SIEMENS

C511/C511A
C513/C513A
C513A-H

8-Bit CMOS Microcontroller

User's Manual 06.96

User's Manual C511/C511A/C513/C513A/C513A-H **Revision History :**	**Current Version : 06.96**
Previous Releases :	02.96, 05.95
Page	Subjects (changes since last revision)
Several 10-14 10-15	Corrections of text Figure 10-12 : external clock configuration corrected Updated package outline

Edition 06.96

This edition was realized using the software system FrameMaker®.

Published by Siemens AG,
Bereich Halbleiter, Marketing-
Kommunikation, Balanstraße 73,
81541 München

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

Table of Contents **Page**

1 Introduction 1-1

2 Fundamental Structure 2-1
2.1 CPU 2-2
2.2 CPU Timing 2-4

3 Memory Organization 3-1
3.1 Program Memory 3-2
3.2 Data Memory 3-2
3.2.1 Internal Data Memory 3-2
3.2.2 External Data Memory 3-2
3.3 XRAM Operation (C513A/A-H only) 3-3
3.3.1 Reset Operation of the XRAM 3-3
3.3.2 Accesses to XRAM using the DPTR (16-bit Addressing Mode) 3-3
3.3.3 Accesses to XRAM using the Registers R0/R1 (8-bit Addressing Mode) 3-3
3.4 General Purpose Registers 3-4
3.5 Special Function Registers 3-4

4 External Bus Interface 4-1
4.1 Accessing External Memory 4-1
4.1.1 Role of P0 and P2 as Data/Address Bus 4-1
4.1.2 Timing 4-3
4.1.3 External Program Memory Access 4-3
4.2 PSEN - Program Store Enable 4-3
4.3 ALE - Address Latch Enable 4-3
4.4 XRAM Access Enable (SAB-C513A/A-H only) 4-4
4.5 Overlapping External Data and Program Memory Spaces 4-4

5 System Reset 5-1
5.1 Hardware Reset 5-1
5.2 Hardware Reset Timing 5-2

6 On-Chip Peripheral Components 6-1
6.1 Parallel I/O 6-1
6.1.1 Port Structures 6-1
6.1.2 Port 0 and Port 2 used as Address/Data Bus 6-8
6.1.3 Alternate Functions 6-9
6.1.4 Port Handling 6-12
6.1.4.1 Port Timing 6-12
6.1.4.2 Port Loading and Interfacing 6-13
6.1.4.3 Read-Modify-Write Feature of Ports 1, 2 and 3 6-13
6.2 Timers/Counters 6-15
6.2.1 Timer/Counter 0 and 1 6-16
6.2.1.1 Mode 0 6-18
6.2.1.2 Mode 1 6-19
6.2.1.3 Mode 2 6-20
6.2.1.4 Mode 3 6-21
6.2.2 Timer/Counter 2 (SAB-C513/C513A/C513A-H only) 6-22

Table of Contents **Page**

6.2.2.1 Auto-Reload Mode (Up or Down Counter) 6-24
6.2.2.2 Capture Mode 6-26
6.3 General Purpose Serial Interface USART (SAB-C513/C513AC513A-H only) 6-27
6.3.1 Multiprocessor Communications 6-28
6.3.2 Serial Port Control Register 6-29
6.3.3 Baud Rates 6-30
6.3.3.1 Using Timer 1 to Generate Baud Rates 6-31
6.3.3.2 Using Timer 2 to Generate Baud Rates 6-32
6.3.4 Details about Mode 0 6-34
6.3.5 Details about Mode 1 6-37
6.3.6 Details about Modes 2 and 3 6-40
6.4 Synchronous Serial Channel (SSC) 6-43
6.4.1 SSC Block Diagram 6-43
6.4.2 General Operation of the SSC 6-44
6.4.3 Enable/Disable Control 6-44
6.4.4 Baudrate Generation (Master Mode only) 6-45
6.4.5 Write Collision Detection 6-45
6.4.6 Master/Slave Mode Selection 6-46
6.4.7 Data/Clock Timing Relationships 6-47
6.4.7.1 Master Mode Operation 6-47
6.4.7.2 Slave Mode Operation 6-48
6.4.8 Register Description 6-49
6.4.8.1 SSC Control Register SSCCON 6-49
6.4.8.2 SSC Interrupt Enable Register SCIEN 6-51
6.4.8.3 Status Register SCF 6-52
6.4.8.4 Data Registers STB and SRB 6-53
6.4.8.5 Mode Test Register SSCMOD 6-53
6.4.8.6 Location of Bitaddressable Control and Status Bits 6-54

7 Interrupt System 7-1
7.1 Interrupt Structure 7-3
7.2 Interrupt Control Bits 7-4
7.2.1 Interrupt Enables 7-4
7.2.2 Interrupt Priorities 7-5
7.3 How Interrupts are Handled 7-7
7.4 External Interrupts 7-8
7.5 Response Time 7-9

8 Power Saving Modes 8-1
8.1 Idle Mode 8-2
8.2 Power-Down Mode 8-4

9 EEPROM Programming Interface of the SAB-C513A-H 9-1
9.1 Programming Configuration 9-1
9.2 Programming Interface 9-4
9.2.1 Selection of the Programming Mode 9-5
9.2.2 Resetting the Programming Interface 9-5

Table of Contents **Page**

9.2.3 Interface Bus Operation 9-5
9.2.4 Programming Interface Registers 9-7
9.2.4.1 Programming Control Register PCNTRL 9-8
9.2.4.2 Memory Select Register PMSEL 9-9
9.2.4.3 Data Register PDATA 9-10
9.2.4.4 Address Low Register PADRL and PADRH 9-10
9.2.5 EEPROM Access Operations 9-11
9.2.5.1 Erase Operation 9-11
9.2.5.2 Write Operation 9-12
9.2.5.3 Read Operation 9-12

10 Device Specifications 10-13
10.1 Absolute Maximum Ratings 10-13
10.2 DC Characteristics 10-14
10.3 AC Characteristics (applies to all C511/513 Family Microcontrollers) 10-16
10.3.1 Program Memory Characteristics 10-16
10.3.2 External Data Memory Characteristics 10-17
10.3.3 SSC Interface Characteristics 10-18
10.3.4 External Clock Characteristics 10-18
10.3.5 ROM Verification Characteristics (only C511/C511A/C513/C513A) 10-22
10.4 AC Characteristics of C513A-H Programming Interface 10-23
10.4.1 Reset Characteristics (C513A-H only) 10-25
10.5 Package Outlines 10-27

1 Introduction

The C511-R, C511A-R, C513-R, C513A-R, C513A-2R, and C513A-H are members of a family of low cost microcontrollers, which are software compatible with the components of the SAB 8051, SAB 80C51 and C500 families.

