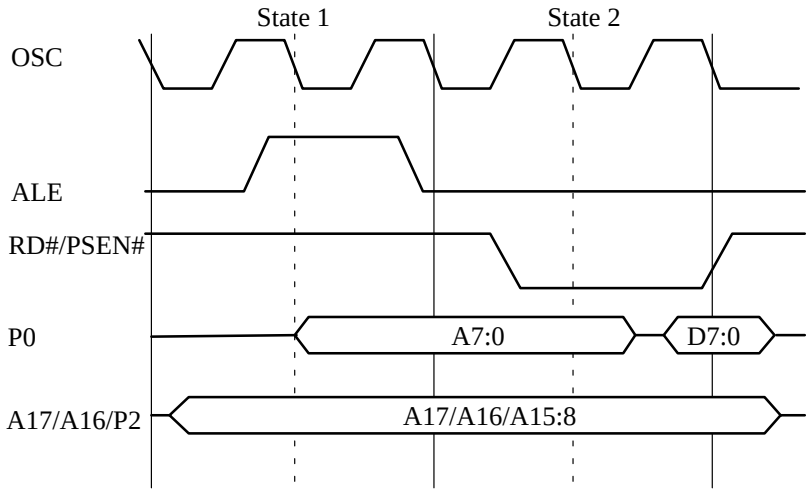


Figure 1.2. External bus cycle: code fetch, non–page mode

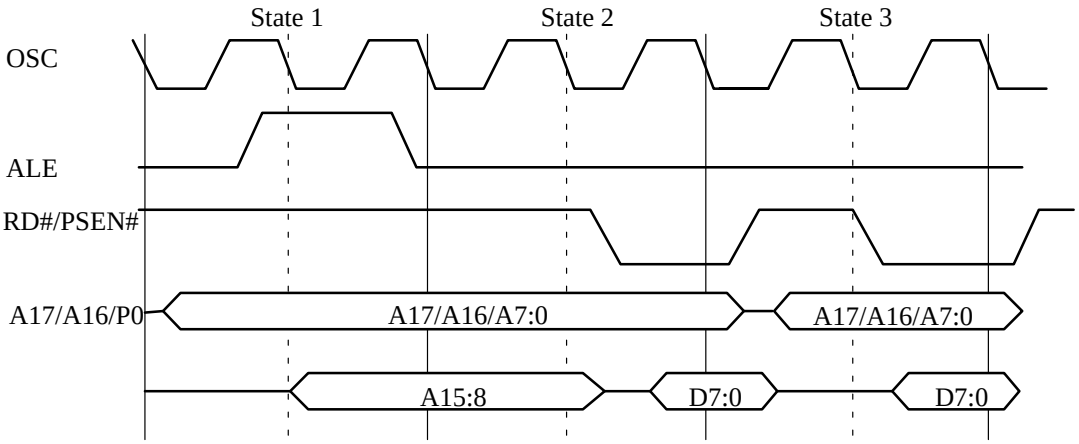


Figure 1.3. External bus cycle: code fetch, page mode

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 1.2.2.). The Wait State B bit (WSB bit in CONFIG1 byte) adds one state for external data accesses only (See segment 01: in paragraph 1.2.2.).

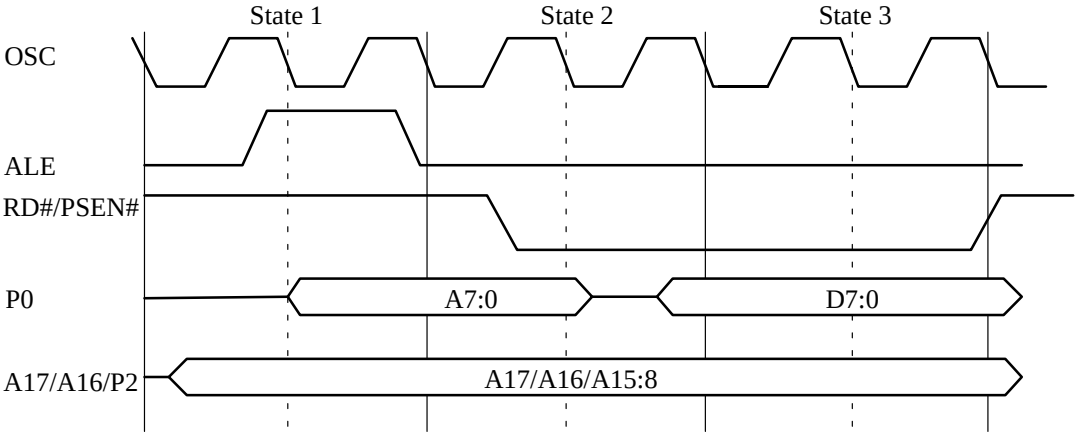


Figure 1.4. External bus cycle: code fetch with one RD#/PSEN# wait state in non–page mode

1.2.2. 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 TSC80251A1 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.

1.2.2.1. 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 (P0, P2, A16, A17) to control 256 Kbytes in four segments. In this configuration, the program/code and data spaces share the same external memory segments.

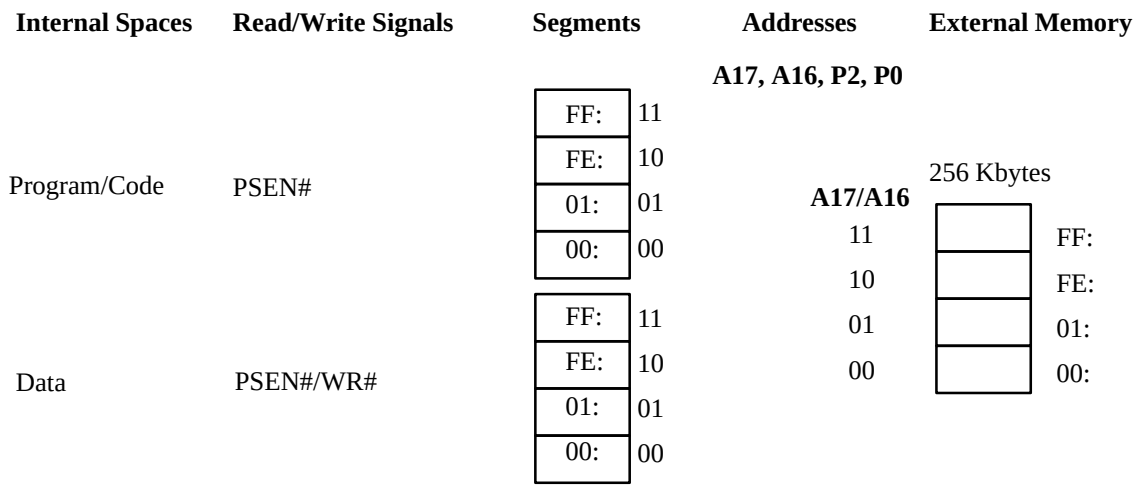


Figure 1.5. Internal/external memory segments (RD1:0 = 00)

1.2.2.2. 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.

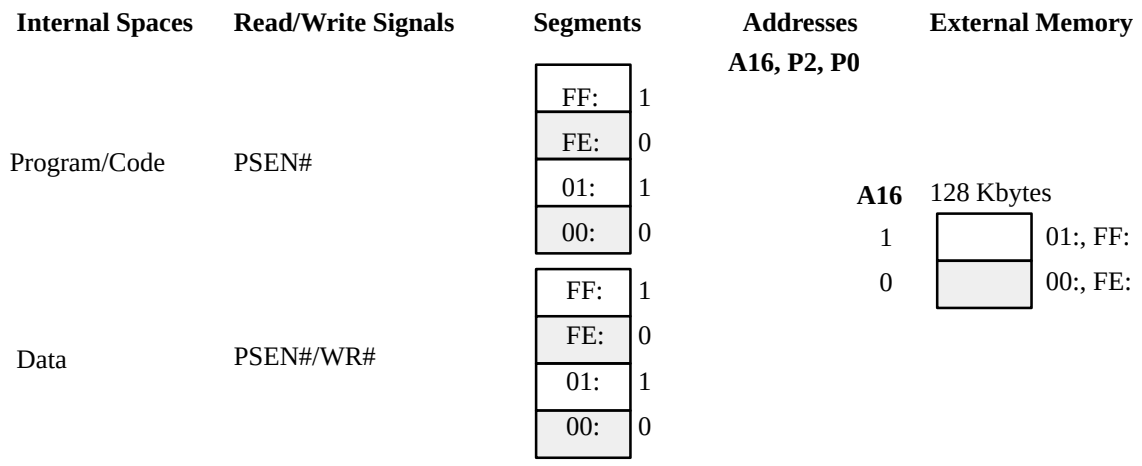


Figure 1.6. Internal/external memory segments (RD1:0 = 01)

1.2.2.3. How to address 64 Kbytes

Two I/O pins (P1.7/A17, 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.

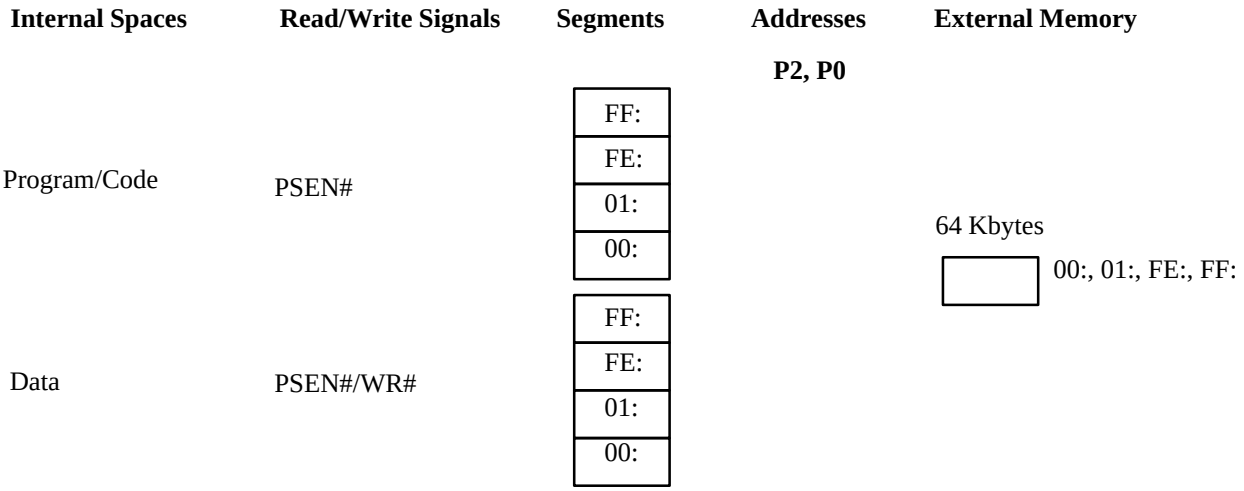


Figure 1.7. Internal/external memory segments (RD1:0 = 10)

1.2.2.4. 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 read while RD# is used as a read signal and WR# is used as a write signal for data memory accesses. Sixteen address

bits are provided externally (Port 0, Port 2). 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 one internal space.

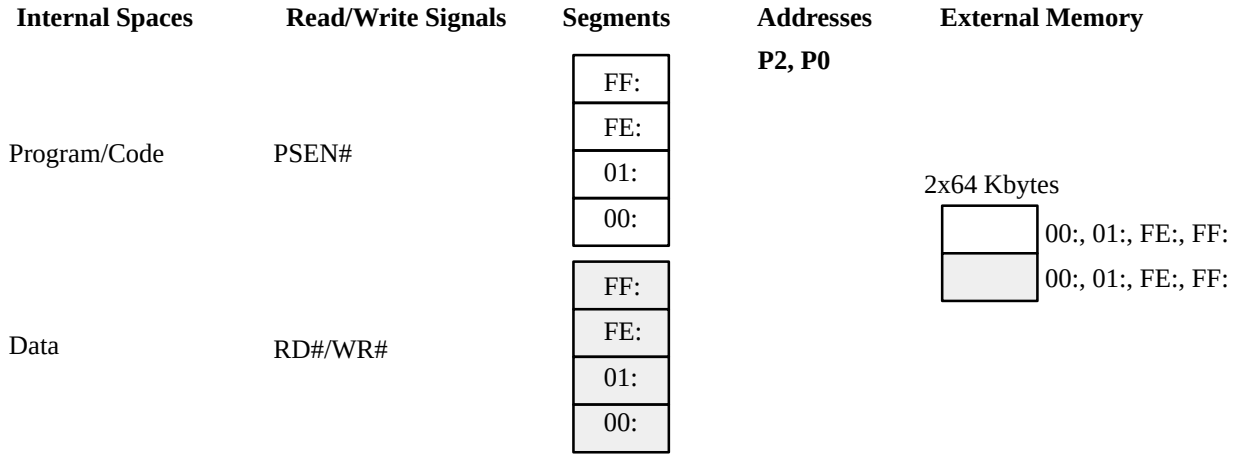


Figure 1.8. Internal/external memory segments (RD1:0 = 11)

1.3. Memory Mapping

The specific internal memories of the TSC80251A1 derivatives fall into the following categories:

- 2 Configuration bytes,
- 24 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 TSC87251A1 EPROM and OTPROM versions, these bytes are programmable in an EPROM area (See “EPROM programming” chapter). For the TSC83251A1 masked ROM versions, these bytes are additional information provided in a masked ROM area. For the TSC80251A1 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 TSC80251A1 derivatives.

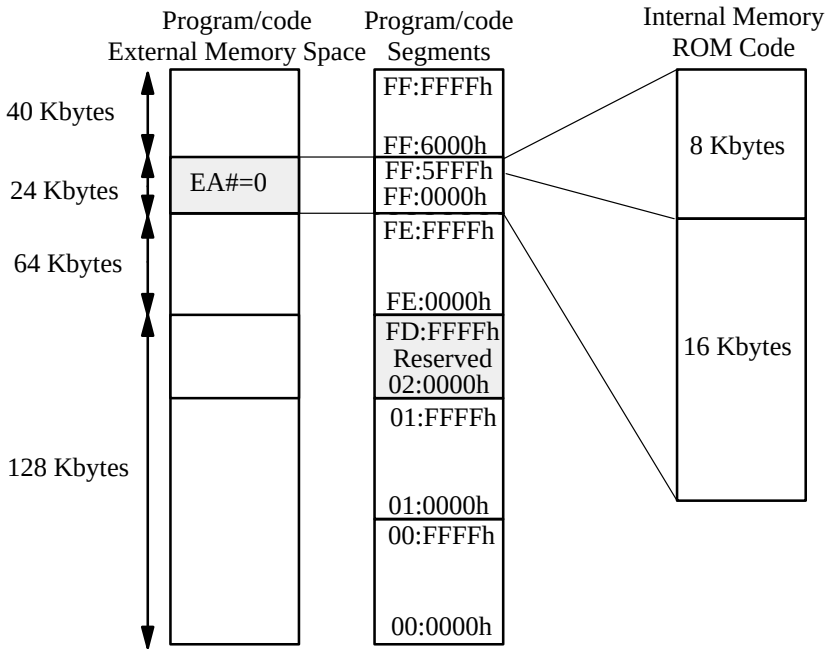


Figure 1.9. Programmable Memory Mapping

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 24–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. Table 1.1. lists the minimum times to fetch on–chip and external memory.

Table 1.1. Minimum Times to fetch two bytes of code

Type of code memory	State times
On–chip code memory	1
External memory (page mode)	2
External memory (nonpage mode)	4

For the TSC87251A1 EPROM and OTPROM versions, the internal program/code is programmable in EPROM (See “EPROM programming” chapter). For the TSC83251A1 masked ROM versions, the internal program/code is provided in a masked ROM. For the TSC80251A1 ROMless versions, there is no possible internal program/code and EA# must be tied to a low level. In fact, for TSC83251A1 and TSC87251A1 versions, the upper 8 Kbytes of the internal ROM are also mapped in the data space (See paragraph 1.3.3.).

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:5FF8h–FF:5FFFh). Because of its pipeline capability, the 80C251A1 may attempt to prefetch code from external memory (at an address above FF:5FF8H/FF:5FFFH) and thereby disrupt I/O Ports 0 and 2. Fetching code constants from these eight 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 it going into the reserved area).

1.3.3. Data Memory

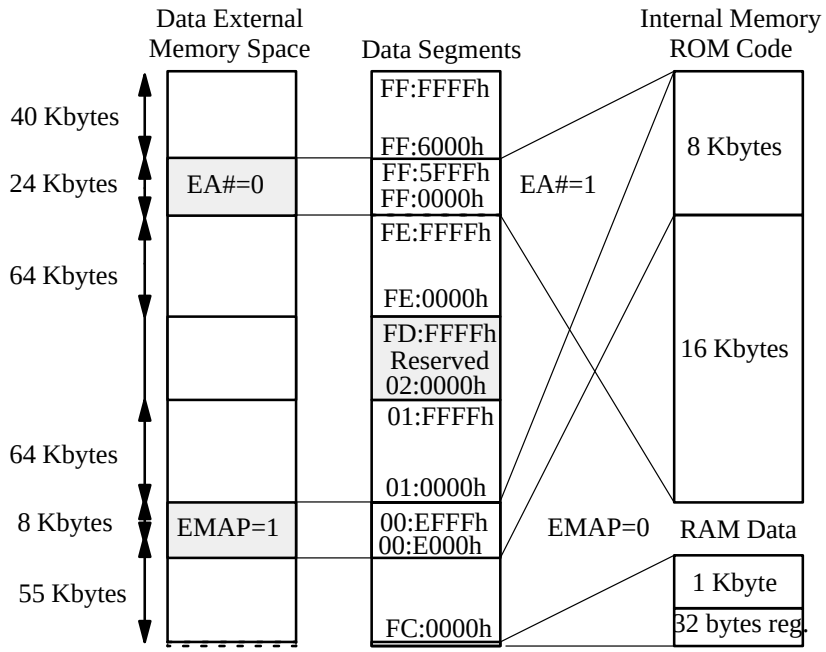


Figure 1.10. Data Memory Mapping

The split of the internal and external data memory space is shown on Figure 1.10. All the TSC80251A1 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 TSC83251A1 and TSC87251A1 versions, its upper 8 Kbytes are also mapped in the data space 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 and the TSC80251A1 derivative is running as a ROMless, the code is actually fetched in the corresponding external memory (i.e. the upper 8 Kbytes of the lower 24 Kbytes of segment FF:). If EMAP bit is set, the internal ROM is not accessible through the data memory space.

All the accesses to the portion of the data space with no internal memory mapped onto are redirected to the external memory, see paragraph 1.2.2. to determine to which external memory location each segment actually maps.

1.3.4. Special Function Registers

The Special Function Registers (SFRs) of the TSC80251A1 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)
- Serial Port and Baud Rate Generator registers (SCON, SBUF, SADDR, SADEN, BDRCON, BRL)
- Pulse Measurement Unit registers (PMU, PMCON, PMSCAL0, PMSCAL1, PMSCAL2, PMPER0, PMPER1, PMPER2, PMWID0, PMWID1, PMWID2)
- Event and Waveform Controller registers:
  - Counters (CCON, CMOD, CMOD0, CMOD1, CMOD2, COF, CRC, CIE, CL0, CL1, CL2, CL3, CL4, CH0, CH1, CH2, CH3, CH4)
  - Compare/Capture (CCAPM0, CCAPM1, CCAPM2, CCAPM3, CCAPM4, CCAPL0, CCAPL1, CCAPL2, CCAPL3, CCAPL4, CCAPH0, CCAPH1, CCAPH2, CCAPH3, CCAPH4)
- Analog to Digital Converter registers (ADCON, ADAT)
- Power monitoring/management and clock control registers (PCON, PFILT, POWM, CKRL)
- 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 within S: 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 1.2. SFR addresses and Reset values

F8h		CH = CH0 0000 0000	CCAP0H XXXX XXXX	CCAP1H XXXX XXXX	CCAP2H XXXX XXXX	CCAP3H XXXX XXXX	CCAP4H XXXX XXXX	CMOD3 0000 0000
F0h	*B*** *0000 0000*				CH1 0000 0000	CH2 0000 0000	CH3 0000 0000	CH4 0000 0000
E8h		CL = CL0 0000 0000	CCAP0L XXXX XXXX	CCAP1L XXXX XXXX	CCAP2L XXXX XXXX	CCAP3L XXXX XXXX	CCAP4L XXXX XXXX	CMOD2 0000 0000
E0h	*ACC*** *0000 0000*	COF XXX0 0000	CRC 0000 0000	CIE XXX0 0000	CL1 0000 0000	CL2 0000 0000	CL3 0000 0000	CL4 0000 0000
D8h	CCON 0000 0000	CMOD 00XX X000	CCAPM0 X000 0000	CCAPM1 X000 0000	CCAPM2 X000 0000	CCAPM3 X000 0000	CCAPM4 X000 0000	CMOD1 0000 0000
D0h	*PSW*** *0000 0000*	*PSW1*** *0000 0000*						
C8h								
C0h						ADCON XXX0 0X00	ADAT* XXXX XXXX	
B8h	IPL0 0000 0000	SADEN 0000 0000					*SPH*** *0000 0000*	
B0h	P3 1111 1111	IE1 X000 0000	IPL1 0000 0000	IPH1 0000 0000				IPH0 0000 0000
A8h	IE0 0000 0000	SADDR 0000 0000	PMSCAL0 XXXX XXXX	PMSCAL1 XXXX XXXX	PMSCAL2 XXXX XXXX	PMCON X000 X000	PMSTAT X000 X000	
A0h	P2 1111 1111		PMPER0* XXXX XXXXh	PMWID0* XXXX XXXXh	PMPER1* XXXX XXXXh	PMWID1* XXXX XXXXh	PMPER2* XXXX XXXXh	PMWID2* XXXX XXXXh
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000				PMU XXXX XXX0
90h	P1 1111 1111							
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	CKRL 0000 1000	POWM 0XX0 0000
80h	P0 1111 1111	*SP*** *0000 0111*	*DPL*** *0000 0000*	*DPH*** *0000 0000*	*DPXL*** *0000 0001*		PFILT 0000 1000	PCON 000X 0000
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F

* read only
**C251 core registers described in the TSC80251 Programmer's Guide

reserved

S:00h – S7Fh 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** The value read from this bit is indeterminate. Do not set this bit.
7	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
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, 2	RD1, RD0	**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 P3.7 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
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 80C51microcontrollers).
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

Note:

To configure the TSC80251A1 in C51 microcontroller mode, use the following bit values in CONFIG0: 1101 1110B.

CONFIG1
Configuration byte 1

–	–	–	INTR	WSB	–	–	EMAP
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	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** 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	EMAP	**EPROM Map bit** Clear to map the upper 8 Kbytes of on–chip code memory (FF:3000h-FF:5FFFh) to 00:C000h-00:FFFFh. Set to map the upper 12 Kbytes of on–chip code memory to FF:3000h-FF:5FFFh.

Figure 1.12. Configuration byte 1

Note:

To configure the TSC80251A1 in C51 microcontroller mode, use the following bit values in CONFIG1: 1110 0111B.