The five versions with the “-R” extension contain a non-volatile read-only program memory (ROM). The C513A-H is a version with a 12 Kbyte EEPROM instead of ROM. This device can be used for prototype designs which have a demand for reprogrammable on-chip code memory.

The microcontroller versions differ in functionality according **table 1-1**. They offer different ROM sizes, different RAM/XRAM sizes and a different timer/USART configuration. Common to all devices is an advanced SSC serial port, a second synchronous serial interface, which is compatible to the SPI serial bus industry standard. Except the EEPROM size, the functionality of the C513A-H is a superset of all ROM versions of the C511/513 family.

In this user manual the microcontroller family is also referenced with the term C511/513 and the ROM versions are referenced without the “-R” extension. If some of the features described are applicable to selected family members only, this will be explicitly stated.

Table 1-1
Functionality of the C511/513 MCUs

Device	ROM Size	EEPROM Size	RAM Size	XRAM Size	Timers [1]	USART [2]	SSC [3]
C511	2.5 KB	–	128 B	–	T0, T1	–	✓
C511A	4 KB	–	256 B	–	T0, T1	–	✓
C513	8 KB	–	256 B	–	T0, T1, T2	✓	✓
C513A	12/16 KB	–	256 B	256 B	T0, T1, T2	✓	✓
C513A-H	–	12 KB	256 B	256 B	T0, T1, T2	✓	✓

Note: 1) T0 and T1 are the standard 16-bit timer. T2 is the 16-bit timer with autoreload.
2) USART is the Universal Synchronous/Asynchronous Receive/Transmit interface.
3) SSC is the Synchronous Serial Channel (SPI compatible interface).

According **table 1-1**, **figure 1-1** to **figure 1-4** show the functional units of the members of the C511/513 family microcontrollers.

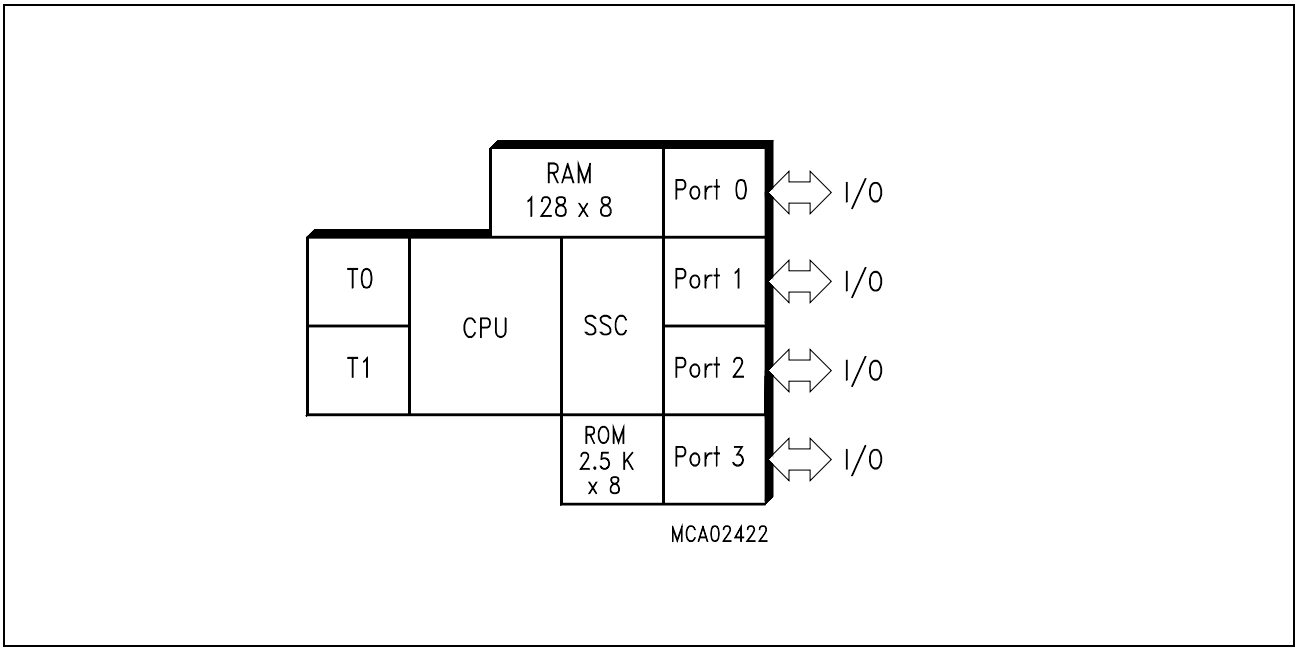


Figure 1-1
C511 Functional Units

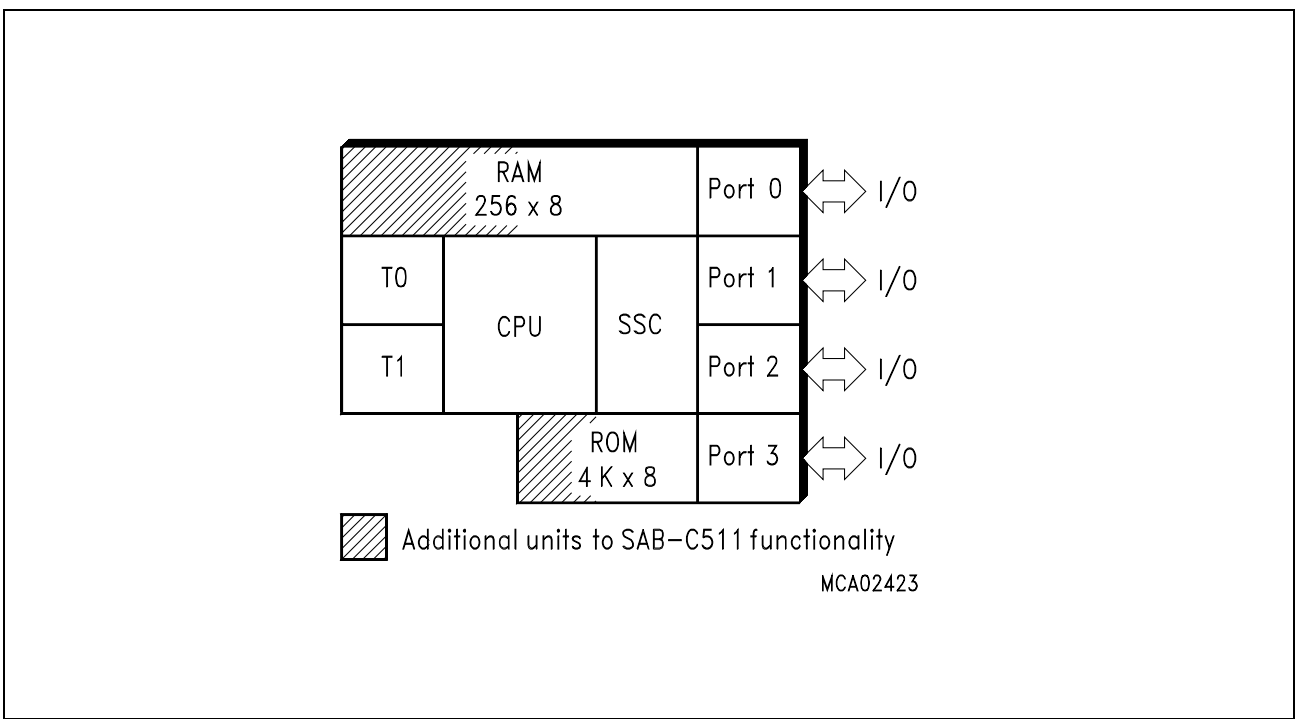


Figure 1-2
C511A Functional Units

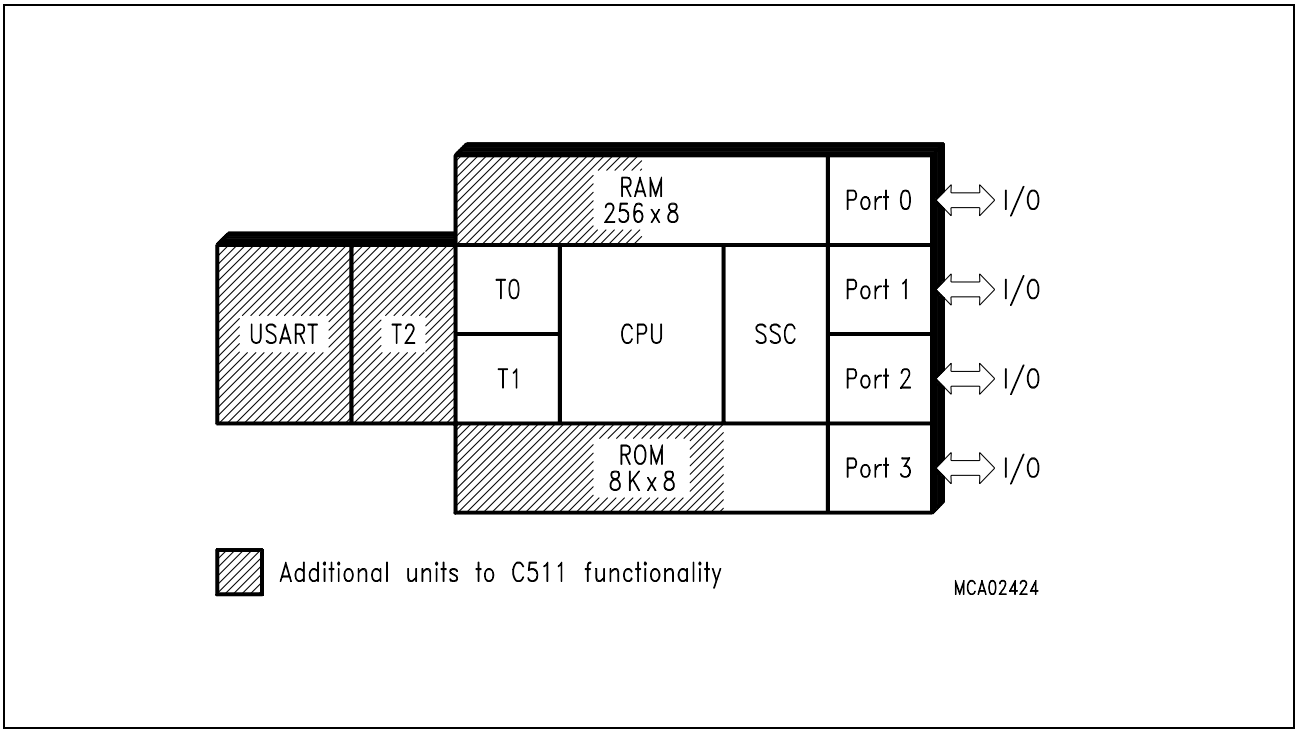


Figure 1-3
C513 Functional Units

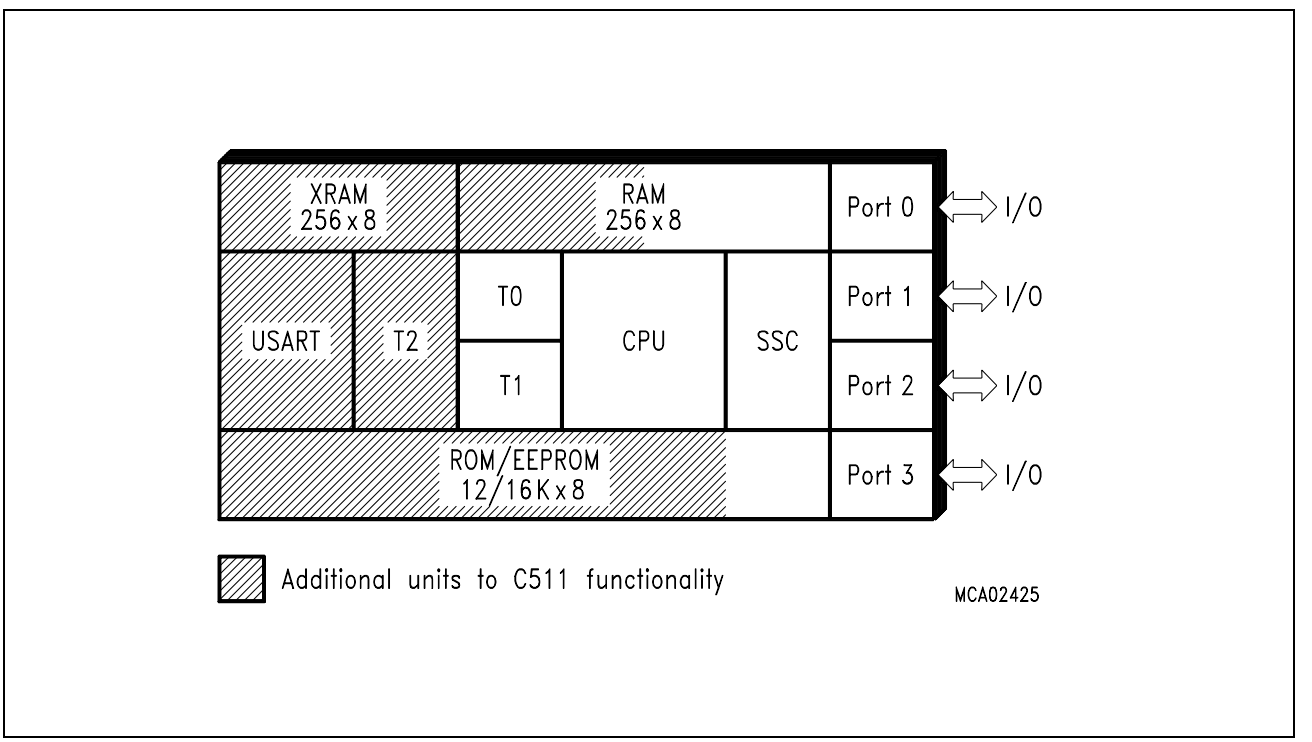


Figure 1-4
C513A / C513A-H Functional Units

Listed below is a summary of the main features of the C511/513 family members:

- Fully software compatible to standard 8051/8052 microcontrollers
- Up to 12 MHz operating frequency
- Up to 12 K×8 ROM / EEPROM
- Up to 256×8 RAM
- Up to 256x8 XRAM
- Four 8-bit ports
- Up to three 16-bit Timers / Counters (Timer 2 with Up/Down and 16-bit autoreload feature)
- Synchronous Serial Channel (SSC)
- Optional USART
- Up to seven interrupt sources, two priority levels
- Power Saving Modes
- P-LCC-44 package (C513A also in P-MQFP-44 package)
- Temperature ranges: SAB-C511 / 511A / 513 / 513A / C513A-H T_A: 0 °C to 70 °C
  SAF-C513A T_A: -40 °C to 85 °C