Parallel I/O Ports

2.1. Introduction

The TSC80251A1 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 TSC80251A1 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 2.1. Port pin descriptions

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

Pin Name	Type	Alternate Pin Name	Alternate Description	Alternate Type
P1.0	I/O	AN0	Analog input 0	I
P1.1	I/O	AN1	Analog input 1	I
P1.2	I/O	ECI AN2	EWC external clock input Analog input 2	I I
P1.3	I/O	CEX0 AN3	EWC module 0 Capture input/PWM output Analog input 3	I/O I
P1.4	I/O	CEX1	EWC module 1 Capture input/PWM output	I/O
P1.5	I/O	PMI0 CEX2	PMU input 0 EWC module 2 Capture input/PWM output	I I/O
P1.6	I/O	PMI1 CEX3	PMU input 1 EWC module 3 Capture input/PWM output	I I/O
P1.7	I/O	A17 PMI2 CEX4	Address line 17 PMU input 2 EWC module 4 Capture input/PWM output	I/O I I/O

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

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 I/O

Notes:

- EWC = Event Waveform Controller
- PMU = Pulse Measurement Unit
- PWM = Pulse Width Modulation

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 2.1.).
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 driver FET.

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.

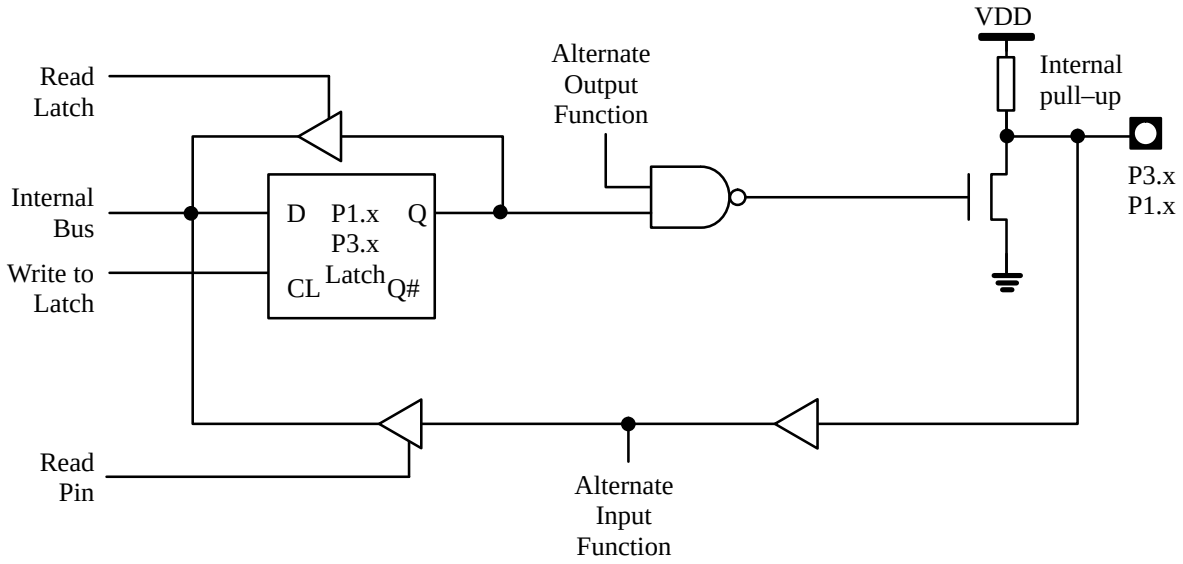


Figure 2.1. Port 1 and Port 3 structure

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.

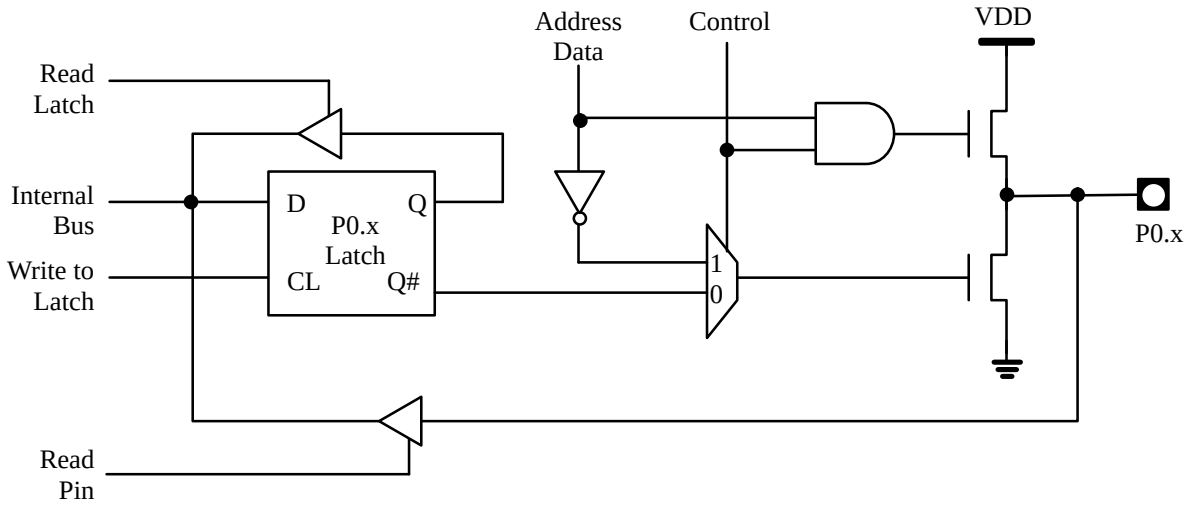


Figure 2.2. Port 0 structure

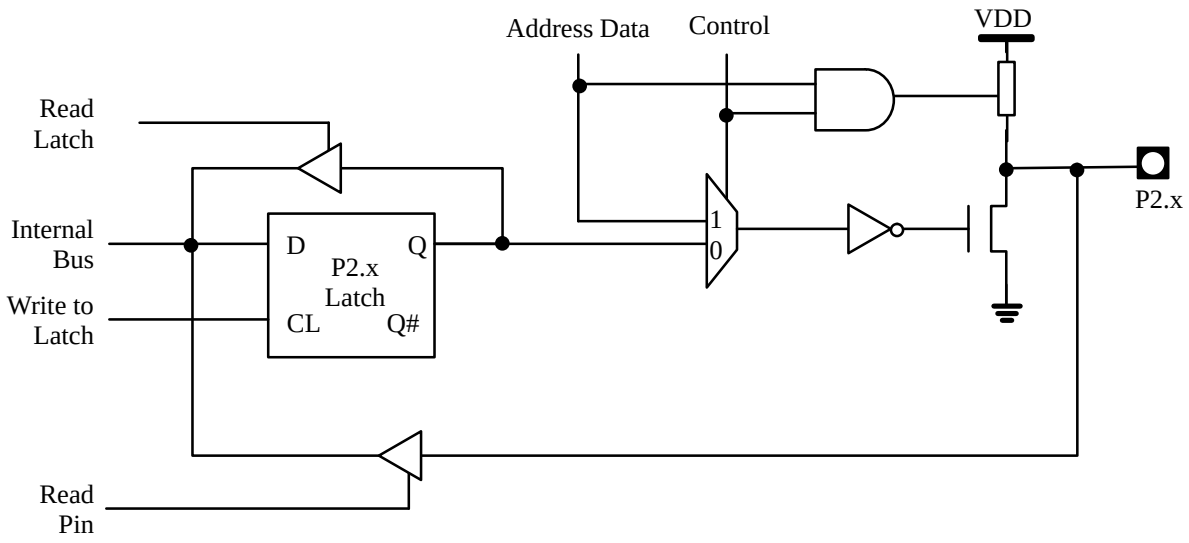


Figure 2.3. Port 2 structure

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

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

Instruction	Description	Example
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

It is not obvious the last three instructions in this list are Read–Modify–Write instructions. These instructions read the Port (all 8 bits), modify the specifically addressed bit and write the new byte back to the latch. These Read–Modify–Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor cannot 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.

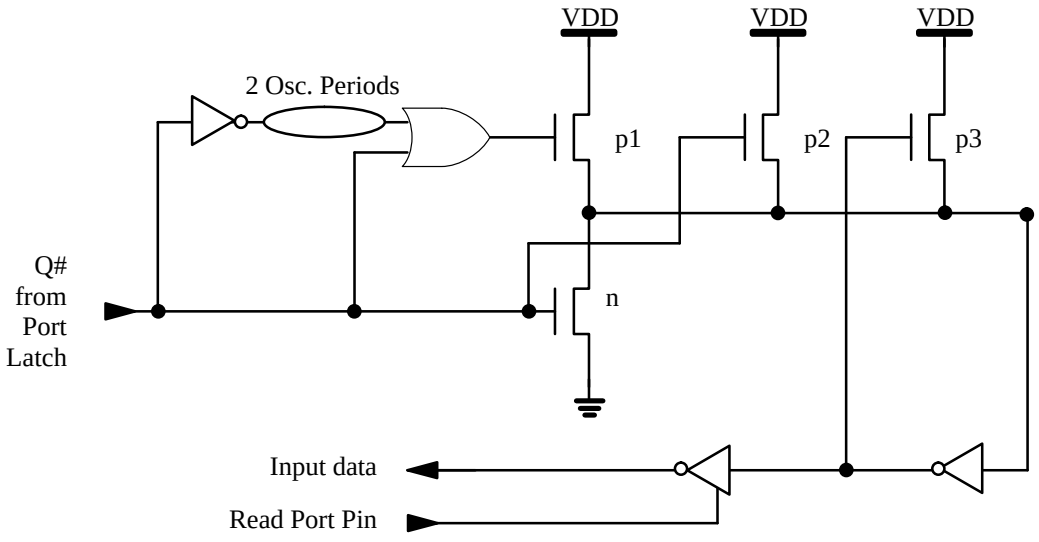


Figure 2.4. Internal pull–up configurations

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. 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. 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 TSC80251A1 CPU writes FFh to the Port 0 register for all external memory bus cycles. This overwrites previous information in Port 0. In contrast, the Port 2 register is unmodified for external bus cycles. When address bits or data bits are not on the Port 2 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 TSC80251A1 uses PSEN# or RD# for data reads (See “Configuration bits RD0 and RD1”).

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 2.2. lists the instructions that can be used for the these bus widths.

Table 2.2. 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
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 TSC80251A1 contains two general–purpose, 16–bit Timers/Counters. Although they are identified as Timer 0 and Timer 1, you can independently configure each to operate in a variety of modes as a Timer or as an event Counter. Each Timer employs two 8–bit Timer registers, used separately or in cascade, to maintain the count. 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. Two of the SFRs provide programmable control of the Timers as follows:

- Timer/Counter Mode Control register (TMOD).
- Timer/Counter Control register (TCON) for Timer 0 and Timer 1.

These registers are described 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
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
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

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

3.2. Timer/Counter Operations

For example, a basic operation is Timer registers THx and TLx (x = 0 or 1) connected in cascade to form a 16–bit Timer. Setting the run control bit (TRx) 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 register. Setting the run control bit does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. Timer 0 and Timer 1 can also be controlled by external pin INTx# to facilitate pulse width measurements.

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.

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.

For Counter operation (C/Tx# = 1), the Timer register counts the negative transitions on the Tx external input pin. When the sample of the external inputs is high in one cycle and low in the next, the Counter is incremented. 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. and Figure 3.4. show the logical configuration of each mode.

Timer 0 is controlled by the four low–order bits of TMOD register (See Figure 3.6.) and bits 0, 1, 4 and 5 of TCON register (See Figure 3.5.). 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. This setup can be used to make pulse width measurements.

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.

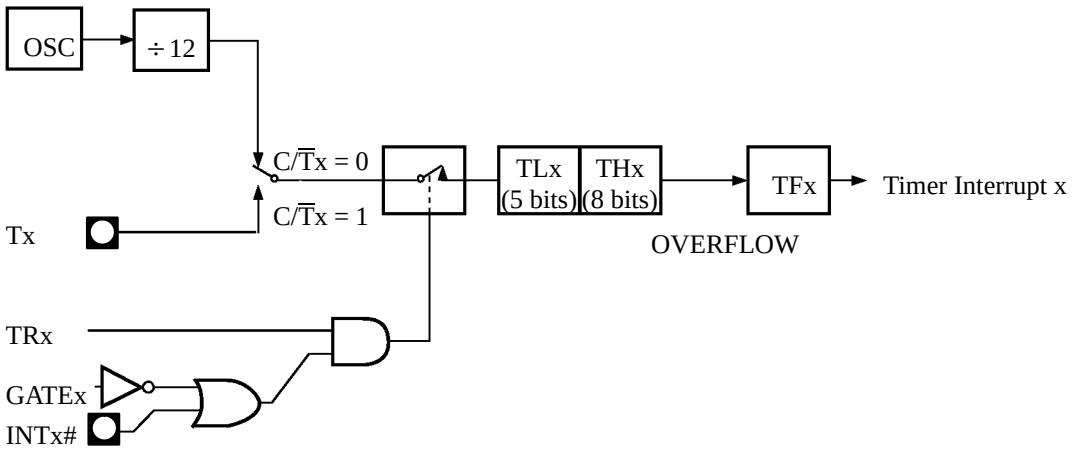


Figure 3.1. Timer/Counter x (x = 0 or 1) in mode 0 and mode 1

3.3.2. Mode 1 (16–bit Timer)

Mode 1 configures Timer 0 as a 16–bit Timer with TH0 and TL0 connected in cascade (See Figure 3.2.). The selected input increments TL0.

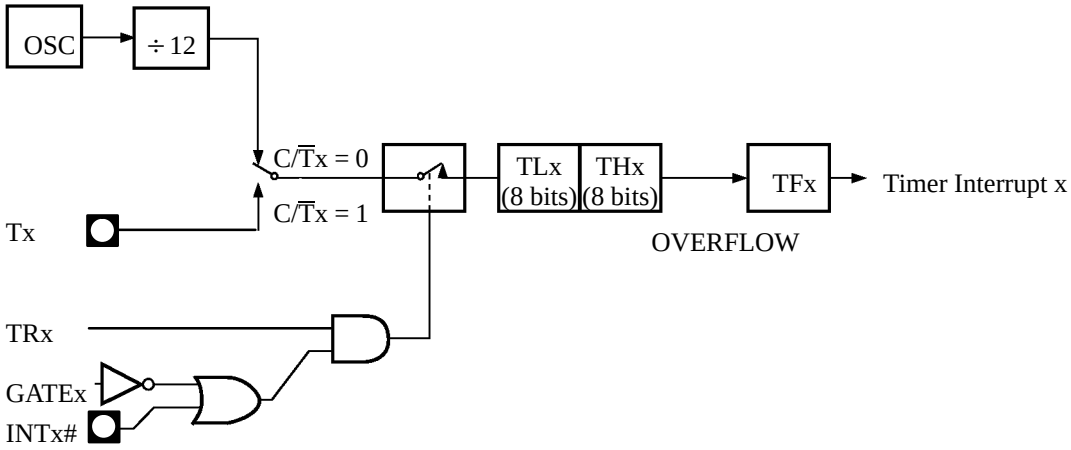


Figure 3.2. Timer/Counter x (x = 0 or 1) in mode 1

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.

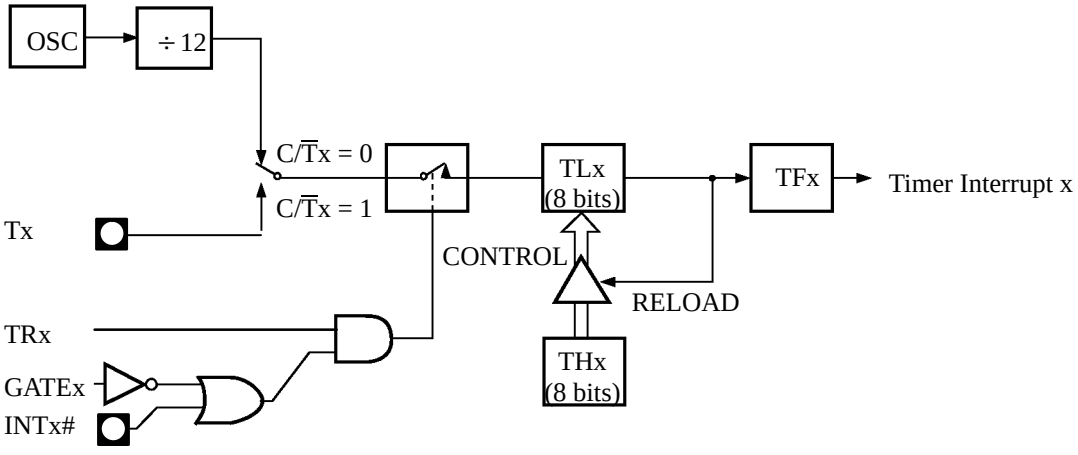


Figure 3.3. Timer/Counter x (x = 0 or 1) in mode 2

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

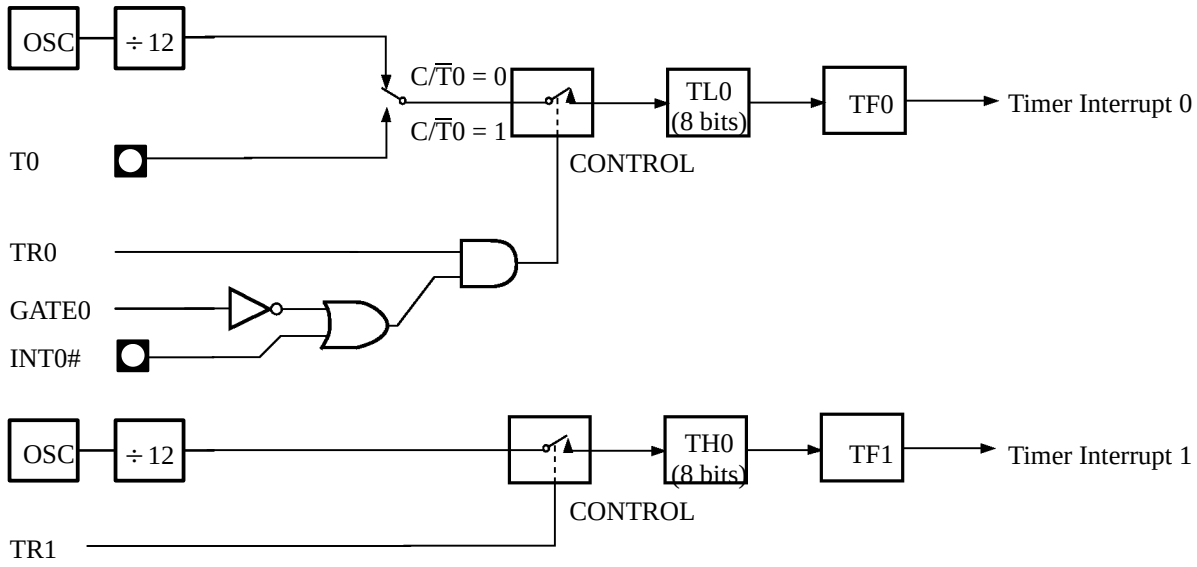


Figure 3.4. Timer/Counter in mode 3 : Two 8-bit Counters

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 3.6.) and bits 2, 3, 6 and 7 of TCON register (See Figure 3.5.). 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 Timer register TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin INT1# to control Timer operation. This setup can be used to make pulse width measurements.

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 connected in cascade (See Figure 3.2.). The selected input increments TL1.

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.). Overflow from TL1 sets overflow flag TF1 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. Registers

TCON (088h)
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.
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.
4	TR0	**Timer 0 Run Control bit** Clear to turn off Timer/Counter 0. Set to turn on Timer/Counter 0.
3	IE1	**Interrupt 1 Edge flag** Cleared by hardware when interrupt is processed if edge-triggered (See IT1). Set by hardware when external interrupt is detected 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 3.5. TCON register

TMOD (089h)
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/Counter 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select bit** Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from T1 input pin).
5	M11	**Timer 1 Mode Select bits** M11 M01 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH1) with 5–bit prescalar (TL1) 0 1 Mode 1: 16–bit Timer/Counter 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL1). Reloaded from TH1 at overflow 1 1 Mode 3: Timer 1 halted. Retains count.
4	M01	
3	GATE0	**Timer 0 Gating Control bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select bit** Cleared for Timer operation (input from internal system clock) Set for Counter operation (input from T0 input pin).
1	M10	**Timer 0 Mode Select bit** M10 M00 Operating mode 0 0 Mode 0: 8–bit Timer/Counter (TH0) with 5–bit prescalar (TL0). 0 1 Mode 1: 16–bit Timer/Counter. 1 0 Mode 2: 8–bit auto–reload Timer/Counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3: TL0 is an 8–bit timer/counter. TH0 is an 8–bit timer using timer 1's TR1 and TF1 bits.
0	M00	

Reset value = 0000 0000B

Figure 3.6. TMOD 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 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 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. Figure 4.1. is a block diagram of the Serial Port.

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
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

Mnemonic	Description	Address
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
SADDR	**Serial Address** Defines the individual address for a slave device connected on the serial lines.	S:0A9h
SADEN	**Serial Address Enable register** Specifies the mask byte that is used to define the given address for a slave device.	S:0B9h
BDRCON	**Baud Rate Control register** Enables and configures the internal Baud Rate register.	S:09Bh
BRL	**Baud Rate Reload register** Contains the auto–reload value of the Baud Rate Generator.	S:09Ah

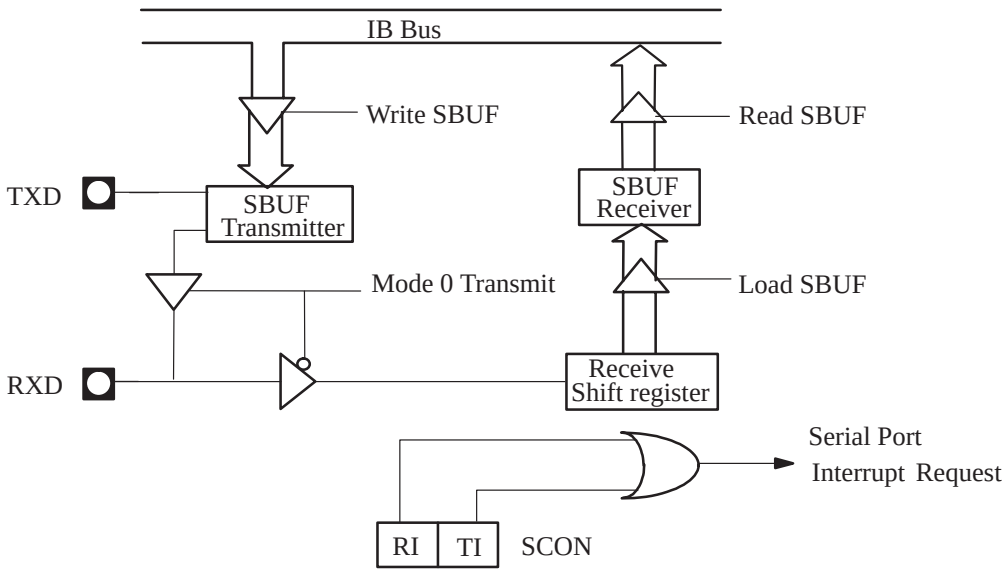


Figure 4.1. Serial Port block diagram

4.2. Modes of Operation

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

4.3. 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.