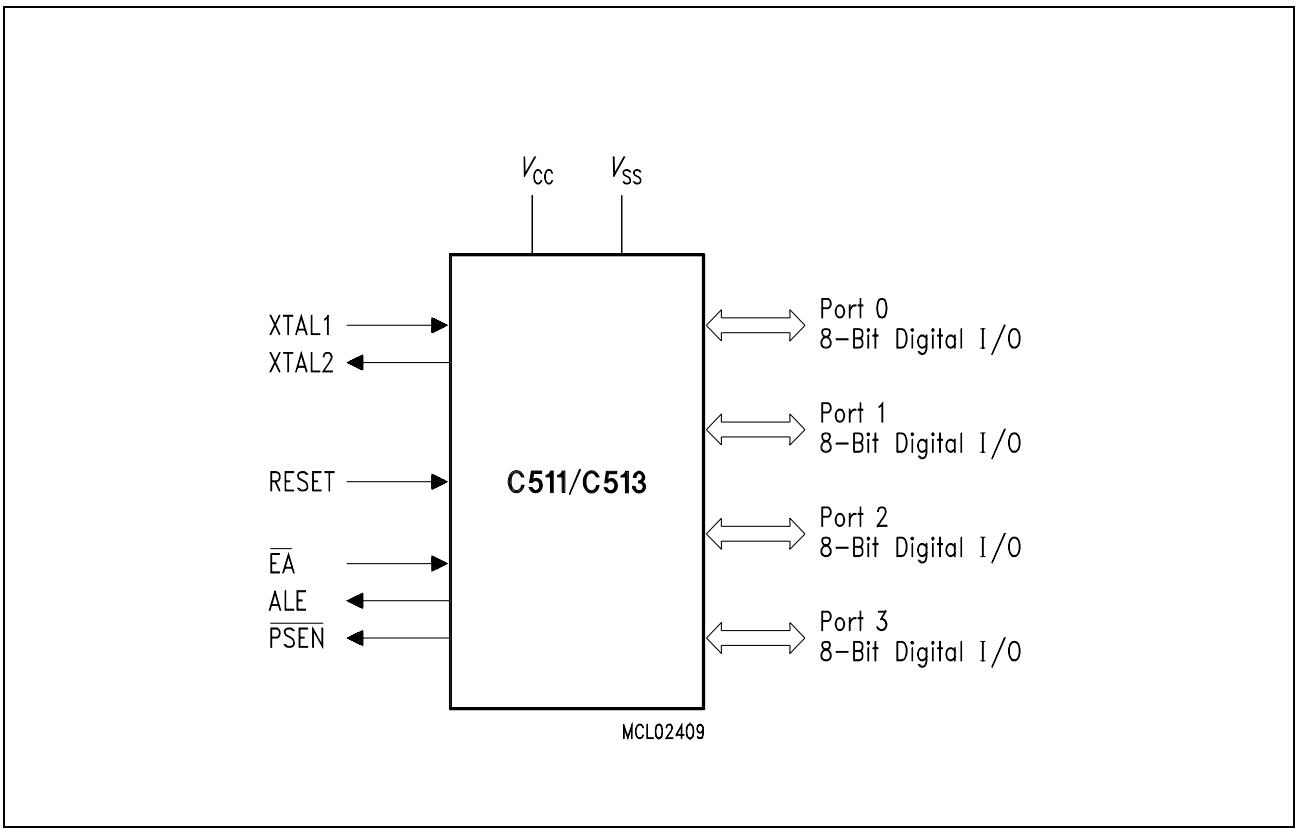


Figure 1-5
C511/C513 Logic Symbol

If the C513A-H is used in programming mode, its logic symbol is different from **figure 1-5** (see **figure 9-1**).

Pin Configuration
(top view)

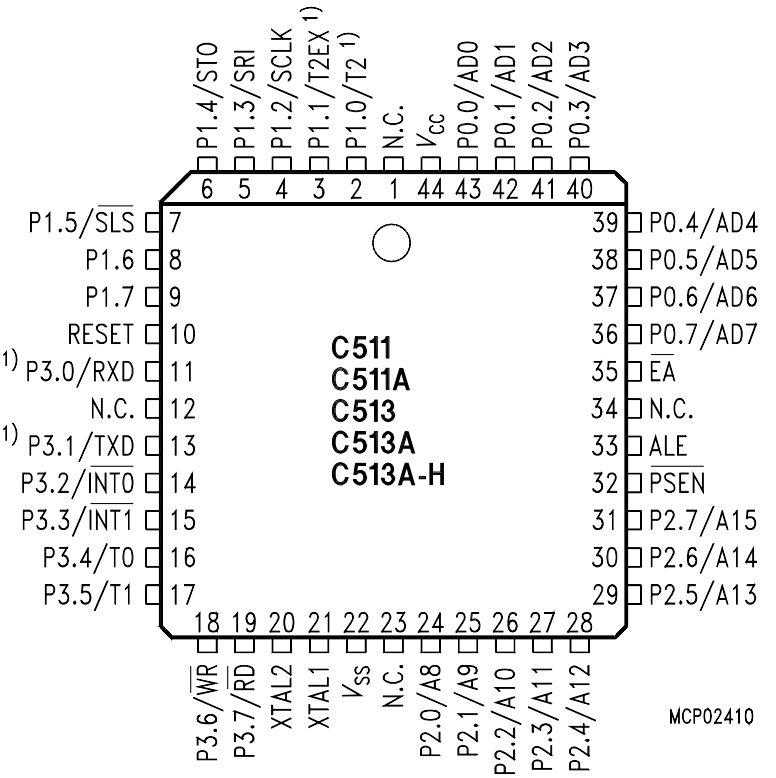


Figure 1-6
P-LCC-44 Package Pin Configuration (Top View)