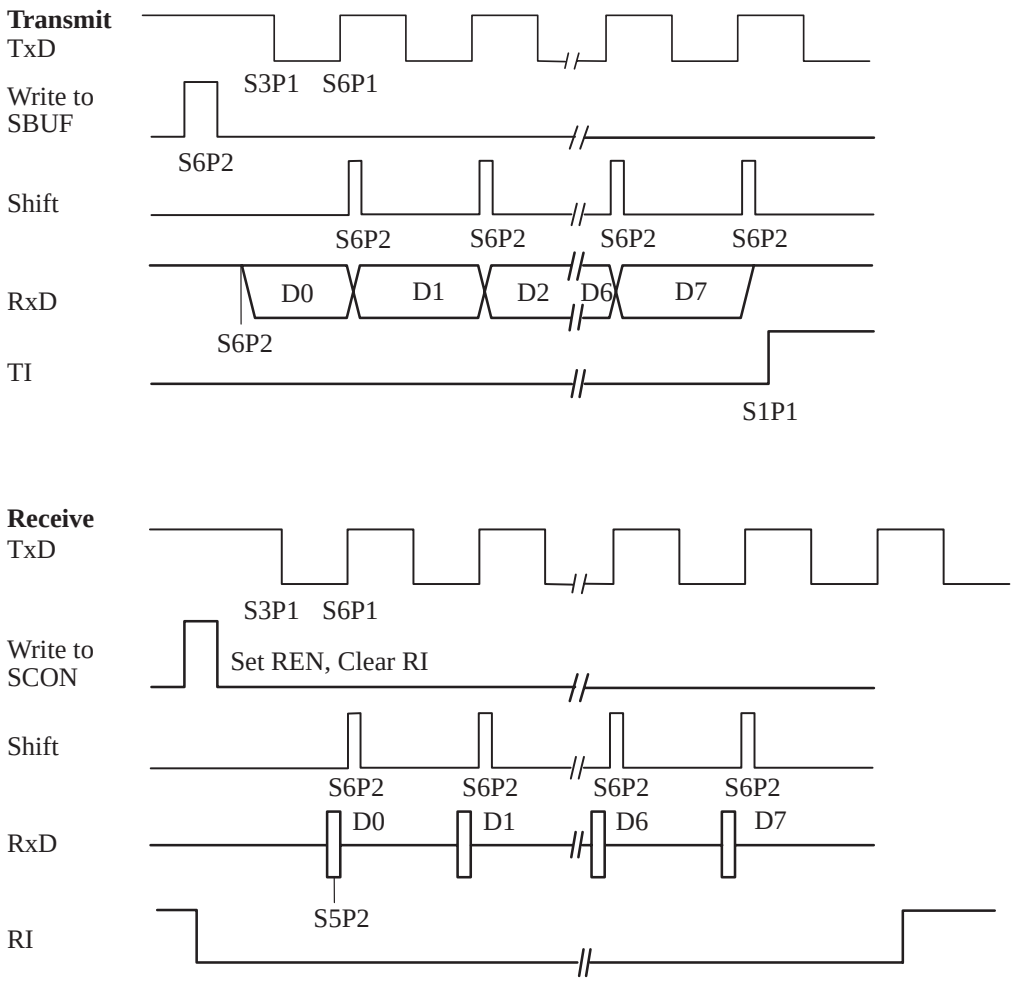


Figure 4.2. Mode 0 timings

4.3.1. 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.

4.3.2. Reception (Mode 0)

To start a reception in mode 0, write to the SCON register. Clear bits SM0, SM1 and RI 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 to indicate acompleted reception. Software can then read the received byte from SBUF.

4.4. 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 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 internal Baud Rate Generator.

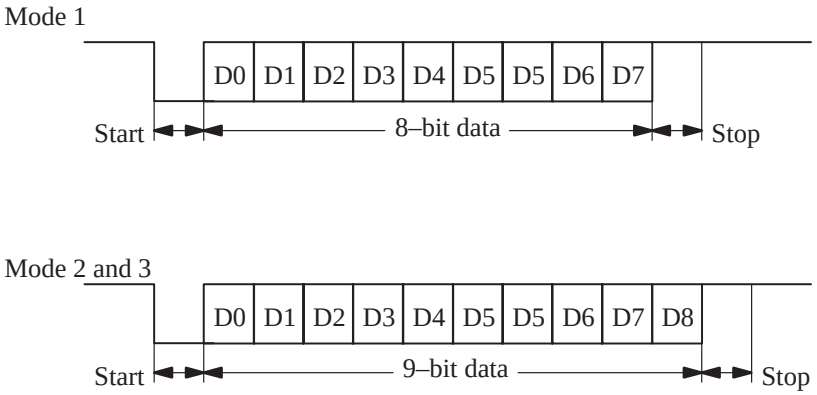


Figure 4.3. Data frames (Modes 1, 2 and 3)

4.4.1. 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.

4.4.2. 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.

4.5. 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.

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

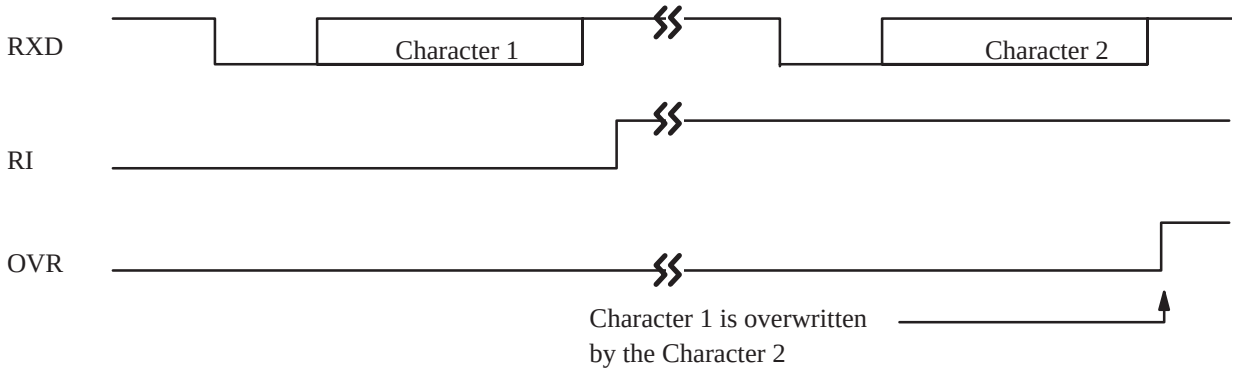


Figure 4.4. Overrun Error (Modes 1, 2 and 3)

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

4.7. 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 microcontroller 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.

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

4.8. 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.

Notes:

- 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).

To support automatic address recognition, a device is identified by a given address and a broadcast address.

4.8.1. 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`).

4.8.2. 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 0FFh.

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.

4.8.3. 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.

4.9. Baud Rates

4.9.1. Internal Baud Rate Generator

The Baud Rate Control register (BDRCON, see Figure 4.9. is added to the TSC80251A1 derivatives in order to manage the new functionality of the UART. Two Baud Rate Generators can supply the transmission clock to the UART: Timer 1 and the internal Baud Rate Generator as detailed below

4.9.2. Baud Rate for Mode 0

The transmission clock is either the internal Baud Rate Generator or the internal fixed prescaler. This selection is done by setting bit SRC in BDRCON register. The transmission clock is shown in Figure 4.5.

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.

4.9.3. Transmission Clock Selection

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

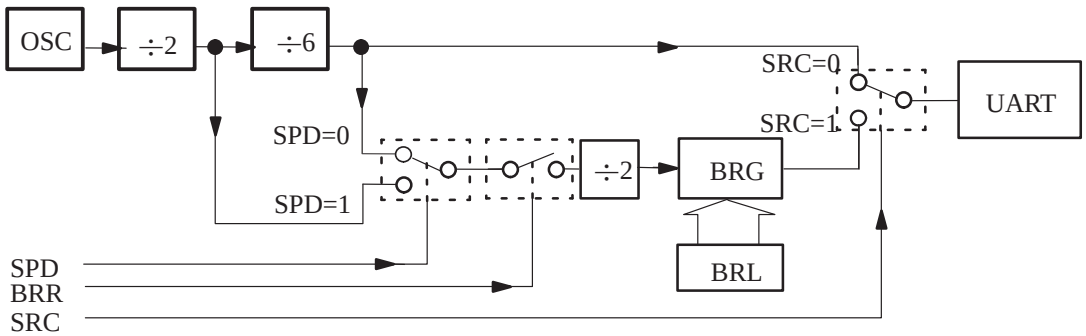


Figure 4.5. Clock transmission sources in mode 0

4.9.4. Baud Rate for Modes 1 and 3

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

4.9.4.1. 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 as follows:

Mode 1 and 3,

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

and if the Baud Rate is known the value of TH1 is:

$$TH1 \quad 256 \div \frac{2^{SMOD1} \quad f_{OSC}}{384 \quad \text{Baud\_Rate}}$$

The configuration is shown in Figure 4.6.

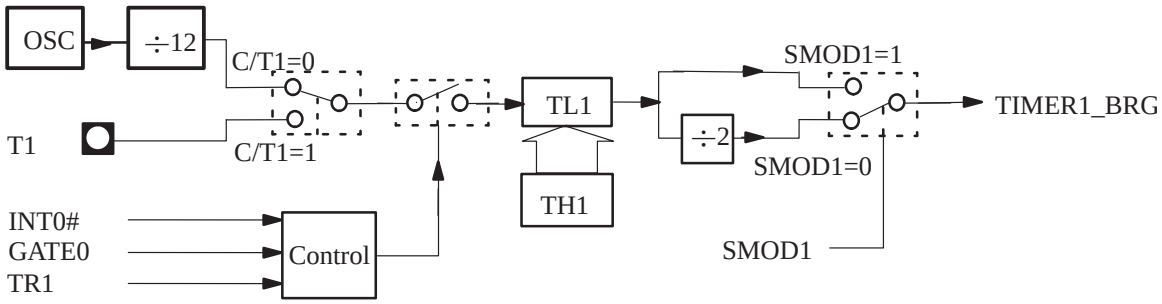


Figure 4.6. Timer 1 as Baud Rate Generator in modes 1 and 3

4.9.4.2. Internal Baud Rate Generator

When the internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow, the value of SPD bit (Speed Mode) and the value of the SMOD1 bit (Serial Mode).

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

$$\text{BRL} \quad 256 \div \frac{2^{SMOD1} \quad F_{OSC}}{64 \quad \text{Baud\_Rate}}$$

If the slow Mode is selected (SPD = 0, default mode), the Baud Rate is as follows:

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

$$\text{BRL} \quad 256 \div \frac{2^{SMOD1} \quad F_{OSC}}{384 \quad \text{Baud\_Rate}}$$

The configuration is shown in the Figure 4.7.

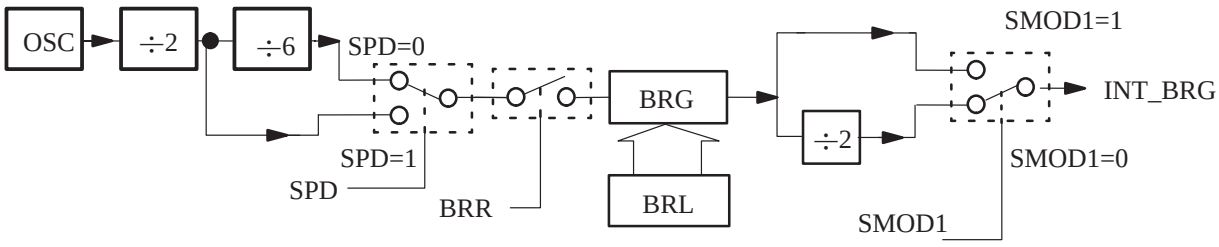


Figure 4.7. Internal Baud Rate Generator in modes 1 and 3

4.9.4.3. Baud Rate Selection

The Baud Rate Generator for transmit and receive clocks can be selected separately via the BDRCON register (See Figure 4.10.)

Figure 4.8. gives the configuration of RBCK and TBCK bits to select the source of RX Clock and TX Clock.

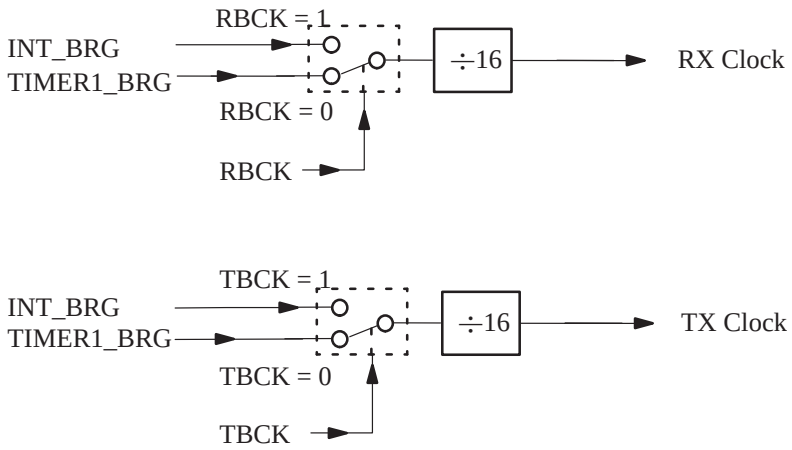


Figure 4.8. Baud Rate Generator selection

4.9.5. 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.

The formula is given below: $\text{Baud\_Rate} \quad \frac{2^{SMOD1} \quad F_{OSC}}{64}$

The configuration is shown in Figure 4.9.

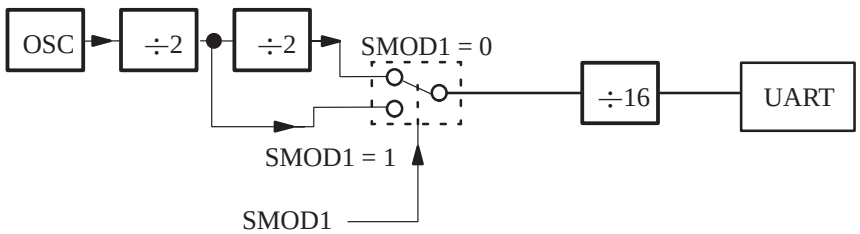


Figure 4.9. UART in mode 2

4.10. Registers

BDRCON (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** = 1, selects the INTERNAL Baud Rate Generator, = 0, selects teh 1/12 clock as the Baud Rate Generator (fixed transmission clock in Mode 0)

Reset value = XXX0 0000B

Figure 4.10. BDRCON register

BRL (9Ah)
Baud Rate Reload register (8–bit)

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

Reset value = 0000 0000B

Figure 4.11. BRL register

SADDR (0A9h)
Serial Address register

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

Reset value = 0000 0000B

Figure 4.12. SADDR register

SADEN (0B9h)
Serial Address Enable register

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

Reset value = 0000 0000B

Figure 4.13. SADEN register

SBUF (099h)
Serial Buffer register

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

Reset value = XXXX XXXXB

Figure 4.14. SBUF register

SCON (098h)
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 4.15. SCON register

Pulse Measurement Unit

5.1. Introduction

This chapter describes the Pulse Measurement Unit (PMU) which allows to measure the width and the period of pulses. It is useful for each application using a smart analog sensor which provide a Pulse Width Modulated information.
With standard peripherals, measuring both the period and the width of pulses series involve two Timers, hence two I/O Port lines. The PMU is specially designed to measure the period and the width of pulses using only one Timer and one I/O Port line. Compared to the standard solution, this new one saves one I/O Port line.

5.2. Description

Just after reset, the Pulse Measurement Mode selection bit (PMMOD) bit is equal to zero which places the PMU in test mode (PMU register, see Figure 5.13.). This bit **must be set to one** before any PMU configuration, otherwise the TSC80251A1 behavior is unpredictable.
The PMU includes three identical modules, as shown in Figure 5.1. Each module features one Pulse Measurement Input (PMIn) connected to one pin of Port 1 which provides the pulses to measure. The internal oscillator provide a clock reference common to all the modules to count cycles between pulse edges. When a new measurement is detected, the corresponding Pulse Measurement Finished flag (PMFn) is set. However, if the PMU Timer overflows before the measurement completion, the corresponding PMU overflow flags (PMVn) is set. When any of these flags is set, the PMU interrupt request which is shared by the three modules is sent to the Interrupt System (see IS in section 9).

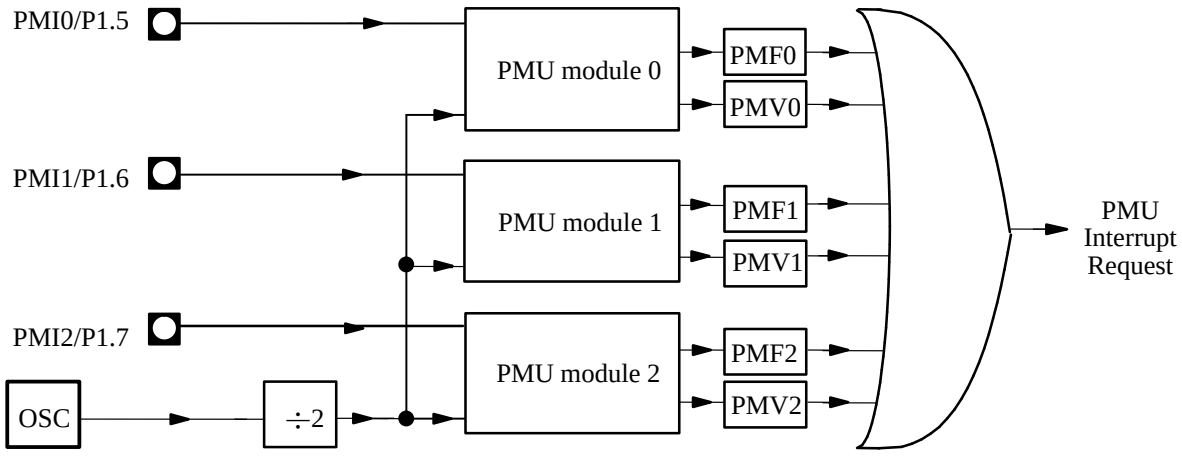


Figure 5.1. PMU block diagram

The PMU module structure is detailed in Figure 5.2. Each module features its own 8–bit Pulse Measurement prescaler (PMSCALn) which allows to adapt the PMU time base to the sensor. If the PMSCALn value is well chosen, the PMPERn value will be comprised between 128 and 255. Using the TSC80251A1 at its nominal speed, the prescaler then allows to achieve a measurement accuracy better than 1% while managing wave periods ranging from 20 µs to 1 ms.
The PWM ratio is simply obtained by dividing the 8–bit PMU width value (PMWIDn) by the 8–bit PMU Period value (PMPERn). As shown on Figure 5.3. , the Timer is set to zero at the beginning of one measurement, hence the errors on the PMPERn value and on the PMWIDn value are both negative (+0/–1 LSB). However, due to the division, the maximum relative error on the PWM ratio then will be +/–1 LSB.

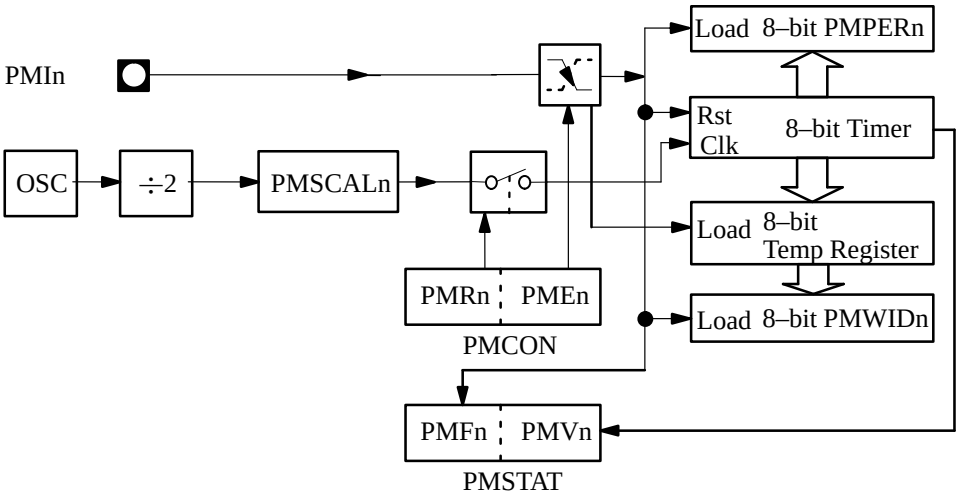


Figure 5.2. PMU module n (n = 0, 1, 2)

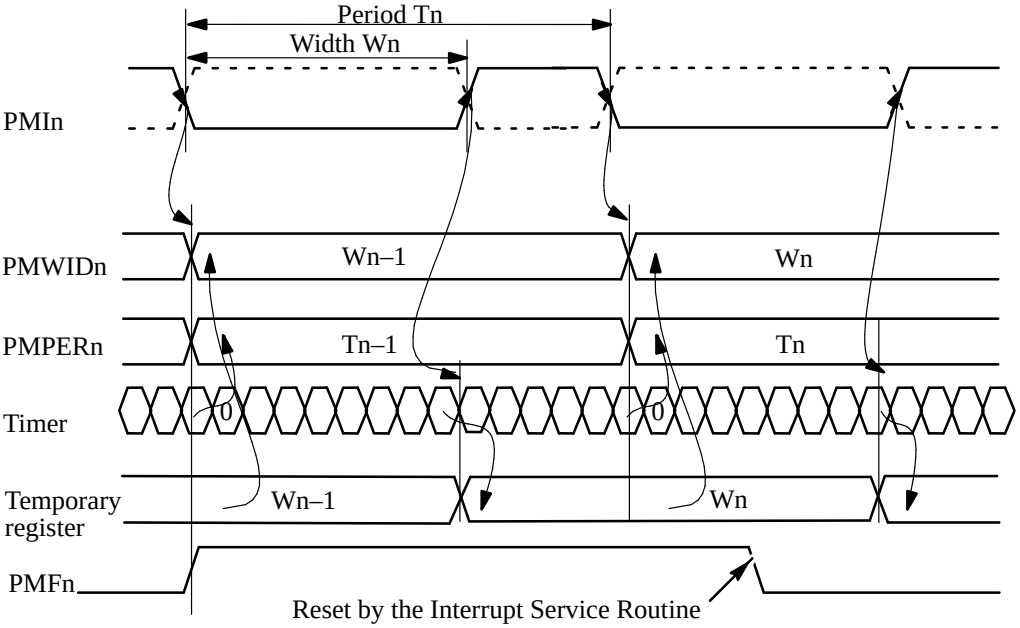


Figure 5.3. PMU measurement

All the status information regarding each module are gathered in the Pulse Measurement Status register (PMSTAT, See Figure 5.12.). When an overflow occurs in one PMU, its PMSCALn value must be increased to slow down the PMU time base until the measured period is less than 256 PMU time base clock cycles.
The Pulse Measurement Control register (PMCON, See Figure 5.5.) allows to enable or disable each PMU module operation through the Pulse Measurement Run control bits (PMRn, n = 0, 1, 2). When PMUn is stopped, its Timer is disabled and its PMPERn and PMWIDn registers are frozen. When PMUn is running, its PMPERn and PMWIDn registers are periodically updated. Hence, in order to get a consistent measurement from PMUn (i.e. PMPERn and PMWIDn values relating to the same period), its flags must be reset by software before any measurement and its measurement must be read as soon as possible after completion (i.e. when PMFn is set and before the end of the next period). When PMUn overflows, it should be stopped before resetting its flag to prevent a false measurement update if the measurement is not yet completed.
The PMCON register also allows to define the input polarity for each PMU through the Pulse Measurement Edge select bits (PMEn). The width measurement is performed either on the low level or the high level state as shown on Figure 5.4.

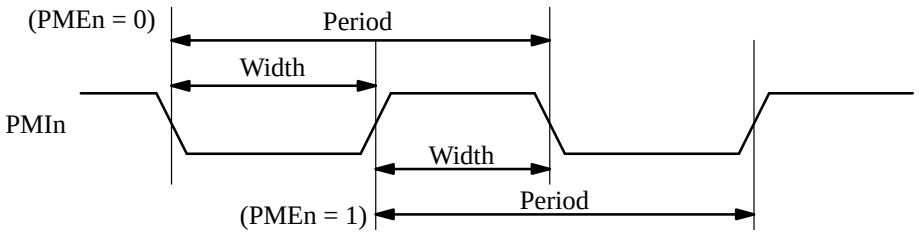


Figure 5.4. Pulse measurement polarity

5.3. Registers

PMCON (0ADh)
Pulse Measurement Control register

–	PME2	PME1	PME0	–	PMR2	PMR1	PMR0
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	PME2	**Pulse Measurement 2 edge select bit** Clear this bit to start PMU module n (n = 2) on falling edge. Set this bit to start PMU module n (n = 2) on rising edge**.**
5	PME1	**Pulse Measurement 1 edge select bit** Clear this bit to start PMU module n (n = 1) on falling edge. Set this bit to start PMU module n (n = 1) on rising edge**.**
4	PME0	**Pulse Measurement 0 edge select bit** Clear this bit to start PMU module n (n = 0) on falling edge. Set this bit to start PMU module n (n = 0) on rising edge**.**
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	PMR2	**Pulse Measurement 2 run control bit** Clear this bit to stop PMU module n (n = 2). Set this bit to start PMU module n (n = 2)**.**
1	PMR1	**Pulse Measurement 1 run control bit** Clear this bit to stop PMU module n (n = 1). Set this bit to start PMU module n (n = 1)**.**
0	PMR0	**Pulse Measurement 0 run control bit** Clear this bit to stop PMU module n (n = 0). Set this bit to start PMU module n (n = 0)**.**

Reset Value = X000 X000B

Figure 5.5. PMCON register

PMPER0 (0A2h)
Pulse Measurement Period register 0 (8–bit read only)

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

Reset Value = X000 X000B

Figure 5.6. PMPER0 register

PMPER1 (0A4h)
Pulse Measurement Period register 1 (8–bit read only)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.7. PMPER1 register

PMPER2 (0A6h)
Pulse Measurement Period register 2 (8–bit read only)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.8. PMPER2 register

PMSCAL0 (0AAh)
Pulse Measurement Prescaler register (8–bit)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.9. PMSCAL0 register

PMSCAL1 (0ABh)
Pulse Measurement Prescaler register (8–bit)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.10. PMSCAL1 register

PMSCAL2 (0ACh)
Pulse Measurement Prescaler register (8–bit)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.11. PMSCAL2 register

PMSTAT (0AEh)
Pulse Measurement Status register

–	PMV2	PMV1	PMV0	–	PMF2	PMF1	PMF0
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	PMV2	**PMU Overflow flag** Set by hardware when an overflow of the Counter has occured during the pulse measurement. Must be cleared by software.
5	PMV1	**PMU Overflow flag** Set by hardware when an overflow of the Counter has occured during the pulse measurement. Must be cleared by software.
4	PMV0	**PMU Overflow flag** Set by hardware when an overflow of the Counter has occured during the pulse measurement. Must be cleared by software.
3	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	PMF2	**Pulse Measurement flag** Cleared by hardware when PMU module 2 is stopped. Set by hardware when PMU module 2 detects a transition. Must be cleared by software to allow a new measurement.
1	PMF1	**Pulse Measurement flag** Cleared by hardware when PMU module 1 is stopped. Set by hardware when PMU module 1 detects a transition. Must be cleared by software to allow a new measurement.
0	PMF0	**Pulse Measurement flag** Cleared by hardware when PMU module 0 is stopped. Set by hardware when PMU module 0 detects a transition. Must be cleared by software to allow a new measurement.

Reset Value = X000 X000B

Figure 5.12. PMSTAT register

PMU (09Fh)
Pulse Measurement Unit Mode Control register

–	–	–	–	–	–	–	PMU.0
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	–	**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	PMMOD	**Pulse Measurement Unit** Must be set to one before any PMU configuration, otherwise the TSC80C251A1 behavior is unpredictable.

Reset Value = XXXX XXX0B

Figure 5.13. PMU register

PMWID0 (0A3h)
Pulse Measurement Width register (8–bit read only)

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

Reset Value = XXXX XXX0B

Figure 5.14. PMWID0 register

PMWID1 (0A5h)
Pulse Measurement Width register (8–bit read only)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.15. PMWID1 register

PMWID2 (0A7h)
Pulse Measurement Width register (8–bit, read only)

7	6	5	4	3	2	1	0

Reset Value = XXXX XXXXB

Figure 5.16. PMWID2 register

Event and Waveform Controller

6.1. Introduction

This chapter describes the Event and Waveform Controller (EWC) which is a superset of the Programmable Counter Array (PCA) found in some 80C51 microcontrollers. This is an on–chip peripheral of the TSC80251A1 which performs a variety of timing and counting operations, including Pulse Width Modulation (PWM).

The EWC can be configured in two modes:

- PCA
- Enhanced PCA (EPCA)

The PCA mode has up to five Compare/Capture modules using the same time base and event Counter. The EPCA mode has the Compare/Capture modules using their own time base and event Counter. The EWC also provides the capability for a software Watchdog Timer (WDT).

6.2. Features

- Compatible with PCA: Programmable Counter Array (PCA mode)
- Enhanced PCA (EPCA mode)
- Programmable Counter mode with 8–bit parallel output on Port 1 (External Counter mode)
- Five 16–bit Counter
- Five 16–bit Compare/Capture modules
- The last module can also be programmed as a Watchdog Timer (WDT)
- Each module may use up to seven clock sources:
  - 1/12 of the clock frequency
  - 1/4 of the clock frequency
  - Timer 0 overflow (Modes 1, 2 and 3)
  - External input on ECI (P1.2)
  - $F_{OSC}/2$ (EPCA mode)
  - Timer 1 overflow (EPCA mode)
  - Baud Rate Generator (EPCA mode)
- Each module can be programmed in any of the following modes:
  - Rising and/or falling edge Capture
  - Software Timer
  - High-speed Output
  - Pulse Width Modulation (PWM)

6.3. PCA Mode

6.3.1. Timers/Counters

Figure 6.2. depicts the basic logic of the Timer/Counter portion of the PCA. The CH/CL special function register pair operates as a 16–bit Timer/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.

CPS1 and CPS0 bits in CMOD register select one of four signals as the input to the Timer/Counter (See Figure 6.2.):

- F_{OSC} /12
  Provides a clock pulse at S5P2 of every peripheral cycle. With F_{OSC} = 16 MHz, the Timer/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 Timer/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 Timer/Counter on, if the output of the NAND gate (See Figure 6.2.) equals logic 1. The PCA Timer/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.

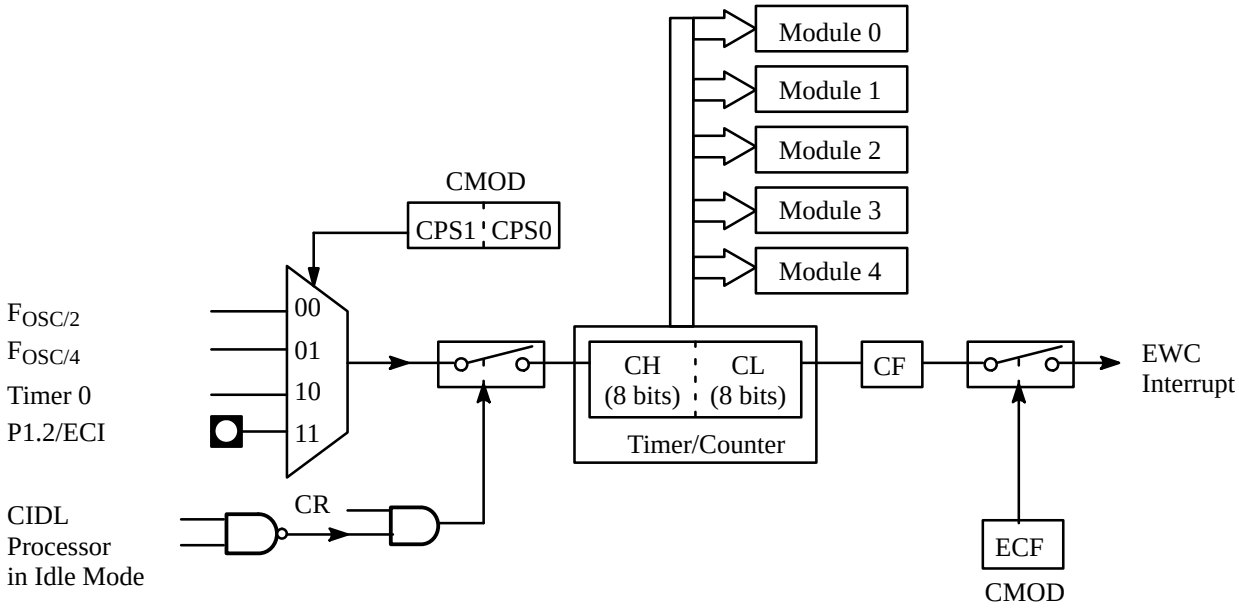


Figure 6.1. EWC Timer/Counter in PCA mode

6.3.2. Compare/Capture Modules

Each Compare/Capture module is made up of a Compare/Capture register pair (CHx/CLx; x = 0, 1, 2, 3, 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 Counter the duty cycle of the output waveform. The logical configuration of a Compare/Capture module controls 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 Timer/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 Timer/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 Timer/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 Timer/Counter.
- Load the comparison value into the module's Compare/Capture register pair.
- Set the PCA Timer/Counter run Counter bit.
- After a match causes an interrupt, clear the module's Compare/Capture flag.

- No operation

Bit combinations programmed into a Compare/Capture module's mode register (CCAPMx) determine the operation mode. Figure 6.10. provides bit definition and Table 6.3. 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 Timer/Counter, runs. The Timer/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 6.1. PCA module modes

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 (CCAPM4 only) (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; x = 0, 1, 2, 3, 4.
3. For the PCA WDT mode, set also WDTE bit in CMOD register to enable the reset output signal.

6.3.2.1. 16-bit Capture Mode

The Capture mode (See Figure 6.16.) 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 CEXO 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 Timer/Counter value. This records the time at which an external event is detected, with a resolution equal to the Timer/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 6.3. 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 Timer/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, a 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.

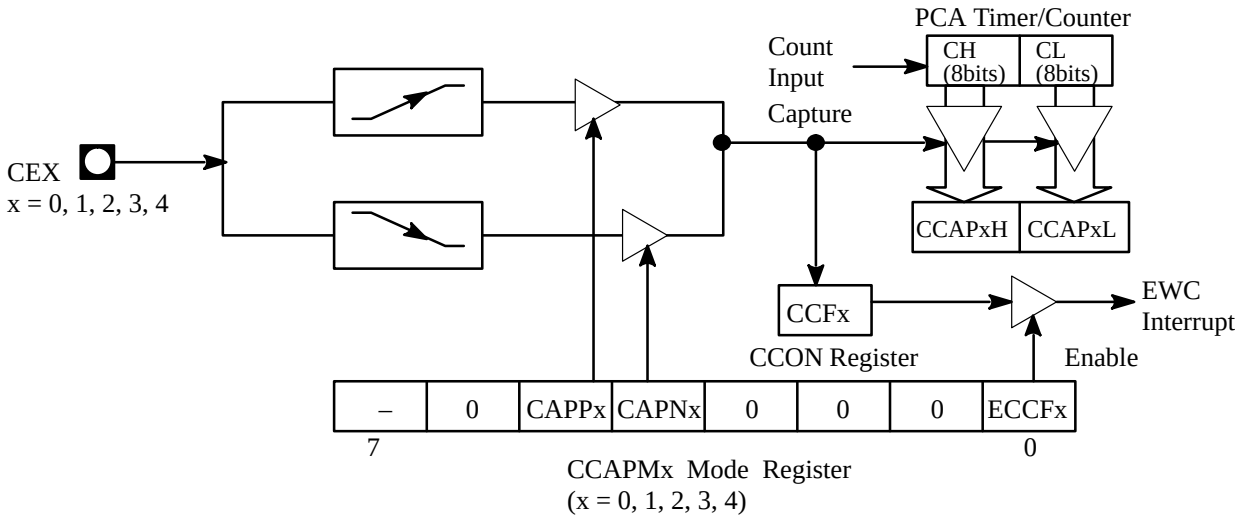


Figure 6.2. PCA 16–bit Capture Mode

6.3.2.2. 16–bit Software Timer Mode

To program a Compare/Capture module for the 16–bit software Timer mode (See Figure 6.3.), set the ECOMx and MATx bits in the module's CCAPMx register. Table 6.3. lists the bit combinations for selecting module modes.

A match between the PCA Timer/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).

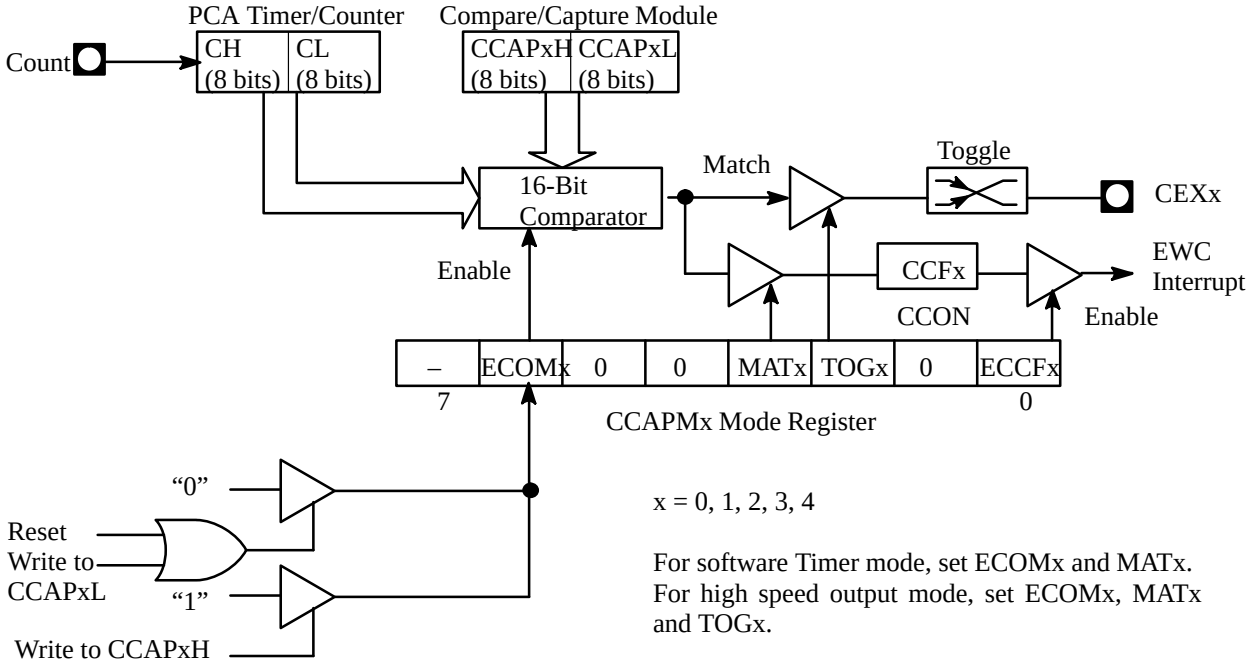


Figure 6.3. PCA Software Timer and High–Speed Output Modes

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 re–enabling the Compare function.

6.3.2.3. High-Speed Output Mode

The high–speed output mode (See Figure 6.3.) 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 6.3. lists the bit combinations for selecting module modes. A match between the PCA Timer/Counter and the Compare/Capture registers (CCAPxH/CCAPxL) toggles the CEXx pin and sets the module's Compare/Capture flag (CCFx in

CCON register). 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.

6.3.2.4. 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 Timer/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 6.18.). 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 6.3. lists the bit combinations for selecting module modes.
- Select the desired input for the PCA Timer/Counter by programming CPS0 and CPS1 bits in CMOD register (See Figure 6.15.).
- Enter a 16–bit comparison value in the Compare/Capture registers (CCAP4H/CCAP4L).
- Enter a 16–bit initial value in the PCA Timer/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 Timer/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 Timer/Counter value so a match never occurs.
  - Disable the module 4 reset output signal by clearing WDTE bit before a match occurs, then later re–enable it.

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.

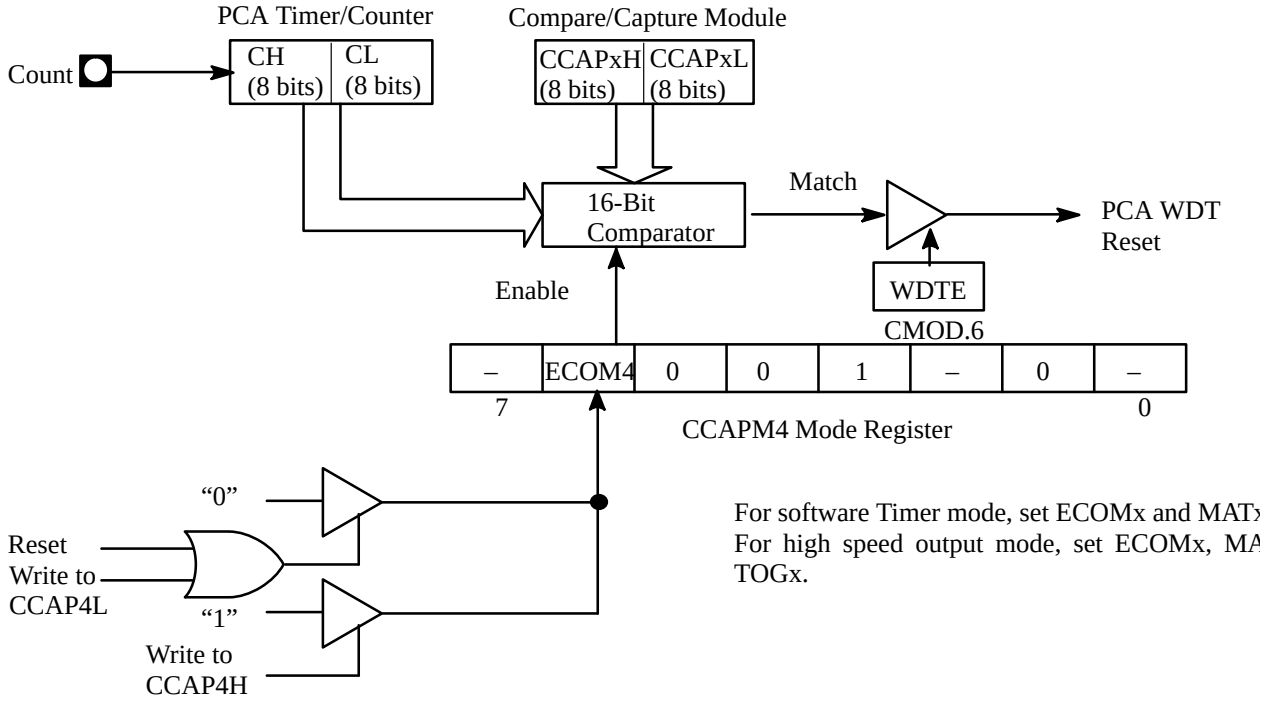


Figure 6.4. PCA Watchdog Timer mode

6.3.2.5. Pulse Width Modulator 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 6.20.). 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 6.20. , 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 6.3. lists the bit combinations for selecting module modes.
- Select the desired input for the PCA Timer/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 Timer/Counter run Counter bit (CR in CCON register) to start the PCA Timer/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 Timer/Counter input signal divided by 256. The highest frequency occurs when the $F_{OSC}/4$ input is selected for the PCA Timer/Counter. For F_{OSC} = 16 MHz, this is 15.6 KHz.