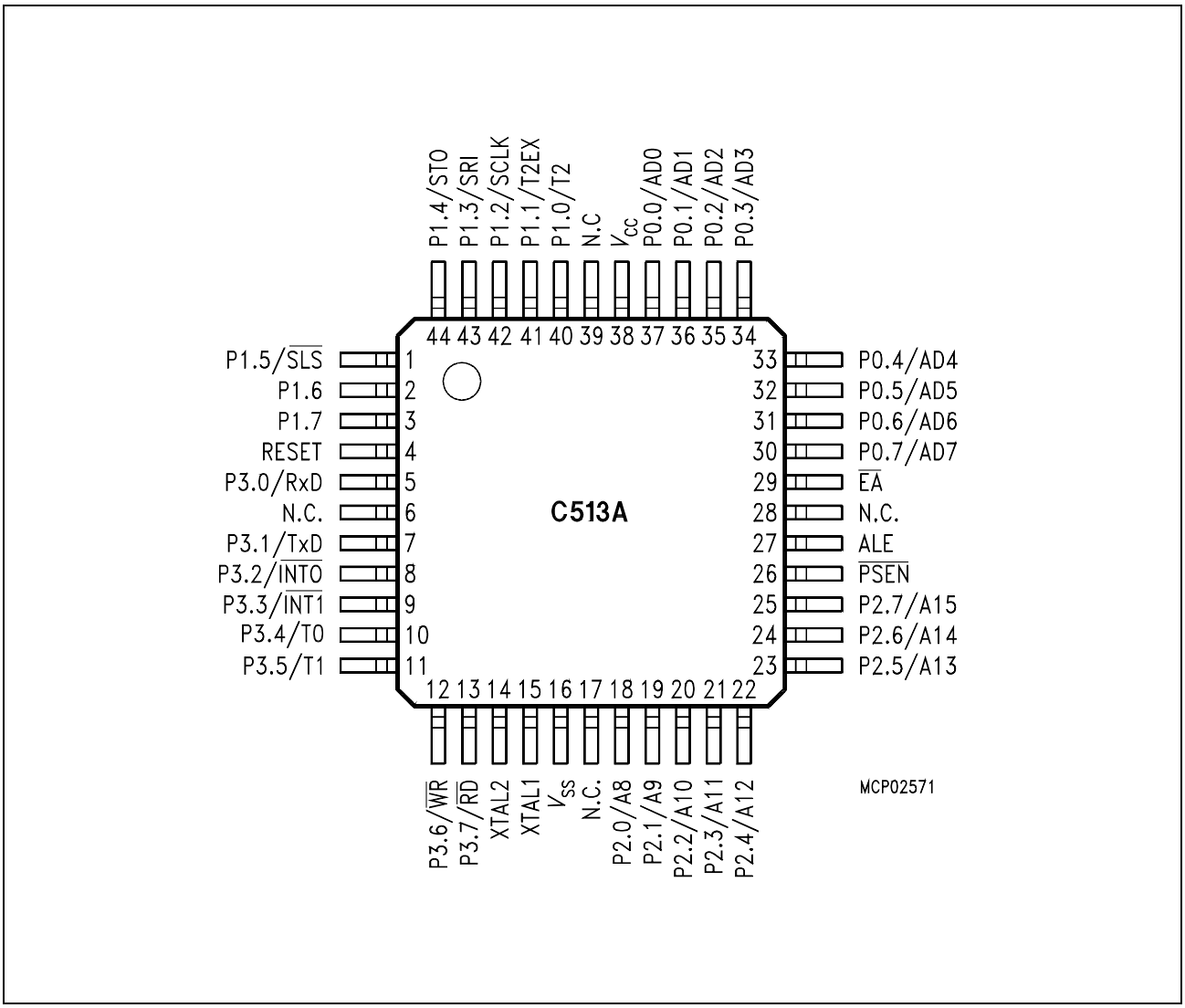


Figure 1-7
P-MQFP-44 Package Pin Configuration of the C513A (Top View)

If the C513A-H is used in programming mode, the pin configuration is different to **figure 1-6** or **figure 1-7.** (see **figure 9-2**)

If the C513A-H is used in programming mode, the pin definitions and functions are different to **table 1-2** (see **table 9-1**).

Table 1-2
Pin Definitions and Functions

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-MQFP-44		
P1.7-P1.0	9-2	3-1, 44-40	I/O	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 1 also contains the timer 2 and SSC pins as secondary function. In general the output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. For the outputs of the SSC (SCLK, STO) special circuitry is implemented, providing true push-pull capability. The STO output in addition will have true tristate capability. When used for SSC inputs, the pull-up resistors will be switched off and the inputs will float (high ohmic inputs). The alternate functions are assigned to port 1, as follows:
	2	40		P1.0 T2 Input to counter 2 [1]
	3	41		P1.1 T2EX Capture -Reload trigger of timer 2 [1] Up-Down count
	4	42		P1.2 SCLK SSC Master Clock Output SSC Slave Clock Input
	5	43		P1.3 SRI SSC Receive Input
	6	44		P1.4 STO SSC Transmit Output
	7	1		P1.5 $\overline{SLS}$ Slave Select Input [1] not available in the C511/511A

*) I = Input
O = Output

Table 1-2
Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-MQFP-44		
P3.0-P3.7	11, 13-19	5, 7-13	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3 as follows:
	11	5		P3.0 RXD Receiver data input (asynchronous) or data input/output (synchronous) of serial interface (USART) [1]
	13	7		P3.1 TXD Transmitter data output (USART) [1] (asynchronous) or clock output (synchronous) of serial interface
	14	8		
	15	9		P3.2 $\overline{INT0}$ $\overline{Interrupt\ 0\ input}$ / timer 0 gate control
	16	10		P3.3 $\overline{INT1}$ $\overline{Interrupt\ 1\ input}$ / timer 1 gate control
	17	11		P3.4 T0 Counter 0 input
	18	12		P3.5 T1 Counter 1 input
				P3.6 $\overline{WR}$ Write control signal : latches the data byte from port 0 into the external
	19	13		data memory
				P3.7 $\overline{RD}$ Read control signal : enables the external data memory to port 0 [1] not available in the C511/511A
XTAL2	20	14	–	**XTAL2** Output of the inverting oscillator amplifier.

*) I = Input
O = Output

Table 1-2
Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-MQFP-44		
XTAL1	21	15	–	**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
P2.0-P2.7	24-31	18-25	I/O	**Port 2** is a bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
$\overline{\text{PSEN}}$	32	26	O	The **$\overline{\text{Program Store Enable}}$** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periodes except during external data memory accesses. Remains high during internal program execution.
RESET	10	4	I	**RESET** A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal resistor to V_{SS} permits power-on reset using only an external capacitor to V_{CC}.

*) I = Input
O = Output

Table 1-2
Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O*)	Function
	P-LCC-44	P-MQFP-44		
ALE	33	27	O	The **Address Latch Enable** output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periodes except during an external data memory access. If no external memory is used, the ALE signal generation can be inhibited, reducing system RFI, by clearing register bit EALE in the SYSCON register.
$\overline{EA}$	35	29	I	**$\overline{External\ Access\ Enable}$** When held at high level, instructions are fetched from the internal ROM when the PC is less than the size of the internal ROM : C511 $0A00_H$; C511A 1000_H; C513 2000_H; C513A/A-H 3000_H; C513A-2R 4000_H. When held at low level, the microcontroller fetches all instructions from external program memory.
P0.0-P0.7	43-36	37-30	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impendance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pullup transistors when issuing 1s. External pullup resistors are required during program verification.
V_{SS}	22	16	–	**Circuit ground potential**
V_{CC}	44	38	–	**Power Supply terminal** for all operating modes
N.C.	1, 12, 23, 34	6, 17, 28, 39	–	**No connection**, do not connect externally

*) I = Input
O = Output

2 Fundamental Structure

The SAB-C511/513 family microcontrollers are based on the SAB-C501 architecture. Therefore they are also fully compatible to the standard 8051 microcontroller family.