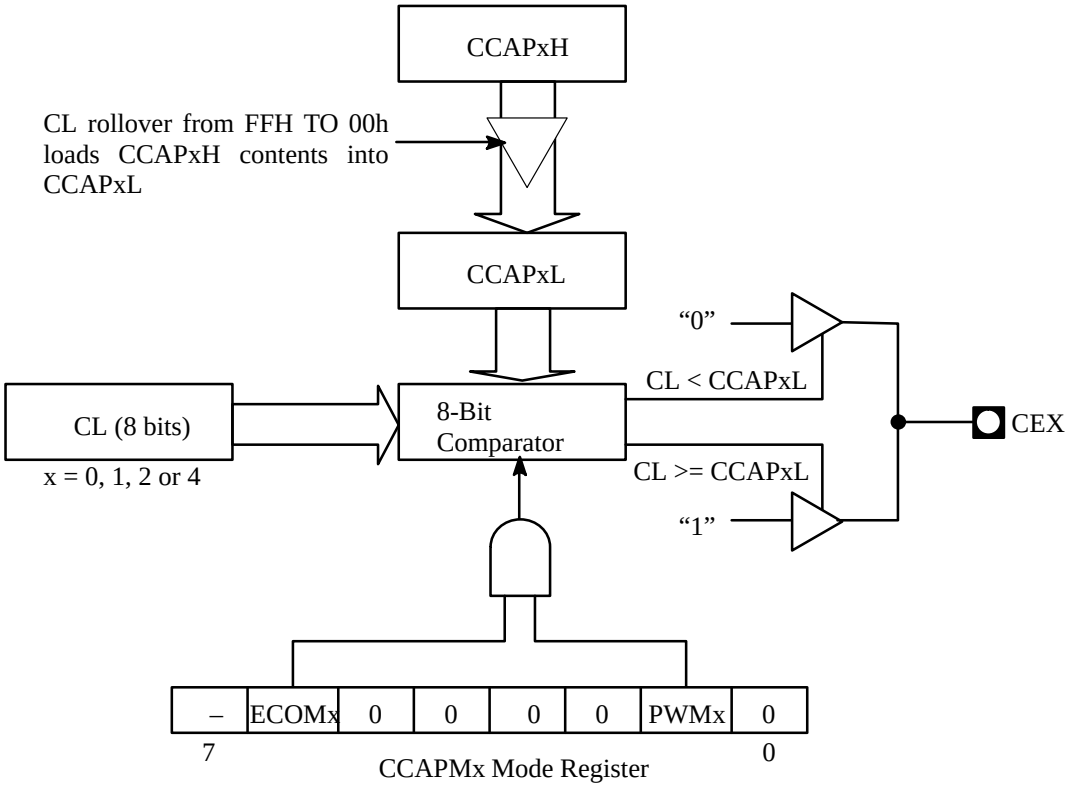


Figure 6.5. PWM mode

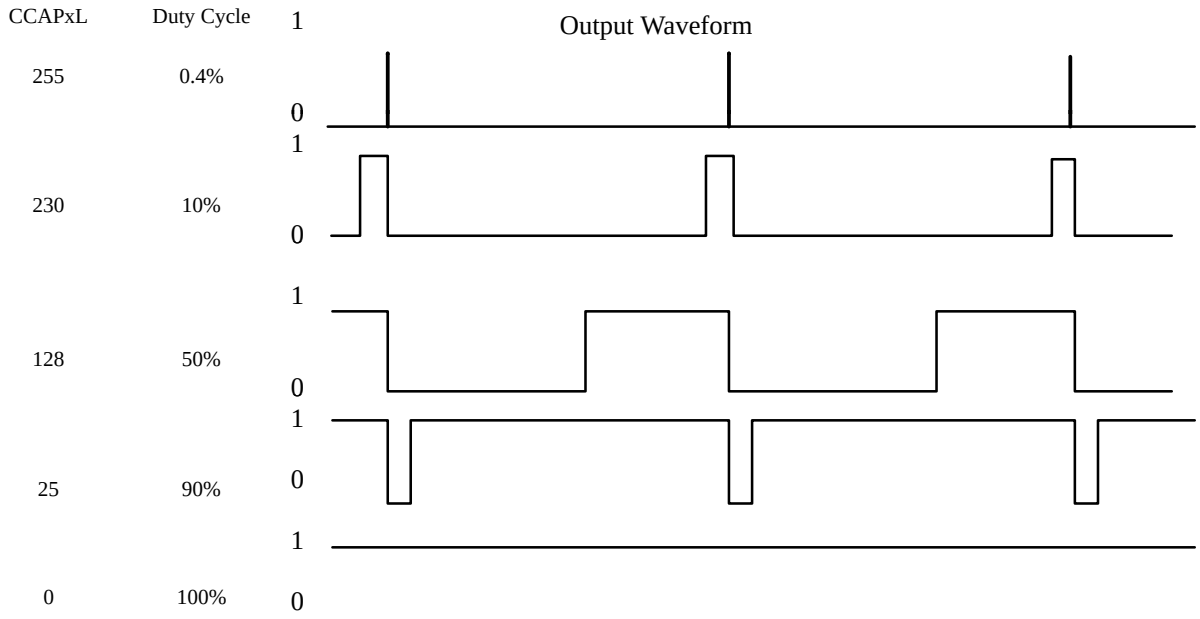


Figure 6.6. PWM variable duty cycle

6.4. Enhanced PCA mode

The Enhanced PCA mode (EPCA) provides all the PCA functionalities with additional features. It has the five Compare/Capture modules using their own EPCA Timer/Counter. One Timer/Counter and its Capture/Compare module form an EPCA unit. These five EPCA units may be linked to form a Time Base Array (TBA).

The EPCA mode is enabled by EPCA bit in CRC register. After reset, EPCA mode is disabled and the EWC is configured in PCA mode.

Please notice that the external Counter mode (See Figure 6.42.) takes precedence over the EPCA mode and should be disabled to have the EPCA working.

6.4.1. Timers/Counters

EPCA mode features five identical Timers/Counters instead of one in PCA mode. Each Timer/Counter is dedicated to one module. The structure of the EPCA unit is shown on Figure 6.3.

EPCA Timers/Counters are very similar to PCA Timer/Counter. The behavior of the Capture/Compare module is exactly the same as in PCA mode. All the differences are highlighted below:

- Independent Counter High and Counter Low registers (CHx and CLx; x = 0, 1, 2, 3, 4). In fact, in EPCA mode, CL is used as CL0 and CH is used as CH0.
- Independent Counter Run Counter bits (CRx; x = 0, 1, 2, 3, 4). These flags are gathered in the Counter Run Counter register (CRC). CR bit of CCON register is not used in EPCA mode.
- Independent Counter Idle Counter bits (CIDLx; x = 0, 1, 2, 3, 4). These flags are in the Counter Mode registers (CMODx; x = 1, 2, 3). CIDL bit of CMOD register is not used in EPCA mode.
- Up to seven different clock sources instead of four. They are selected independently for each Timer/Counter by the Count Pulse Select bits (CPx(2:0); x = 0, 1, 2, 3, 4). Three bits encode seven possible choice and one reserved. If CPx2 = 0, CPx(1:0) is performing the same selection as would CPS1:0 in PCA mode. The three new choices are provided by CPx2 set to one:
  - Fastest clock: $F_{OSC}/4$ is selected by CPx(1:0)=00.
  - Timer 1 overflow: Timer 1 is selected by CPx(1:0)=01.
  - Baud Rate Generator: it is selected by CPx(1:0)=11.
- Independent Counter Overflow flags (CFx; x = 0, 1, 2, 3, 4). These flags are gathered in the Counter Overflow Flag register (COF). CF bit of CCON register is not used in EPCA mode. When a flag is set, it produces an EWC interrupt request if the corresponding Enable Counter Overflow flag (ECFx; x = 0, 1, 2, 3, 4) is set. These flags are gathered in the Enable Counter Overflow Flag register (ECOF). ECF bit of CMOD register is not used in EPCA mode. They must be cleared by software.
- Four independent Compare/Capture interrupt request for CCFx (x = 1, 2, 3, 4). Each of them has its own interrupt vector (See “Interrupt System” chapter). Nevertheless CCF0 bit shares the general EWC interrupt request with the Counter Overflow flags (CFx; x = 0, 1, 2, 3, 4). All CCFx (x = 0, 1, 2, 3, 4) bits are gathered in CCON register as in PCA mode. The Enable CCFx interrupt bits (ECCFx; x = 0, 1, 2, 3, 4) are in the Compare/Capture Module mode registers (CCAPMx; x = 0, 1, 2, 3, 4) which works exactly the same as in PCA mode.

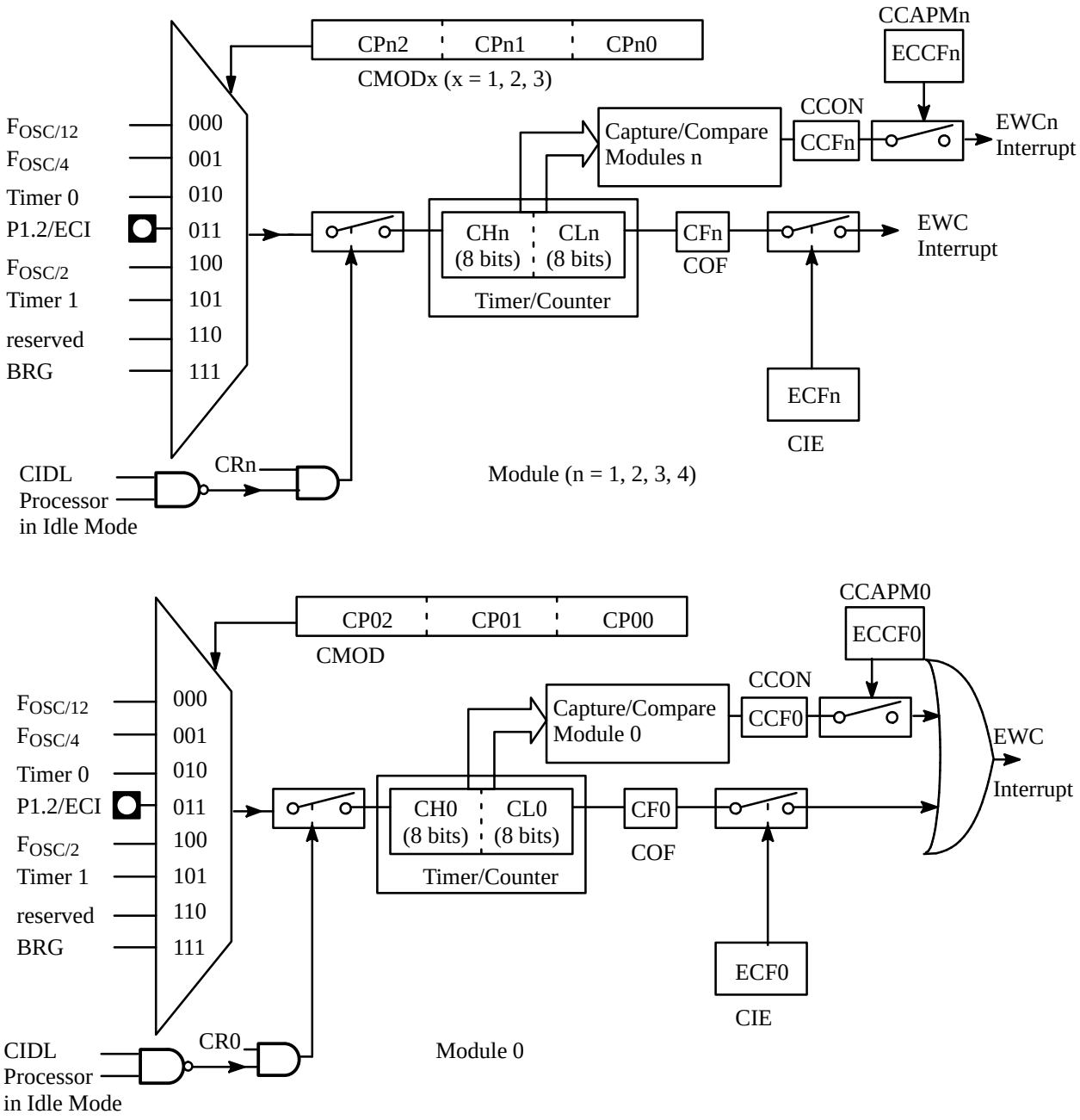


Figure 6.7. EWC Timer/Counter in EPCA mode

6.5. Registers

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

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

Reset Value = 0000 0000B

Figure 6.8. EWC CCAPxH registers (x = 0, 1, 2, 3, 4)

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

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

Reset Value = 0000 0000B

Figure 6.9. EWC CCAPxL registers (x = 0, 1, 2, 3, 4)

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

–	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 bit** Clear to disable the Compare function. Set to enable the Compare function. The Compare function is used to implement the software Timer, high-speed output, PWM and WDT modes.
5	CAPPx	**Capture Mode (Positive) 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) 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 bit** Set by hardware when a match of the PCA Timer/Counter with the Compare/Capture register sets the CCFx bit in the CCON register, flagging an interrupt. Must be cleared by software.
2	TOGx	**Toggle bit** The toggle mode is configured by setting ECOMx, MATx and TOGx bits. Set by hardware when a match of the PCA Timer/Counter with the Compare/Capture register toggles the CEXx pin. Must be cleared by software.
1	PWMx	**Pulse Width Modulation Mode bit** Set to configure the module for operation as an 8-bit Pulse Width Modulator with output waveform on CEXx pin. Must be cleared by software.
0	ECCFx	**Enable CCFx Interrupt bit** Set to enable Compare/Capture flag CCFx in CCON register to generate an interrupt request. Must be cleared by software.

Reset Value = X000 0000B

Figure 6.10. EWC CCAPMx (x = 0, 1, 2, 3, 4) registers

CCON (0D8h)
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 interrupt enable bit in CMOD register is set. CF can be set by hardware or software but must be cleared by software.
6	CR	**PCA Timer/Counter Run Control bit** Clear to turn the PCA Timer/Counter off. Set to turn the PCA Timer/Counter on.
5	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	CCF4	**PCA Module 4 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF4 interrupt enable bit in the corresponding CCAPM4 register is set. Must be cleared by software.
3	CCF3	**PCA Module 3 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF3 interrupt enable bit in the corresponding CCAPM3 register is set. Must be cleared by software.
2	CCF2	**PCA Module 2 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF2 interrupt enable bit in the corresponding CCAPM2 register is set. Must be cleared by software.
1	CCF1	**PCA Module 1 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF1 interrupt enable bit in the corresponding CCAPM1 register is set. Must be cleared by software.
0	CCF0	**PCA Module 0 Compare/Capture flag** Set by hardware when a match or capture occurs. This generates a PCA interrupt request if the ECCF0 interrupt enable bit in the corresponding CCAPM0 register is set. Must be cleared by software.

Reset Value = 00X0 0000B

Figure 6.11. EWC CCON register

CH0=CH (0F9h)
CH1 (0F4h)
CH2 (0F5h)
CH3 (0F6h)
CH4 (0F7h)
Counter x (x = 0, 1, 2, 3, 4) High registers

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

Reset Value = 0000 0000B

Figure 6.12. EWC CHx registers (x = 0, 1, 2, 3, 4)

CIE (0E3h)
Timer/Counter Interrupt Enable register

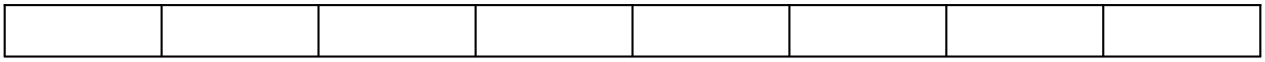
–	–	–	ECF4	ECF3	ECF2	ECF1	ECF0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

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	ECF4	**Enable Counter 4 Overflow bit** Clear to disable the interrupt generated by CF4 bit in COF register. Set to enable CF4 bit in COF register to generate an interrupt.
3	ECF3	**Enable Counter 3 Overflow bit** Clear to disable the interrupt generated by CF3 bit in COF register. Set to enable CF3 bit in COF register to generate an interrupt.
2	ECF2	**Enable Counter 2 Overflow bit** Clear to disable the interrupt generated by CF2 bit in COF register. Set to enable CF2 bit in COF register to generate an interrupt.
1	ECF1	**Enable Counter 1 Overflow bit** Clear to disable the interrupt generated by CF1 bit in COF register. Set to enable CF1 bit in COF register to generate an interrupt.
0	ECF0	**Enable Counter 0 Overflow bit** Clear to disable the interrupt generated by CF0 bit in COF register. Set to enable CF0 bit in COF register to generate an interrupt.

Reset Value = XXX0 0000B

Figure 6.13. EWC CIE register

CL0=CL (0E9h)
CL1 (0E4h)
CL2 (0E5h)
CL3 (0E6h)
CL4 (0E7h)
Counter x (x = 0, 1, 2, 3, 4) Low registers



7	6	5	4	3	2	1	0

Reset Value = 0000 0000B

Figure 6.14. EWC CLx registers (x = 0, 1, 2, 3, 4)

CMOD (0D9h)
Counter Mode register

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

Bit Number	Bit Mnemonic	Description
7	CIDL	**Counter Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running 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 0 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8)
1	CPS0	
0	ECF	**Enable Counter Overflow Interrupt bit** Clear to disable the interrupt generated by CF bit in CCON register. Set to enable CF bit in CCON register to generate an interrupt.

Figure 6.15. EWC CMOD register

CMOD1 (0DFh)
Counter 1 Mode register

CID1	CP12	CP11	CP10	CID0	CP02	CP01	CP00
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	CID1	**Timer/Counter 1 Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running when Idle mode is invoked.
6 5 4	CP12 CP11 CP10	**EWC Module 1 Count Pulse Select bits** CP12 CP11 CP10 Clock source 0 0 0 Internal clock, Fosc/12 0 0 1 Internal clock, Fosc/4 0 1 0 Timer 0 overflow 0 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8) 1 0 0 Internal clock, Fosc/2 1 0 1 Timer 1 overflow 1 1 0 Reserved 1 1 1 Baud Rate Generator overflow
3	CID0	**Timer/Counter 0 Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running when Idle mode is invoked.
2 1 0	CP02 CP01 CP00	**EWC Module 0 Count Pulse Select bits** CP02 CP01 CP00 Clock source 0 0 0 Internal clock, Fosc/12 0 0 1 Internal clock, Fosc/4 0 1 0 Timer 0 overflow 0 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8) 1 0 0 Internal clock, Fosc/2 1 0 1 Timer 1 overflow 1 1 0 Reserved 1 1 1 Baud Rate Generator overflow

Reset Value = 0000 0000B

Figure 6.16. EWC CMOD1 register

CMOD2 (0EFh)
Counter 2 Mode register

CID3	CP32	CP31	CP30	CID2	CP22	CP21	CP20
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	CID3	**Timer/Counter 3 Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running when Idle mode is invoked.
6 5 4	CP32 CP31 CP30	**EWC Module 3 Count Pulse Select bits** CP32 CP31 CP30 Clock source 0 0 0 Internal clock, Fosc/12 0 0 1 Internal clock, Fosc/4 0 1 0 Timer 0 overflow 0 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8) 1 0 0 Internal clock, Fosc/2 1 0 1 Timer 1 overflow 1 1 0 Reserved 1 1 1 Baud Rate Generator overflow
3	CID2	**Timer/Counter 2 Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running when Idle mode is invoked.
2 1 0	CP22 CP21 CP20	**EWC Module 2 Count Pulse Select bits** CP22 CP21 CP20 Clock source 0 0 0 Internal clock, Fosc/12 0 0 1 Internal clock, Fosc/4 0 1 0 Timer 0 overflow 0 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8) 1 0 0 Internal clock, Fosc/2 1 0 1 Timer 1 overflow 1 1 0 Reserved 1 1 1 Baud Rate Generator overflow

Reset Value = 0000 0000B

Figure 6.17. EWC CMOD2 register

CMOD3 (0FFh)
Counter 3 Mode register

–	–	–	–	CID4	CP42	CP41	CP40
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	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	CID4	**Timer/Counter 4 Idle Control bit** Clear to let the EWC running during Idle mode. Set to stop the EWC running when Idle mode is invoked.
2	CP42	**EWC Module 4 Count Pulse Select bits** CP42 CP41 CP40 Clock source 0 0 0 Internal clock, Fosc/12 0 0 1 Internal clock, Fosc/4 0 1 0 Timer 0 overflow 0 1 1 External clock at ECI/P1.2 pin (Max. Rate = Fosc/8) 1 0 0 Internal clock, Fosc/2 1 0 1 Timer 1 overflow 1 1 0 Reserved 1 1 1 Baud Rate Generator overflow
1	CP41	
0	CP40	

Reset Value = 0000 0000B

Figure 6.18. EWC CMOD3 register

COF (0E1h)
Timer/Counter Overflow Flag register

–	–	–	CF4	CF3	CF2	CF1	CF0
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	CF4	**EWC Timer/Counter 4 Overflow flag** Set by hardware when the Counter rolls over. CF4 flags an interrupt if ECF4 bit in ECF register is set. CF4 can be set by hardware or software but must be cleared by software
3	CF3	**EWC Timer/Counter 3 Overflow flag** Set by hardware when the Counter rolls over. CF3 flags an interrupt if ECF3 bit in ECF register is set. CF3 can be set by hardware or software but must be cleared by software.
2	CF2	**EWC Timer/Counter 2 Overflow flag** Set by hardware when the Counter rolls over. CF2 flags an interrupt if ECF2 bit in ECF register is set. CF2 can be set by hardware or software but must be cleared by software.
1	CF1	**EWC Timer/Counter 1 Overflow flag** Set by hardware when the Counter rolls over. CF1 flags an interrupt if ECF1 bit in ECF register is set. CF1 can be set by hardware or software but must be cleared by software.
0	CF0	**EWC Timer/Counter 0 Overflow flag** Set by hardware when the Counter rolls over. CF0 flags an interrupt if ECF0 bit in ECF register is set. CF0 can be set by hardware or software but must be cleared by software.

Reset Value = XXX0 0000B

Figure 6.19. EWC COF register

CRC (0E2h)
Counter Run Control register

STPM	–	MODE	CR4	CR3	CR2	CR1	CR0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	STPM	**Stop Mode bit** Clear to stop the Counter immediately upon a reset of the CR0 bit. Set to stop the Counter after the roll-over upon a reset of the CR0 bit.
6	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	MODE	**PCA/EPCA bit** Clear to configure the EWC in PCA mode (configuration per default, after a hardware reset). Set to configure the EWC in EPCA mode. In that case, CR bit in CCON register is don't care.
4	CR4	**EWC Timer/Counter 4 Run bit** If the MODE bit is cleared, setting this bit is irrelevant. Clear to turn the EWC Timer/Counter 4 off. Set to turn the EWC Timer/Counter 4 on.
3	CR3	**EWC Timer/Counter 3 Run bit** If the MODE bit is cleared, setting this bit is irrelevant. Clear to turn the EWC Timer/Counter 3 off. Set to turn the EWC Timer/Counter 3 on.
2	CR2	**EWC Timer/Counter 2 Run bit** If the MODE bit is cleared, setting this bit is irrelevant. Clear to turn the EWC Timer/Counter 2 off. Set to turn the EWC Timer/Counter 2 on.
1	CR1	**EWC Timer/Counter 1 Run bit** If the MODE bit is cleared, setting this bit is irrelevant. Clear to turn the EWC Timer/Counter 1 off. Set to turn the EWC Timer/Counter 1 on.
0	CR0	**EWC Timer/Counter 0 Run bit** If the MODE bit is cleared, setting this bit is irrelevant. Clear to turn the EWC Timer/Counter 0 off. Set to turn the EWC Timer/Counter 0 on.

Reset Value = 0000 0000B

Figure 6.20. EWC CRC register

8-bit Analog to Digital Converter

7.1. Introduction

This chapter describes the Analog to Digital Converter (ADC) and the relating SFR. This ADC is a key for digital processing of real world phenomena when electronic sensors providing a voltage analogy to physical phenomena are used.

7.2. Description

Figure 7.1. shows the ADC structure. It consists of a 4–input analog multiplexer followed by a sample and hold and an 8–bit successive approximation Analog/Digital (A/D) converter. It only requires an external Voltage Reference (Vref) with no other support component. This pin is next to the Analog ground pin (AVSS) to optimize its decoupling. The analog inputs (AN0 to AN3) are next to Vref which allows to easily shield all the analog pins using an AVSS guard ring.

AN0 to AN3 are alternate function of Port 1. Digital inputs on Port 1 can be read any time during an A/D conversion. However, special care should be taken in mixing analog and digital signals on these pins, which may cause cross–talk and degrades the ADC accuracy. Furthermore, if one of these pins is selected to perform a conversion, it will return a digital one when read while the conversion is in progress.

The acquisition is controlled by the ADC Control register (ADCON, See Figure 7.3.). The multiplexer selects one of the four possible analog inputs according to the number coded in two address bits (ADDR1 and ADDR0). Then the ADC Start bit (ADCS) allows to begin an acquisition by setting it to one. It remains set until the end of the conversion, then it automatically reset. This may takes up to 600 oscillator clock periods. This conversion time includes an acquisition time: this is the sum of the times required for the muxed analog signal to settle after the multiplexer command is selected and for the sample and hold procedure to complete.