The completely new units compared to the SAB-C501 are the synchronous serial channel, the XRAM, and the EEPROM code memory.

Figure 2-8 shows a block diagram of the SAB-C511/513 family microcontrollers.

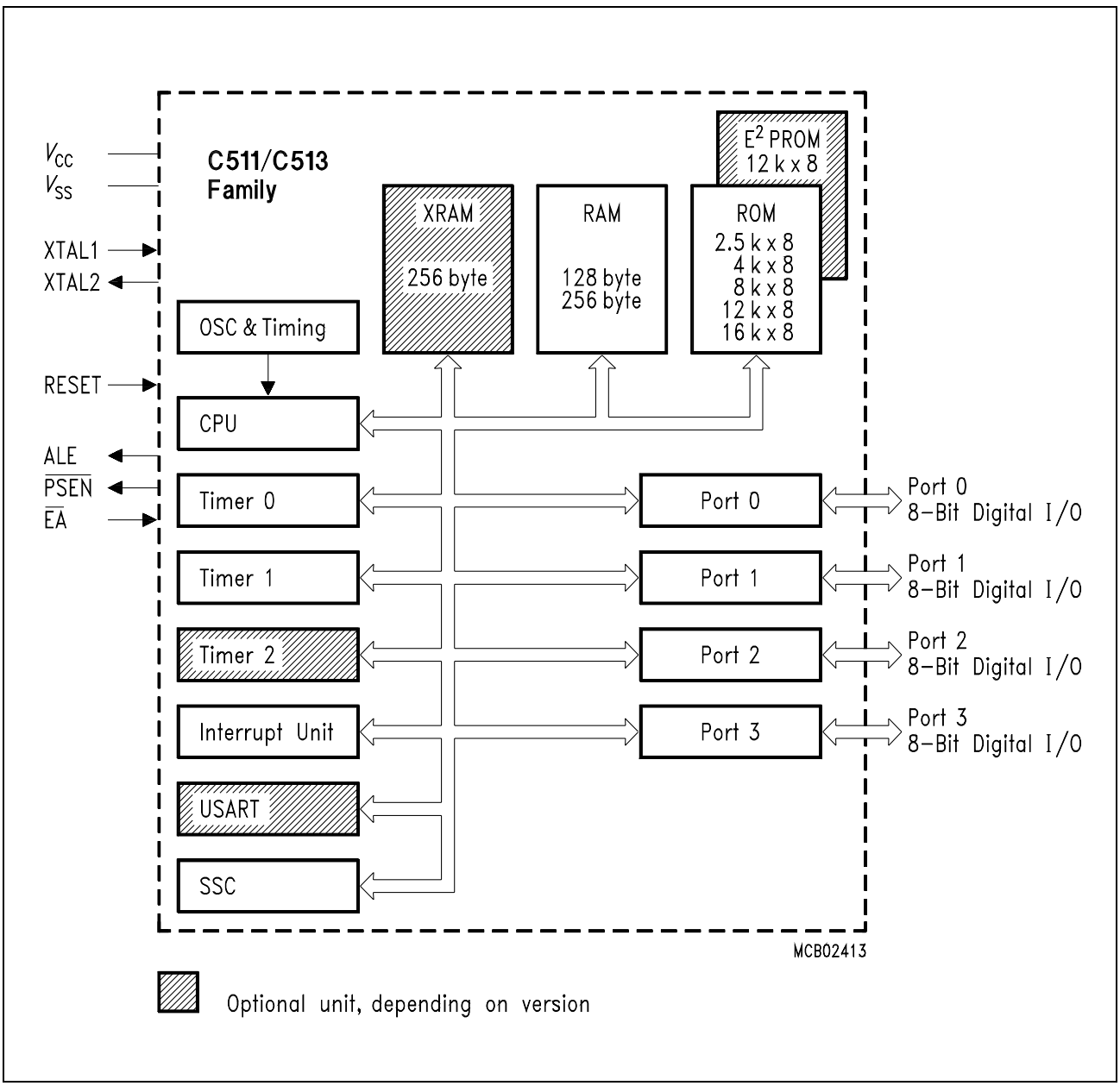


Figure 2-8
Block Diagram of the SAB-C511/513

2.1 CPU

The SAB-C511/513 microcontrollers are efficient both as a controller and as an arithmetic processor. It have extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With an 8 MHz clock, 58% of the instructions are executed in 1.5 µs.

The CPU (Central Processing Unit) of the SAB-C511/513 consists of the instruction decoder, the arithmetic section and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers, and control the ALU processing.

The arithmetic section of the processor performs extensive data manipulation and is comprised of the arithmetic/logic unit (ALU), an A register, B register and PSW register.

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under the control of the instruction decoder. The ALU performs the arithmetic operations add, substract, multiply, divide, increment, decrement, BDC-decimal-add-adjust and compare, and the logic operations AND, OR, Exclusive OR, complement and rotate (right, left or swap nibble (left four)). Also included is a Boolean processor performing the bit operations as set, clear, completement, jump-if-not-set, jump-if-set-and-clear and move to/from carry. Between any addressable bit (or its complement) and the carry flag, it can perform the bit operations of logical AND or logical OR with the result returned to the carry flag.

The program control section controls the sequence in which the instructions stored in program memory are executed. The 16-bit program counter (PC) holds the address of the next instruction to be executed. The conditional branch logic enables internal and external events to the processor to cause a change in the program execution sequence.

Accumulator

ACC is the symbol for the accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU.

Special Function Register PSW (Address $D0_H$) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$D0_H$	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

Bit	Function
CY	Carry Flag
AC	Auxiliary Carry Flag (for BCD operations)
F0	General Purpose Flag
RS1 RS0 0 0 0 1 1 0 1 1	Register Bank select control bits Bank 0 selected, data address 00_H-07_H Bank 1 selected, data address 08_H-$0F_H$ Bank 2 selected, data address 10_H-17_H Bank 3 selected, data address 18_H-$1F_H$
OV	Overflow Flag
F1	General Purpose Flag
P	Parity Flag Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.