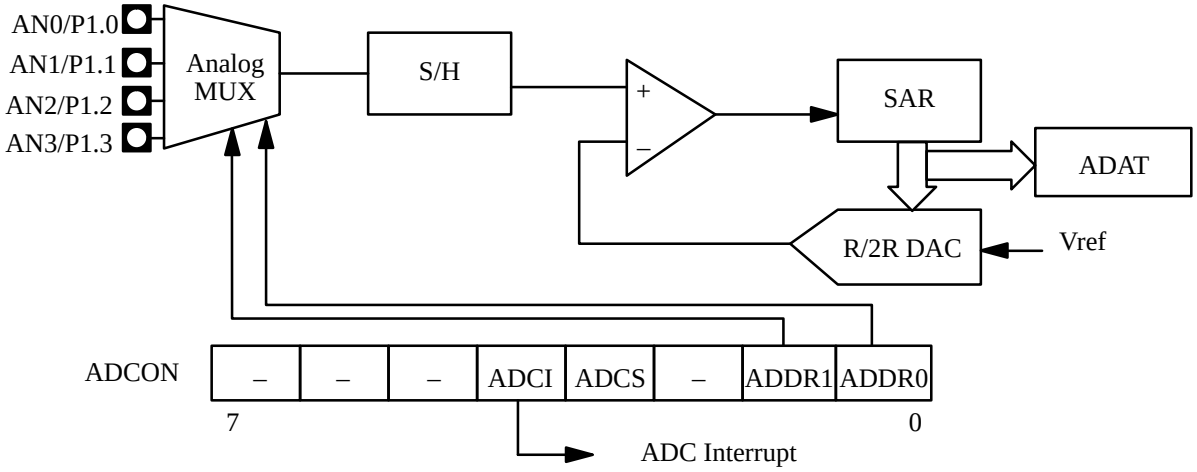


Figure 7.1. Analog Digital Converter structure

No new acquisition can begin while ADCS bit is set (i.e. a conversion is in progress) and this bit cannot be reset by software. When a new result is ready in the 8–bit ADC Data register (ADAT), when the conversion is completed, the ADC Interrupt bit (ADCI) is set and an ADC interrupt request is sent to the Interrupt System (see “Interrupt System” chapter). This bit must be reset by software when the contents of ADAT register can be disposed of (i.e. after it has been read by the interrupt service routine). Then a new acquisition can be requested (i.e. ADCS bit cannot be set while ADCI bit is set). ADCI bit and ADAT register are preserved in Idle mode and in Power–Down mode (see “Power Monitoring and Management”chapter), hence an already completed conversion is not lost. A conversion in progress will be aborted when entering the Idle mode, while it may not be aborted when entering in Power–Down mode. Therefore, it is recommended to wait for ADCS bit is zero before going into this mode, otherwise ADCI bit and ADAT register may change and a false interrupt may occur when this mode is exited through an interrupt. After an hardware reset, ADCON is set to its default value and the Analog to Digital Converter is inactive.

7.3. Registers

ADAT (0C6h)
Analog Data register (8–bit, read only)

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

Reset value = XXXX XXXXB

Figure 7.2. ADAT register

ADCON (0C5h)
ADC Control register

–	–	–	ADCI	ADCS	–	ADDR1	ADDR0
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	ADCI	**ADC Interrupt flag** Set by hardware when an A/D result is ready to be read. An interrupt is invoked if the ADC interrupt flag is enabled. Must be cleared by software.
3	ADCS	**ADC Start and Status bit** Cleared by hardware when the A/D conversion is completed, then ADCI is set. Set to start an A/D conversion.
2	–	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	ADDR1	**Input Channel Selection bits** ADDR1 ADDR0 Input pin selection 0 0 AN0 (P1.0) 0 1 AN1 (P1.1) 1 0 AN2 (P1.2) 1 1 AN3 (P1.3)
0	ADDR0	

Reset Value = 0000 0000B

Figure 7.3. ADCON register

Power Monitoring and Management

8.1. Introduction

These features can be used to supervise the Power Supply (VDD) and to start up properly the microcontroller when the power is up.

The power monitoring and management consist of the main features listed below and explained hereafter

- Power–On/Off reset
- Power–Fail detector
- 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).

8.2. Power–On/Off Reset

The Power–On reset ensures a proper starting of the microcontroller.
As long as VDD has not reached the V_{RST+} threshold, the microcontroller is left under reset and the oscillator is not enabled. As soon as VDD has reached V_{RST+}, the oscillator is enabled and starts up.
When the oscillator level on pin XTAL1 has reached the trigger level of the digital monostable, the reset counter is incremented by the oscillator. When the counter rolls off, it stops the reset system.

This system is not sensitive to the VDD rise time, because the oscillator is only enabled when the Power Supply (VDD) is stabilized over a reference level.
It is not either sensitive to the frequency, because the width of the reset pulse: t_{RST} is proportional to the crystal frequency. So this system guarantees a proper starting of the TSC80251A1 by protecting the reset against random conditions of VDD (See Figure 8.3.).

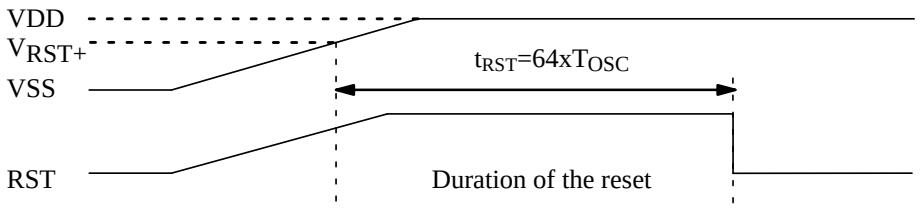


Figure 8.1. Behavior of the reset when the Power Supply is switched on

The Power–Off reset ensures a proper stopping of the TSC80251A1 when VDD fails or the Power Supply is switched off. If VDD reaches the V_{RST+} threshold, the microcontroller is maintained under reset until the Power Supply is completely off or VDD has reached again the V_{RST+} threshold.
This system avoids the TSC80251A1 running while the Power Supply is below the VDD specification.
It also guarantees a correct behavior of the microcontroller for the external components (See Figure 8.4.).

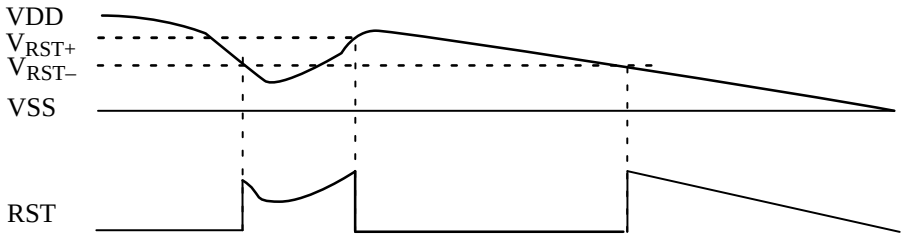


Figure 8.2. Behavior of the reset when the Power Supply is switched off

8.3. Power–Fail Detector

This mechanism is useful for applications which need to save system variables in a non–volatile memory. This feature monitors VDD and warns the TSC80251A1 by generating an early warning Power–Fail interrupt when VDD has dropped below the threshold level $V_{FAIL–}$. In that case Power–Fail Interrupt Enable bit (PFIE) in IE1 register has to be set and Power–Fail Disable bit (PFD) has to be cleared. Power–Fail Interrupt Enable bit (PFIE) should have the highest priority (see IS in paragraph 9).

If VDD drops below $V_{FAIL–}$ and then recovers and reaches V_{FAIL+} a new interrupt is generated and Power–Fail flag (PFF) is set in POWM register. The sequence waveform is shown in Figure 8.5.

To improve the noise immunity on VDD, glitches are filtered through a digital filter to allow only a persistent condition to trigger the internal reset. The filter consists of an 8–bit programmable counter incremented by the system clock as shown in Figure 8.6. The filtering window is programmable from 0 to 255 x $2T_{OSC}$ and is equal to 8 x $2T_{OSC}$ by default (after reset).

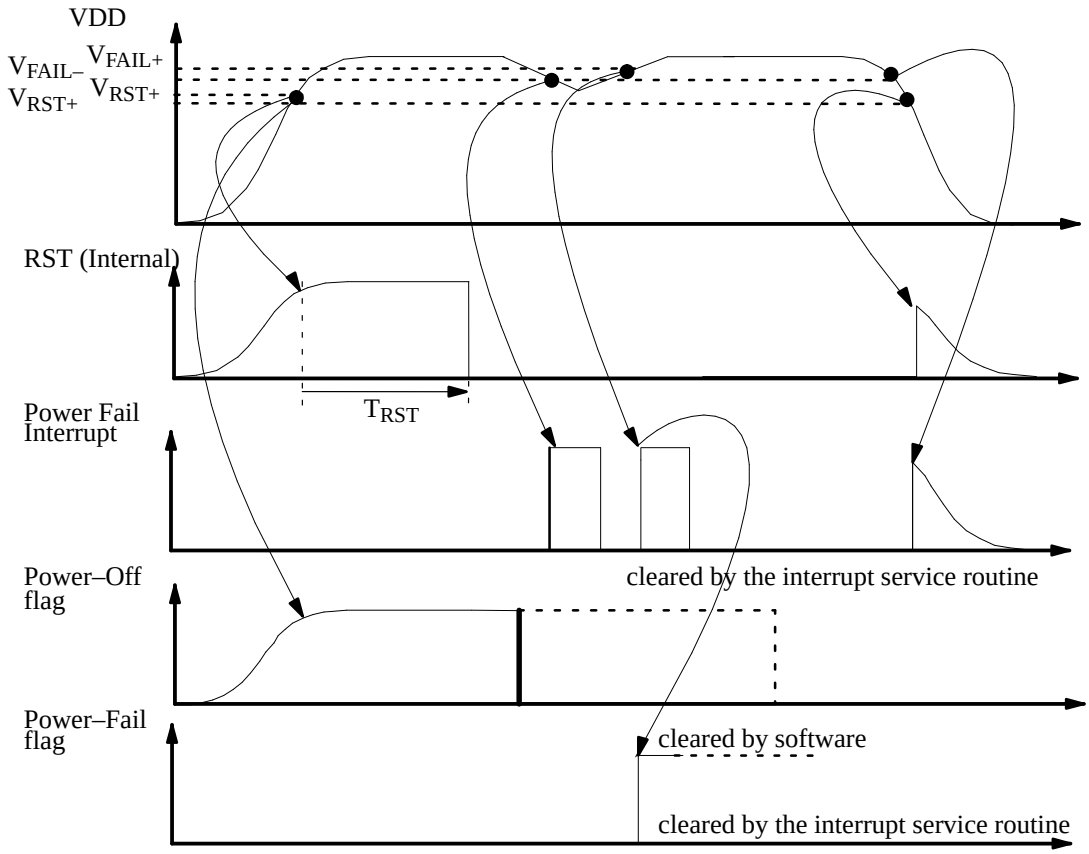


Figure 8.3. Power Management timings

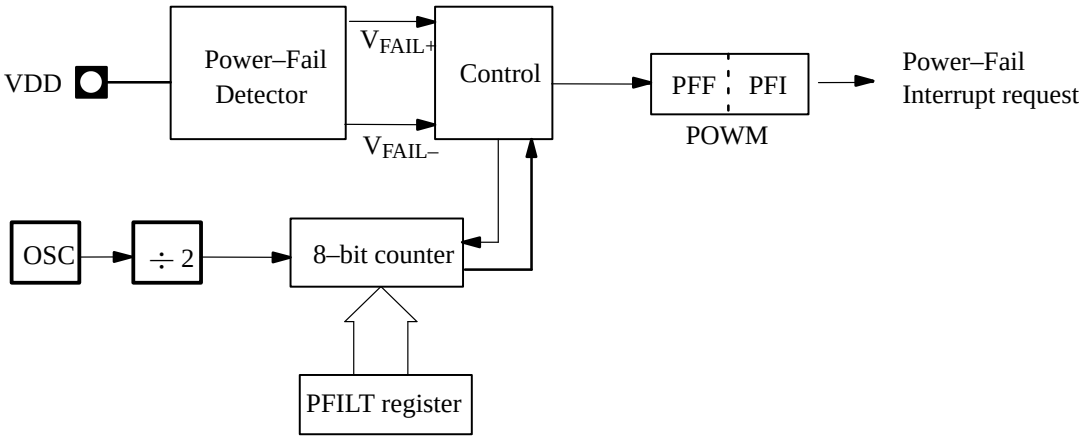


Figure 8.4. Block diagram of the digital filter

Figure 8.7. shows the principle of in the VDD filtering. A signal is considered as a glitch when its width is smaller than the time set–up in the 8–bit PFILT register. In this example filtering period is equal to 6 system clock periods and the A signal is considered as a glitch because its width is less

than 6 system clock periods. The B signal is not considered as a glitch and asserts the Power–Fail interrupt request.

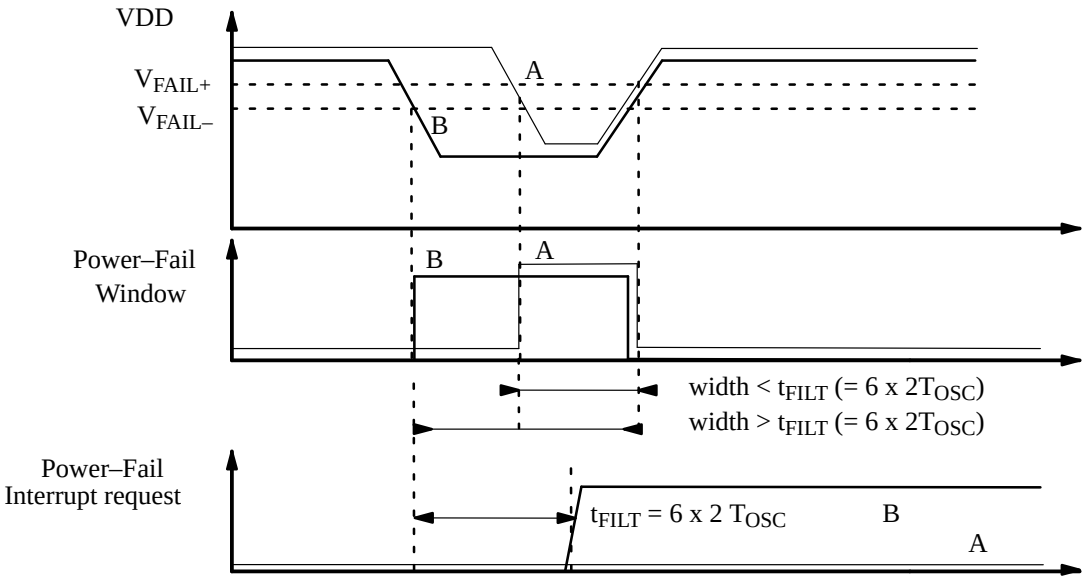


Figure 8.5. Waveforms of the VDD filtering

8.4. Power–Off Flag

The POF bit in PCON register is set to 1 when a hardware reset has been applied during the power is up. This reset is called ”Cold reset”. If a hardware reset is applied during the microcontroller is running, POF bit is not set. This reset is called ”Warm reset”. This flag allows to distinguish a cold from a warm reset and initialization. POF bit is useful in Power–Down mode when it is completed by a hardware reset. When used, this bit must be cleared by software after “Cold reset”.

8.5. Clock Prescaler

In order to optimize the 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 consumption optimizing. Figure 8.9. 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 to one CKSRC bit in POWM register:

$$F_{OSC} \quad \frac{F_{XTAL}}{2(CXRL \div 1)}$$

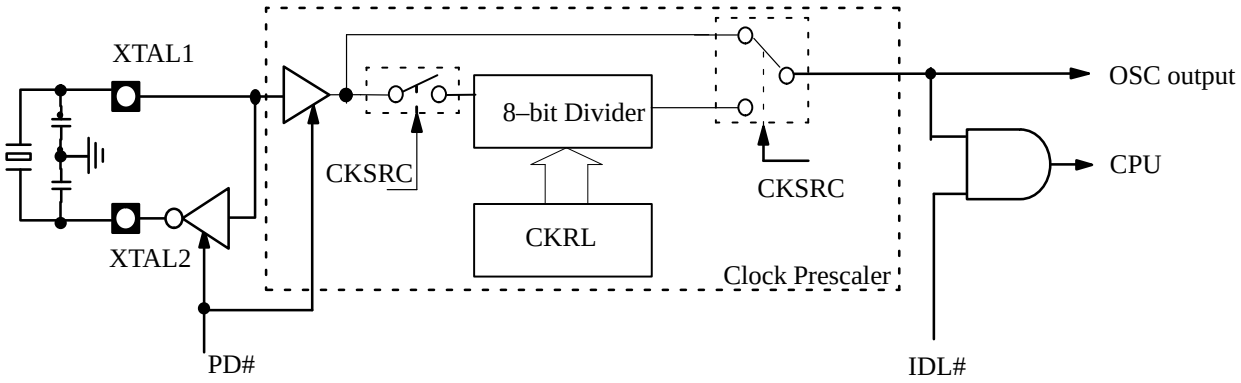


Figure 8.6. Block diagram of the on–chip oscillator

The on–chip oscillator is used to be symbolized by Figure 8.7. in all this datasheet.

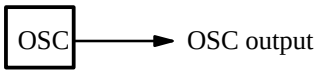


Figure 8.7. Symbolic of the on–chip oscillator

8.6. 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 8.9.). 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 and the Ports 0, 1, 2 and 3 pins are reading data (See Table 8.1.).
- External program memory: the ALE and PSEN# pins are pulled high; the Port 0 pins are floating and the pins of Ports 1, 2 and 3 are reading data (See Table 8.1.).

8.6.1. Entering Idle Mode

To enter Idle mode, set IDL bit in PCON register. The TSC80251A1 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 TSC80251A1 enters Power–Down mode.

8.6.2. Exiting Idle Mode

There are two ways to exit Idle mode:

- Generate an enabled interrupt. Hardware clears IDL bit in the 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 TSC80251A1 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.

Table 8.1. Pin conditions in various modes

Mode	Program Memory	ALE pin	PSEN# pin	Port 0 pin	Port 1 pin	Port 2 pin	Port 3 pin
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

8.7. Power–Down Mode

The Power–Down mode places the TSC80251A1 in a very low power state. Power–Down mode stops the oscillator and freezes all clock at known states (See Figure 8.9.). 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 and the Ports 0, 1, 2 and 3 pins are reading data (See Table 8.1.).
- External program memory: the ALE and PSEN# pins are pulled low; the Port 0 pins are floating and the pins of Ports 1, 2 and 3 are reading data (See Table 8.1.).

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

8.7.1. Entering Power–Down Mode

To enter Power–Down mode, set PD bit in PCON register. The TSC80251A1 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.

8.7.2. Exiting Power–Down Mode

Caution:
If VDD was reduced during the Power–Down mode, do not exit Power–Down 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.

- 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 and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the TSC80251A1 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.

8.8. Registers

PCON (87h)
Power Configuration register

SMOD1	SMODO	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. See Serial Port Control register (SCON).
5	RPD	**Recover for Idle/Power–Down bit** Clear to enable only the enable interrupt sources to exit from Idle or Power–Down mode. Set to permit to recover from Idle or Power–Down modes using external interrupt source. If the interrupt source is not enabled, the program simply continue at the address otherwise it jumps to interrupt service routine.
4	POF	**Power–Off flag** Set by hardware as VDD rises above 3 V to indicate that the Power has been off or VDD had fallen below 3 V and that on–chip volatile memory is indeterminated.
3	GF1	**General Purpose flag 1** One use is to indicate whether an interrupt occured during normal operation or during Idle mode.
2	GF0	**General Purpose flag 0** One use is to indicate whether 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 8.8. PCON register

PFILT (86h)
Power Filter register (8–bit)

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

Reset Value = 0000 1000B

Figure 8.9. PFILT register

POWM (8Fh)
Power Management register

CKSRC	–	–	–	RSTD	PFD	PFF	PFI
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 the CPU clock is the oscillator source divided by two. Set to enable the programmable clock. In that case the clock is divided by the value contained in CKRL register.
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	PFD	**Power-Fail Disable bit** Clear to enable the Power-Fail detector. Set to disable the Power-Fail detector.
1	PFF	**Power-Fail Flag bit** Cleared by hardware after a reset or when VDD falls from V_{FAIL+} to $V_{FAIL–}$. Set by hardware when VDD rises from $V_{FAIL–}$ to V_{FAIL+}. This bit may be cleared by software.
0	PFI	**Power-Fail Interrupt flag bit** Must be cleared by software. Set by hardware when VDD falls from V_{FAIL+} to $V_{FAIL–}$, or when VDD rises from $V_{FAIL–}$ to V_{FAIL+}.

Reset Value = 0000 0000B

Figure 8.10. POWM register

CKRL (8Eh)
Clock Reload register (8–bit)

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

Reset Value = 0000 1000B

Figure 8.11. CKRL register

Interrupt System

9.1. Introduction

The TSC80251A1, 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 TSC80251A1 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. Thirteen of the fourteen 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 TSC80251A1, 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 9.1. 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

Table 9.2. Interrupt System SFRs

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

The TSC80251A1 has one software interrupt: TRAP and thirteen peripheral interrupt sources: two external (INT0# and INT1#), one for Timer 0, one for Timer 1, one for Serial Port, one for Pulse Measurement Unit, five for Event and Waveform Controller, one for Analog to Digital Converter, one for Power–Fail detector.

Note:
NMI interrupt source is not implemented in this derivative.

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 9.1 and Figure 9.2).

Four 8–bit registers are used to establish the priority level of the sixteen sources: IPL0, IPH0, IPL1 and IPH1 (See Figure 9.3, Figure 9.4, Figure 9.5 and Figure 9.6).

9.2. Interrupt System Priorities

Each of the thirteen interrupt sources on the TSC80251A1 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 9.4. and Figure 9.5.) and one in the Interrupt Priority Low registers (IPL0 or IPL1, see Figure 9.6. and Figure 9.7.) This provides each interrupt source four possible priority levels select bits (See Table 9.3.).

Table 9.3. 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 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 9.4. *).*

Table 9.4. 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	–
Reserved	–	FF:003Bh	–
INT0#	3	FF:0003h	H if edge, S if level
Timer 0	4	FF:000Bh	H if edge, S if level
INT1#	5	FF:0013h	H if edge, S if level
Timer 1	6	FF:001Bh	H
Serial Port	7	FF:0023h	S
A/D converter	8	FF:002Bh	S
EWC0	9	FF:0033h	S
PMU	10	FF:0043h	S
EWC1	11	FF:004Bh	S
EWC2	12	FF:0053h	S
EWC3	13	FF:005Bh	S
EWC4	14	FF:0063h	S
Reserved	15	FF:006Bh	–
Reserved	16	FF:0073h	–
Power–Fail	17 Lowest Priority	FF:0083h	S

9.3. External Interrupts

External interrupts INT0# and INT1# (INTn#, n = 0, 1) pins may each be programmed to be level–triggered or edge–triggered, dependent upon bits IT0 and IT1 (ITn, n = 0, 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, 1) in IE0 register. Events on INTn# set the interrupt request flag IEn in TCON. 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.

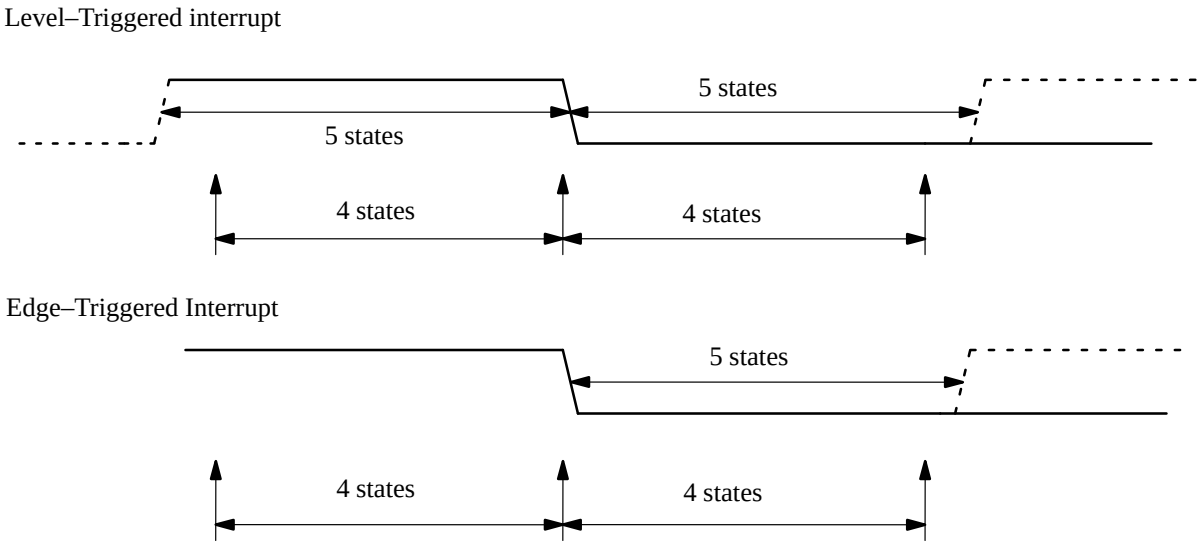


Figure 9.1. Minimum pulse timings.

9.4. Registers

IE0 (0A8h)
Interrupt Enable 0 register

EA	EC	EADC	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 6:0 in IE1 register. Set to enable all interrupts that are individually enabled by bits 6:0 in IE0 register and bits 6:0 in IE1 register.
6	EC	**Enable Counter Interrupt bit** Clear to disable EWC interrupt. Set to enable EWC interrupt.
5	EADC	**Enable Analog to Digital Converter Interrupt bit** Clear to disable ADC interrupt. Set to enable ADC interrupt.
4	ES	**Enable Serial Port Interrupt bit** Clear to disable Serial Port interrupt. Set to enable Serial Port interrupt.
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 9.2. IE0 register

IE1 (0B1h)
Interrupt Enable 1 register

PFIE	–	–	EC4	EC3	EC2	EC1	PMU
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	PFIE	**Power-Fail Interrupt Enable bit** Clear to disable the Power-Fail interrupt. Set to enable the Power-Fail interrupt.
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	EC4	**Enable Counter 4 Interrupt bit** Clear to disable the EWCn Counter 4 interrupt. Set to enable the EWCn Counter 4 interrupt.
3	EC3	**Enable Counter 3 Interrupt bit** Clear to disable the EWCn Counter 3 interrupt. Set to enable the EWCn Counter 3 interrupt.
2	EC2	**Enable Counter 2 Interrupt bit** Clear to disable the EWCn Counter 2 interrupt. Set to enable the EWCn Counter 2 interrupt.
1	EC1	**Enable Counter 1 Interrupt bit** Clear to disable the EWCn Counter 1 interrupt. Set to enable the EWCn Counter 1 interrupt.
0	EPMU	**Enable Pulse Measurement Unit Interrupt bit** Clear to disable the PMU interrupt. Set to enable the PMU interrupt.

Reset Value = 0000 0000B

Figure 9.3. IE1 register

IPH0 (0B7h)
Interrupt Priority High 0 register

–	IPHC	IPHADC	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	IPHADC	**ADC Interrupt Priority level most significant bit** IPHADC IPLADC 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 9.4. IPH0 register

IPH1 (0B1h)
Interrupt Priority High 1 register

IPHPF	–	–	IPHC4	IPHC3	IPHC2	IPHC1	IPHPMU
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	IPHPF	**Power–Fail Interrupt Priority level most significant bit** IPHPF IPLPF Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
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	IPHC4	**EWC Counter 4 Interrupt Priority level most significant bit** IPHEC4 IPLEC4 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
3	IPHC3	**EWC Counter 3 Interrupt Priority level most significant bit** IPHEC3 IPLEC3 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
2	IPHC2	**EWC Counter 2 Interrupt Priority level most significant bit** IPHEC2 IPLEC2 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
1	IPHC1	**EWC Counter 1 Interrupt Priority level most significant bit** IPHEC1 IPLEC1 Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority
0	IPHPMU	**PMU Interrupt 0 Priority level most significant bit** IPHPMU IPLPMU Priority level 0 0 0 Lowest priority 0 1 1 1 0 2 1 1 3 Highest priority

Reset Value = X000 0000B

Figure 9.5. IPH1 register

IPL0 (0B8h)
Interrupt Priority Low 0 register

–	IPLC	IPLADC	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 most significant bit.** Refer to IPHC for priority level.
5	IPLADC	**ADC Interrupt Priority level most significant bit.** Refer to IPHADC for priority level.
4	IPLS	**Serial Port Interrupt Priority level most significant bit.** Refer to IPHS for priority level.
3	IPLT1	**Timer 1 Interrupt Priority level most significant bit.** Refer to IPHT1 for priority level.
2	IPLX1	**External Interrupt 1 Priority level most significant bit.** Refer to IPHX1 for priority level.
1	IPLT0	**Timer 0 Interrupt Priority level most significant bit.** Refer to IPHT0 for priority level.
0	IPLX0	**External Interrupt 0 Priority level most significant bit.** Refer to IPHX0 for priority level.

Reset Value = X000 0000B

Figure 9.6. IPL0 register

IPL1 (0B2h)

Interrupt Priority Low 1 register

IPLPF	–	–	IPLC4	IPLC3	IPLC2	IPLC1	IPLPMU
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Number	Bit Mnemonic	Description
7	IPLPF	**Power–Fail Interrupt Priority level most significant bit.** Refer to IPHPF for priority level.
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	IPLC4	**EWC Counter 4 Interrupt Priority level most significant bit.** Refer to IPHEC4 for priority level.
3	IPLC3	**EWC Counter 3 Interrupt Priority level most significant bit.** Refer to IPHEC3 for priority level.
2	IPLC2	**EWC Counter 2 Interrupt Priority level most significant bit.** Refer to IPHEC2 for priority level.
1	IPLC1	**EWC Counter 1 Interrupt Priority level most significant bit.** Refer to IPHEC1 for priority level.
0	IPLPMU	**PMU Interrupt Priority level most significant bit.** Refer to IPHPMU for priority level.

Reset Value = X000 0000B

Figure 9.7. IPL1 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 +85°C VSS = 0 V VDD = 5 V ± 10 %	TA = –40 +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
$V_{FAIL}+$	VDD–Fail threshold on		4.2		V	
$V_{FAIL}-$	VDD–Fail threshold off		4.1		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.

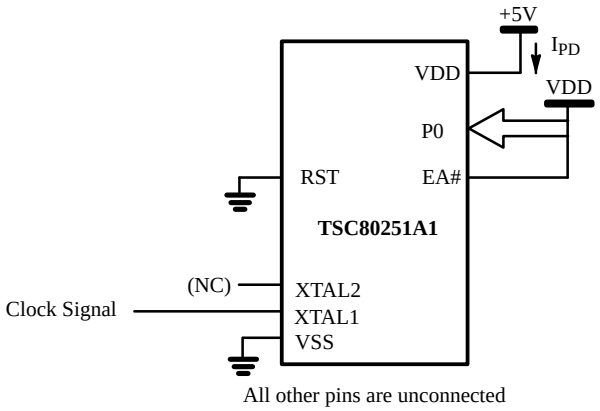


Figure 1.1. I_{PD} Test Condition, Power–Down mode

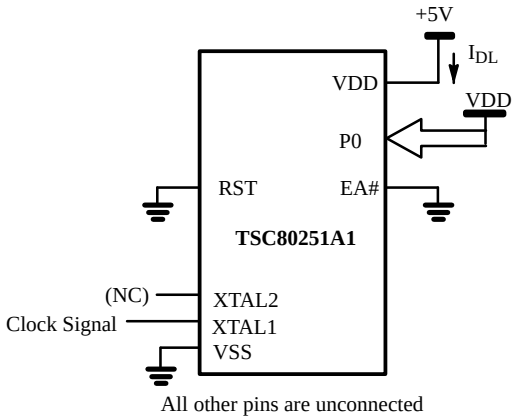


Figure 1.2. I_{DL} Test Condition, Idle mode

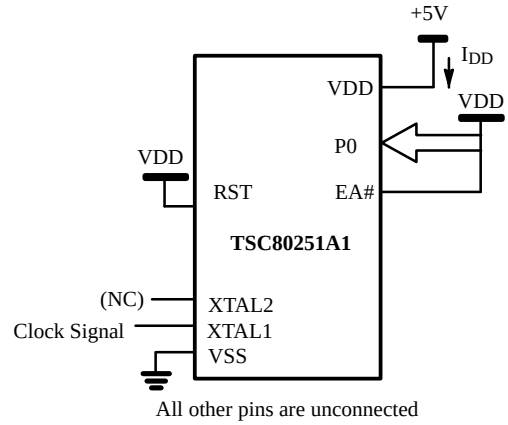


Figure 1.3. I_{DD} Test Condition, Active mode

AC characteristics

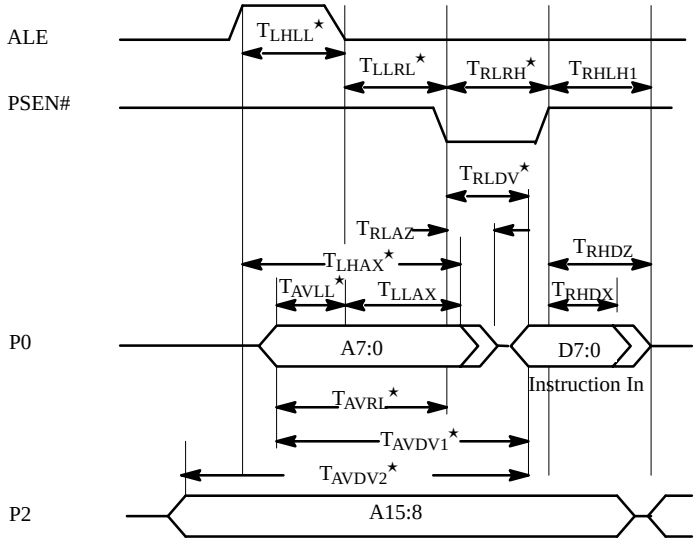
Table 2.1. AC characteristics (Capacitive Loading = 50 pF)

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		107		107		$2T_{OSC}$ - 60	ns

3

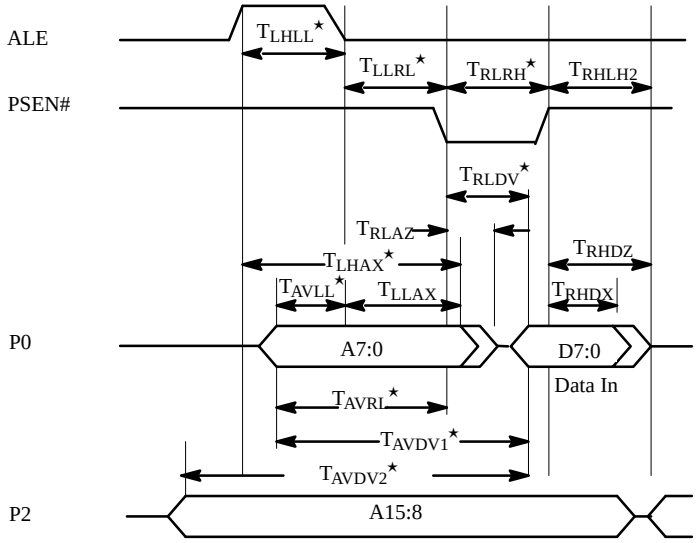
Symbol	Parameter	12 MHz		16 MHz		F_{OSC}		Units
		Min	Max	Min	Max	Min	Max	
T_{AVRL}	Address Valid to RD#/PSEN# Low	143		101		$2T_{OSC}$ - 24		ns (2)
T_{AVWL1}	Address (P0) Valid to WR# Low	143		101		$2T_{OSC}$ - 24		ns (2)
T_{AVWL2}	Address (P2) Valid to WR# Low	220		158		$3T_{OSC}$ - 30		ns (2)
T_{WHQX}	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.

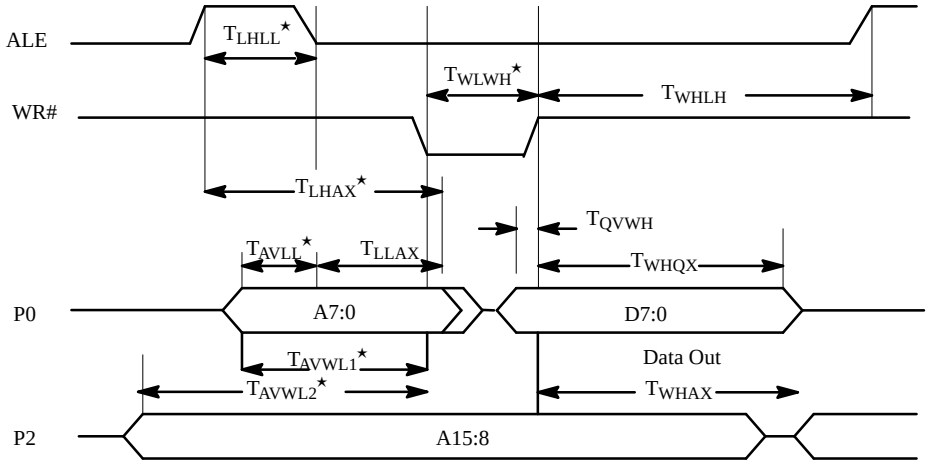
Figure 2.1. External Instruction Bus Cycle in non–page mode



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

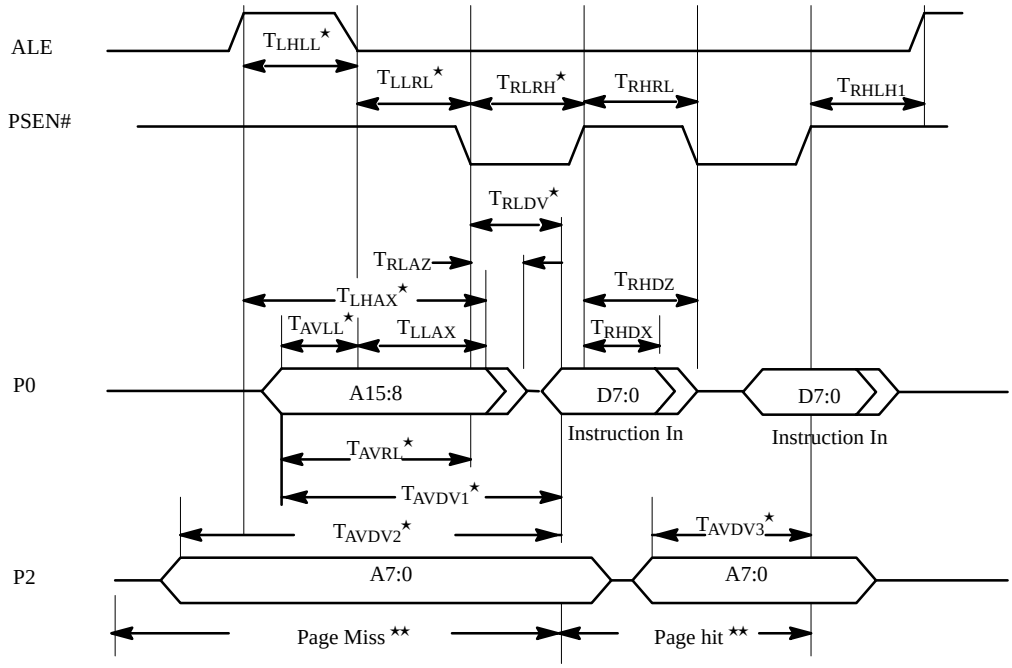
Figure 2.2. External Data Read Cycle in non–page mode

3



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

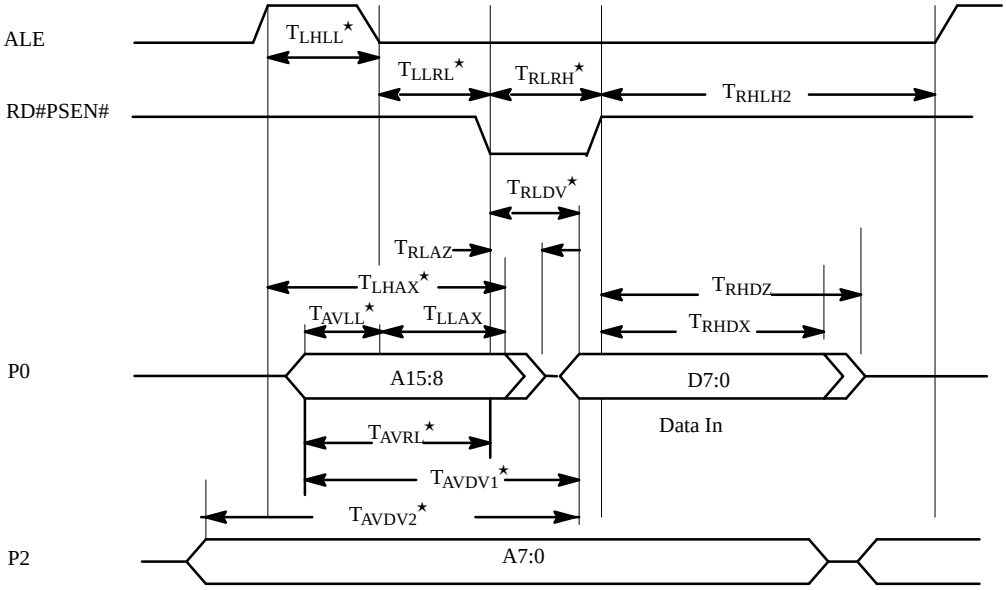
Figure 2.3. External Write Data Bus Cycle in non–page mode



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

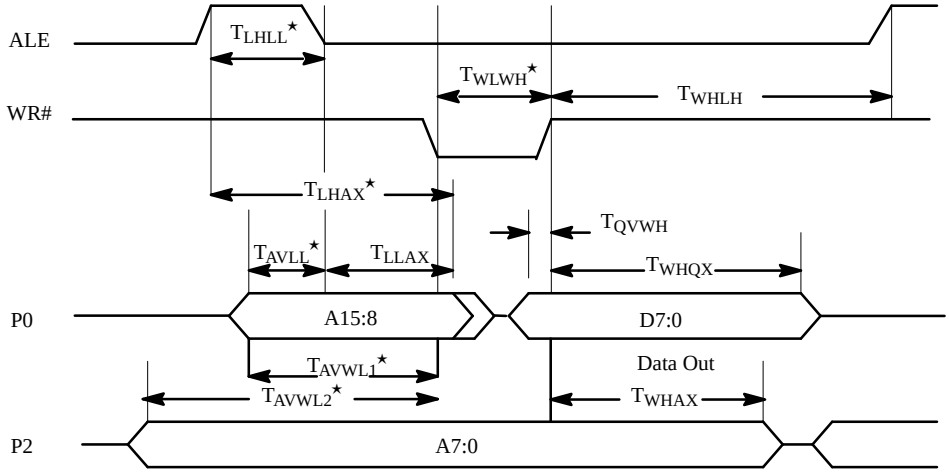
★★ A page hit (i.e., a code fetch to the same 256-byte “page” as the previous code fetch) requires one state ($2T_{OSC}$); a page miss requires two states ($4T_{OSC}$).

Figure 2.4. External Instruction Bus Cycle in page mode



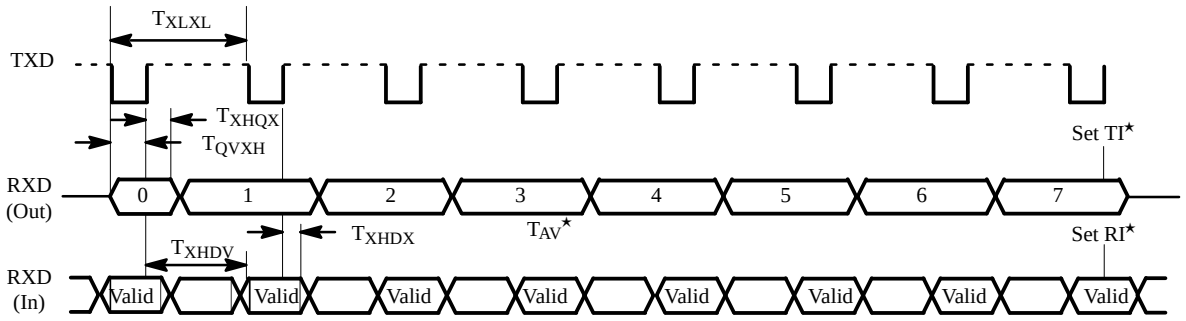
★ The value of this parameter depends on wait states. See the table of AC characteristics.

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



* 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

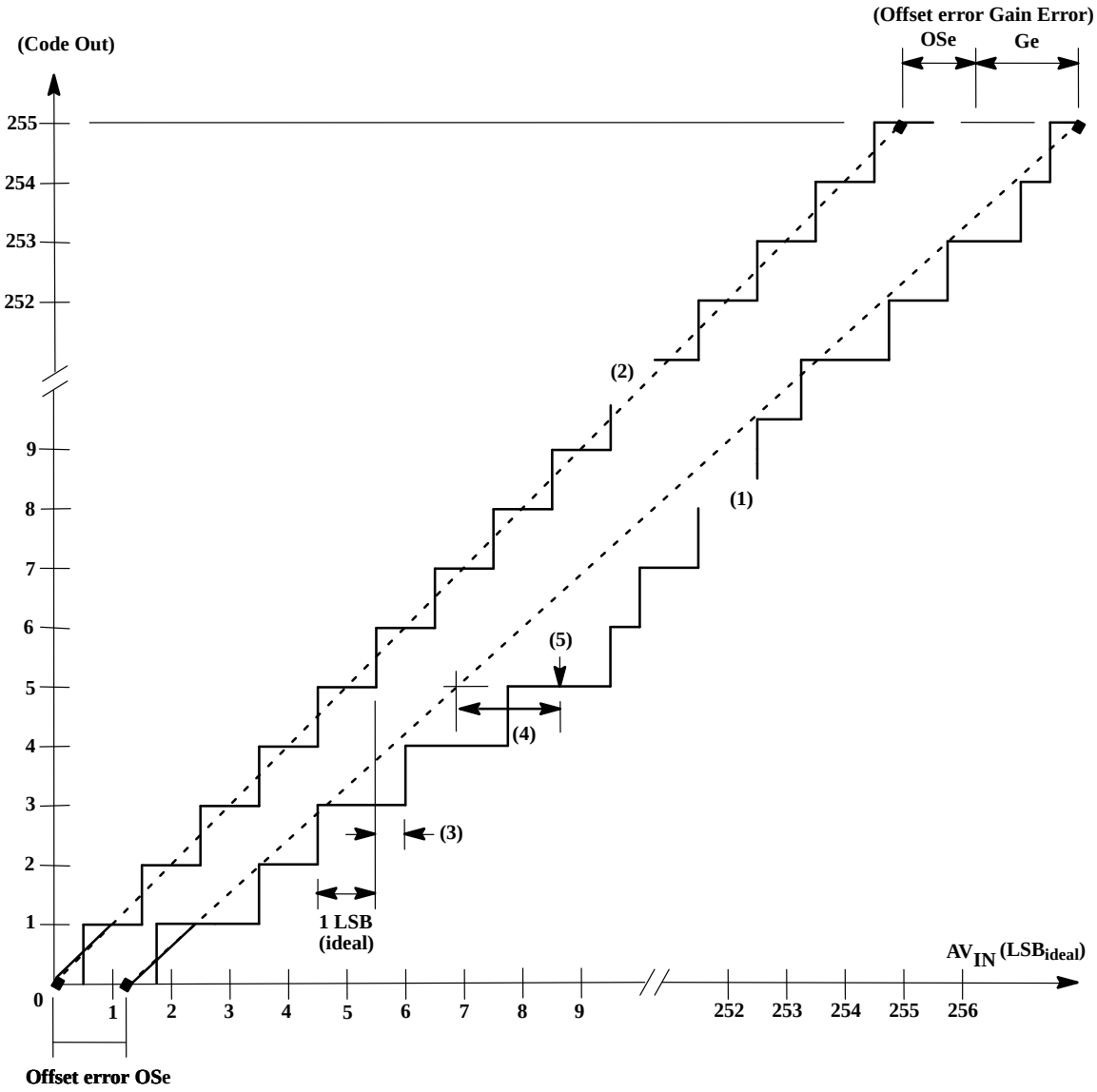
ADC characteristics

Table 3.1. A/D Converter electrical characteristics

Commercial	Industrial	Automotive
TA = 0 to 70°C VSS = 0 V VDD = 5 V ± 10 % F_{OSC} = 1 to 16 MHz	TA = –40 to +85°C ; VSS = 0 V VDD = 5 V ± 10 % F_{OSC} = 1 to 16 MHz	TA = –40 to +125°C VSS = 0 V VDD = 5 V ± 10 % F_{OSC} = 1 to 12 MHz

Symbol	Parameter	Test Conditions	Min	Max	Unit
AVDD	Analog supply voltage	AVDD = VDD±0.2V	4.50	5.50	V
AI_{DD}	Analog supply current: operating	Port 1 = 0 to AVDD		1.20	mA
AV_{IN}	Analog input voltage		AVSS–0.2	AVDD+0.2	V
V_{ref}	Reference voltage		AVSS–0.2	AVDD+0.2	V
R_{ref}	Resistance between V_{ref} and AVSS		1	10	kΩ
C_{IA}	Analog input capacitance			15	pF
t_{ADS}	Sampling time	108 T_{OSC}		6.757 at 16 MHz 9 at 12 MHz 108 at 1 MHz	µs
t_{ADC}	Conversion time (including sampling time)	600 T_{OSC}		37.5 at 16 MHz 50 at 12 MHz 600 at 1 MHz	µs
DLe	Differential non–linearity[1,2]			±1	LSB
ILe	Integral non–linearity[1,3]			±1	LSB
OSe	Offest error[1,4]			±1	LSB
Ge	Gain error[1,5]			0,40	%
M_{CTC}	Channel to channel matching			±1	LSB
C_t	Crosstalk between inputs of Port 1[6]	0 to 100 kHz		–60	dB
T_{OSC}	Oscillator Clock Period		Com, Ind = 62 Auto = 83	1000	ns

Notes:
1. Conditions : AVDD = 5.V; V_{REF} = 5.12V. ADC is monotonic with no missing codes.
2. The differential non–linearity (DLe) is the difference between the actual step width and the ideal step width. (See Figure 3.1.)
3. The integral non–linearity (ILe) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error. (See Figure 3.1.)
4. The offset error (OSe) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. (See Figure 3.1.)
5. The gain error (Ge) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve. (See Figure 3.1.)
6. This should be considered when both analog and digital signals are simultaneously input to Port 1.



(1) Example of an actual transfer curve
(2) The ideal transfer curve
(3) Differential non–linearity (DLe)
(4) Integral on–linearity (ILe)
(5) Center of a step of the actual transfer curve

Figure 3.1. A/D conversion characteristic

EPROM Programming

4.1. Programming modes

The TSC87251A1 derivatives in Window CQPJ are erasable by UV which set all the EPROM memory cells to one and allows a reprogrammation. The other TSC87251A1 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 TSC87251A1 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.

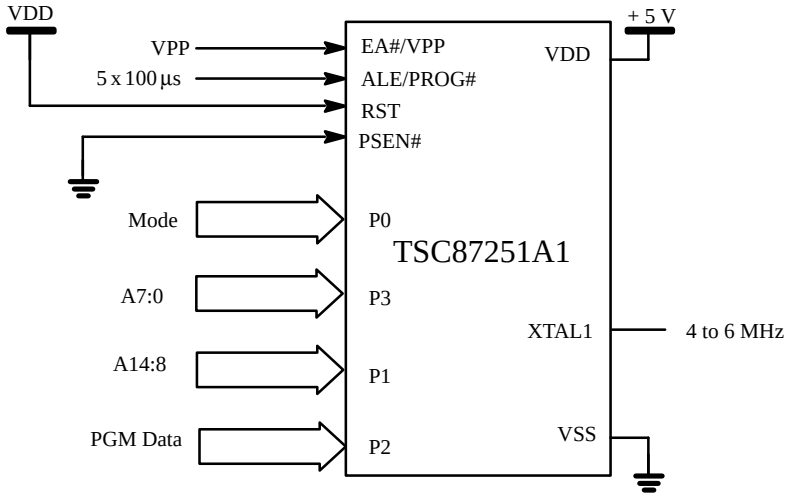
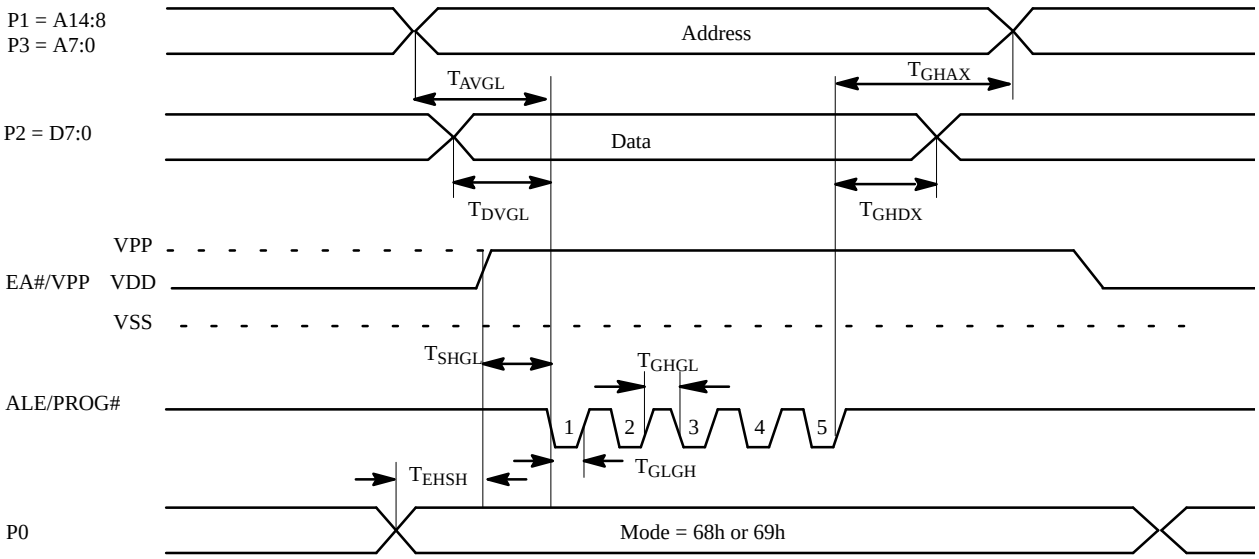


Figure 4.1. Setup for EPROM programming



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 TSC80251A1 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

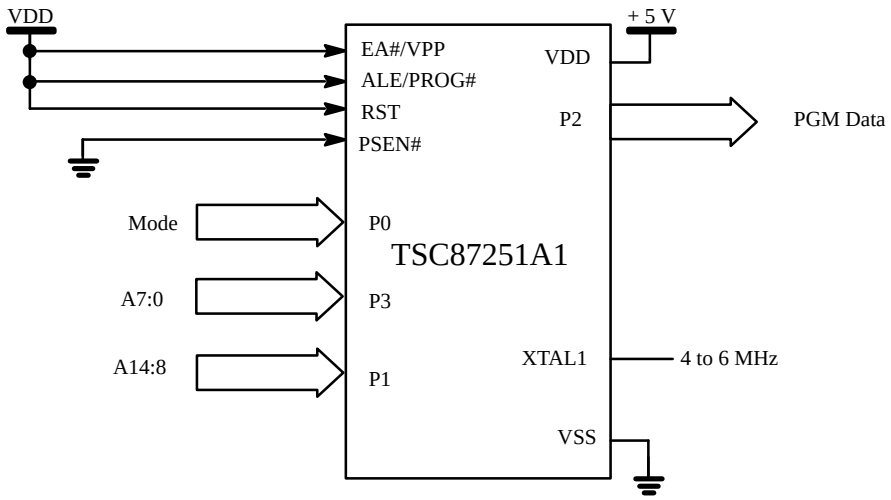


Figure 4.3. Setup for EPROM verification

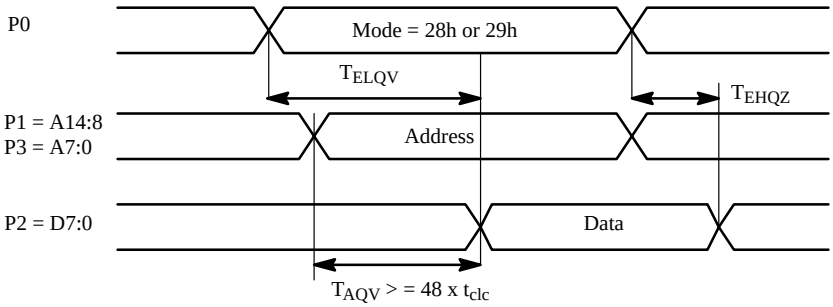


Figure 4.4. Timings for EPROM verification

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. PLCC 44

5.1.1. Mechanical Outline

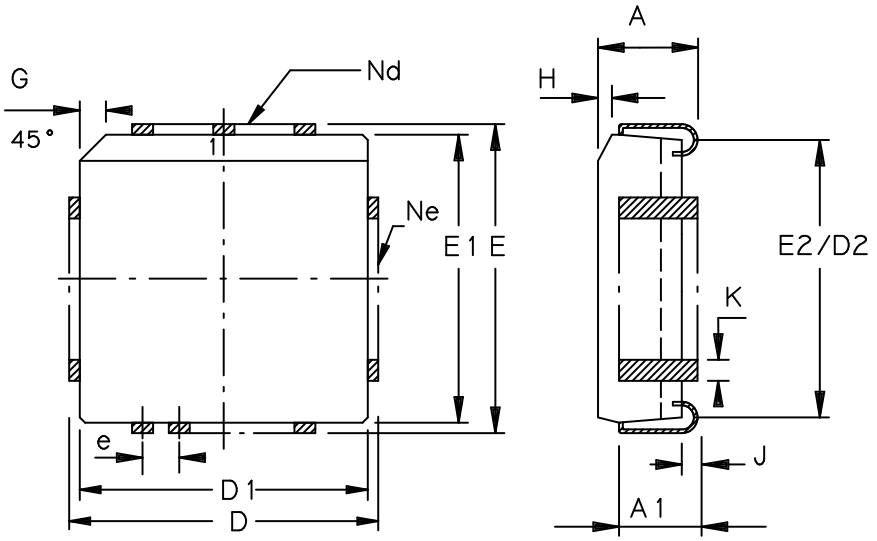


Figure 5.1. Plastic Lead Chip Carrier

Table 5.1. PLCC chip 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	–

	MM		INCH	
	Min	Max	Min	Max
K	0.33	0.53	.013	.021
Nd	11		11	
Ne	11		11	
PKG STD		00		

5.1.2. Pin Assignment

Table 5.2. PLCC pin assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	AVSS	23	P2.0/A8
2	Vref	24	P2.1/A9
3	P1.0/AN0	25	P2.2/A10
4	P1.1/AN1	26	P2.3/A11
5	P1.2/ECI/AN2	27	P2.4/A12
6	P1.3/CEX0/AN3	28	P2.5/A13
7	P1.4/CEX1	29	P2.6/A14
8	P1.5/PMI0/CEX2	30	P2.7/A15
9	P1.6/PMI1CEX3	21	PSEN#
10	P1.7/A17/PMI2/CEX4	32	ALE/PROG#
11	RST	33	VSS0
12	P3.0/RXD	34	VDD0
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	VSS1	44	AVDD

5.2. CQPJ 44 with Window

5.2.1. Mechanical Outline

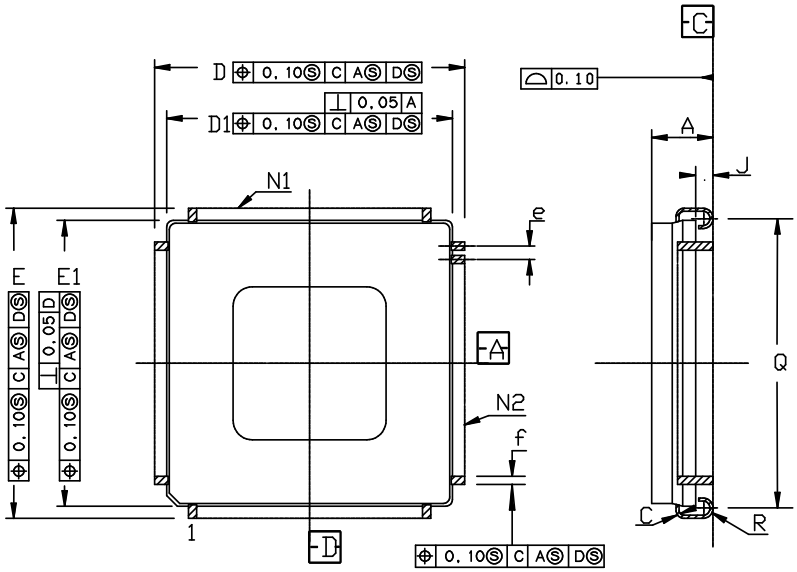


Figure 5.2. Ceramic Quad Pack J

Table 5.3. CQPJ chip 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	

5.2.2. Pin Assignment

Table 5.4. CQPJ pin assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.4/CEX1	23	P2.6/A14
2	P1.5/PMI0/CEX2	24	P2.7/A15
3	P1.6/PMI1CEX3	25	PSEN#
4	P1.7/A17/PMI2/CEX4	26	ALE/PROG#
5	RST	27	VSS0
6	P3.0/RXD	28	VDD0
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	VSS1	38	AVDD
17	P2.0/A8	39	AVSS
18	P2.1/A9	40	Vref
19	P2.2/A10	41	P1.0/AN0
20	P2.3/A11	42	P1.1/AN1
21	P2.4/A12	43	P1.2/ECI/AN2
22	P2.5/A13	44	P1.3/CEX0/AN3

5.3. TQFP 44

5.3.1. Mechanical Outline

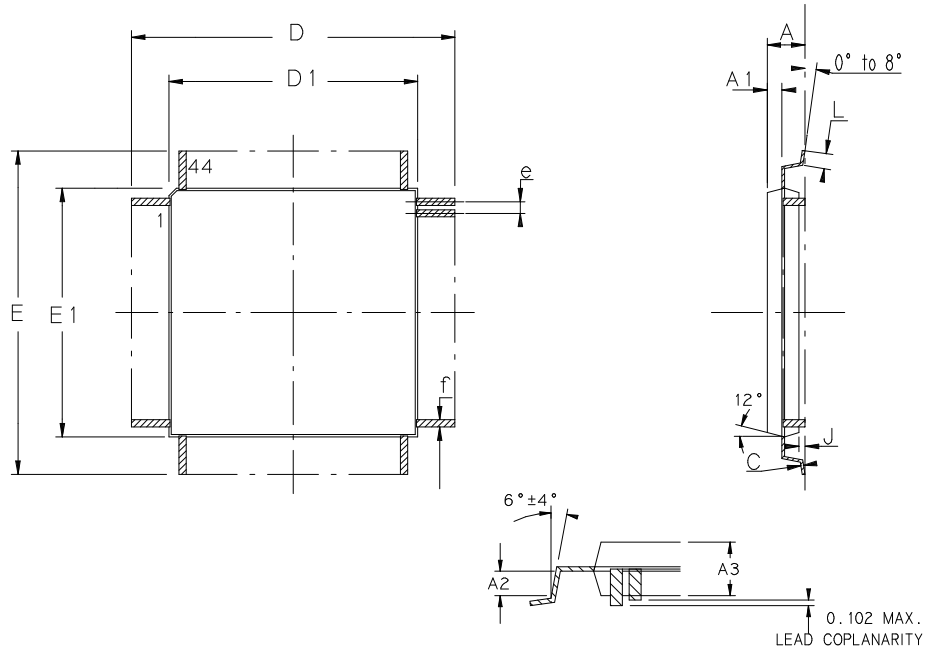


Figure 5.3. Thin Quad Flat Pack (Plastic)

3

Table 5.5. TQFP chip 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	

5.3.2. Pin Assignment

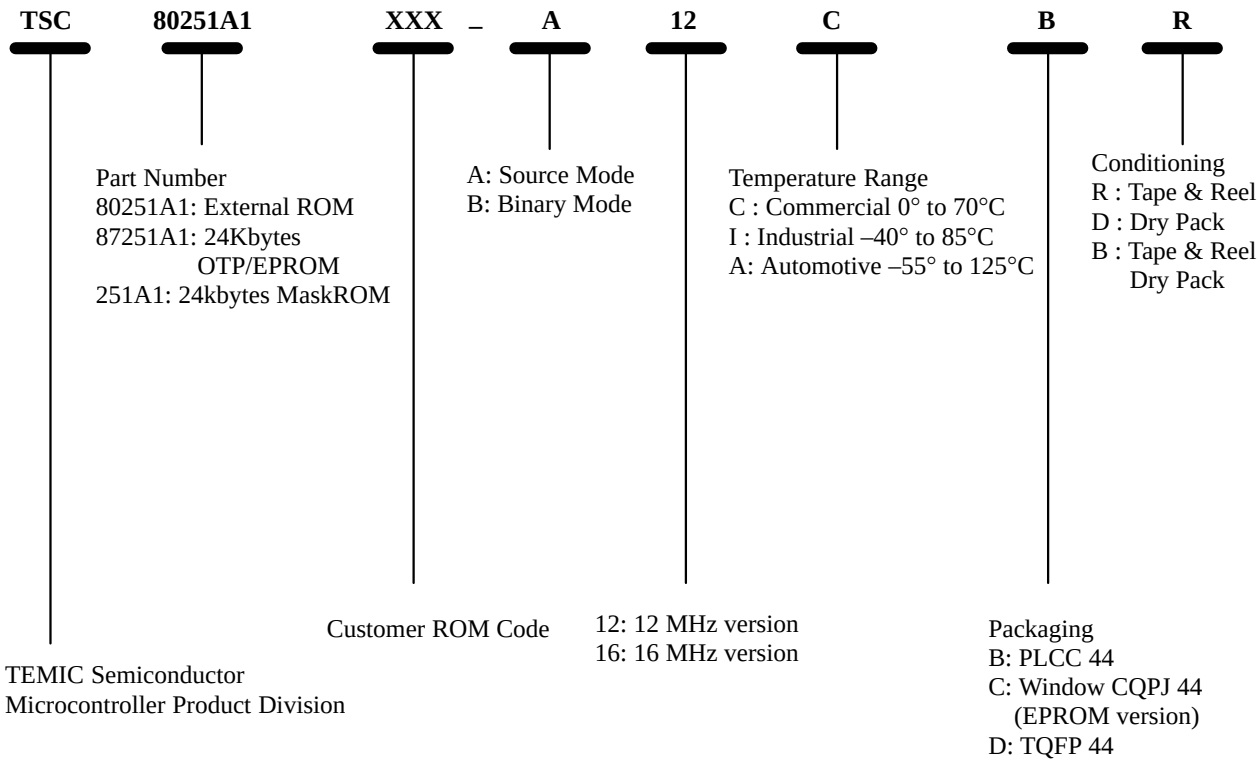
Table 5.6. TQFP pin assignment

Pin Number	Pin Name	Pin Number	Pin Name
1	P1.4/CEX1	23	P2.6/A14
2	P1.5/PMI0/CEX2	24	P2.7/A15
3	P1.6/PMI1CEX3	25	PSEN#
4	P1.7/A17/PMI2/CEX4	26	ALE/PROG#
5	RST	27	VSS0
6	P3.0/RXD	28	VDD0
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	VSS1	38	AVDD
17	P2.0/A8	39	AVSS
18	P2.1/A9	40	Vref
19	P2.2/A10	41	P1.0/AN0
20	P2.3/A11	42	P1.1/AN1
21	P2.4/A12	43	P1.2/ECI/AN2
22	P2.5/A13	44	P1.3/CEX0/AN3

Section IV

Ordering Information

Ordering information



Examples

Part Number	Description
TSC80251A1–A16CBR	ROMless, Source Mode, 16 MHz, PLCC 44, 0 to 70°C, Tape and Reel
TSC87251A1–A12CB	OTP, Source Mode, 12 MHz, PLCC 44, 0 to 70°C
TSC87251A1–A12CBR	EPROM, Source Mode, 12 MHz, PLCC 44, 0 to 70°C, Tape and Reel

Development Tools

Part Number	Description
TSC80251A1–SKA	Software Starter Kit Keil
TSC80251A1–SKB	Software Starter Kit Tasking
TSC80251A1–EKA	Evaluation Kit Keil
TSC80251A1–EKB	Evaluation Kit Tasking

Product Marking :

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