B Register

The B register is used during multiply and divide and serves as both source and destination. For other instructions it can be treated as another scratch pad register.

Stack Pointer

The stack pointer (SP) register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions and decremented after data is popped during a POP and RET (RETI) execution, i.e. it always points to the last valid stack byte. While the stack may reside anywhere in the on-chip RAM, the stack pointer is initialized to 07_H after a reset. This causes the stack to begin a location = 08_H above register bank zero. The SP can be read or written under software control.

2.2 CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is devided into a phase 1 half, during which the phase 1 clock is active, and a phase 2 half, during which the phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts for two oscillator periods. Typically, arithmetic and logical operations take place during phase 1 and internal register-to-register transfers take place during phase 2.

The diagrams in figure 2-2-9 show the fetch/execute timing related to the internal states and phases. Since these internal clock signals are not user-accessible, the XTAL2 oscillator signals and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

Executing of a one-cycle instruction begins at S1P2, when the op-code is latched into the instruction register. If it is a two-byte instruction, the second reading takes place during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op-code) is ignored (discarded fetch), and the program counter is not incremented. In any case, execution is completed at the end of S6P2.

Figures 2-2-9 a) and b) show the timing of a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most SAB-C511/513 instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed. **Figure 2-2-9 c) and d)** show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

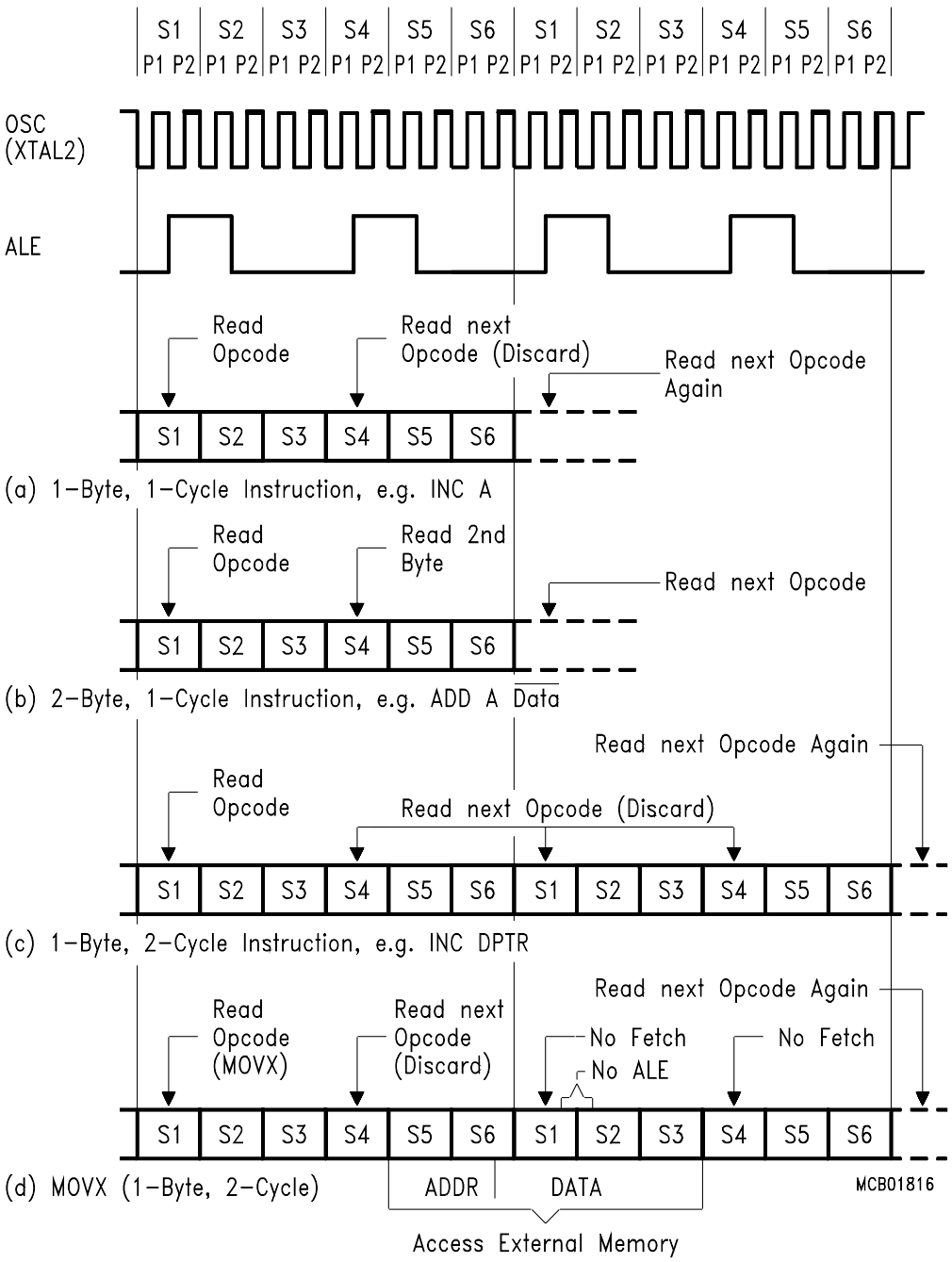


Figure 2-9
Fetch Execute Sequence

3 Memory Organization

The C511/513 CPU manipulates operands in the following four address spaces:

- up to 64 Kbyte of external program memory
- up to 16 Kbyte of internal program memory
- up to 64 Kbyte of external data memory
- up to 256 bytes of internal data memory (includes bitaddressable area of 128 bits)
- 256 bytes additional internal data memory (XRAM)
- a 128 byte special function register area with 16 bitaddressable registers (128 bits)

Figure 3-10 illustrates the memory address spaces of the C511/513.

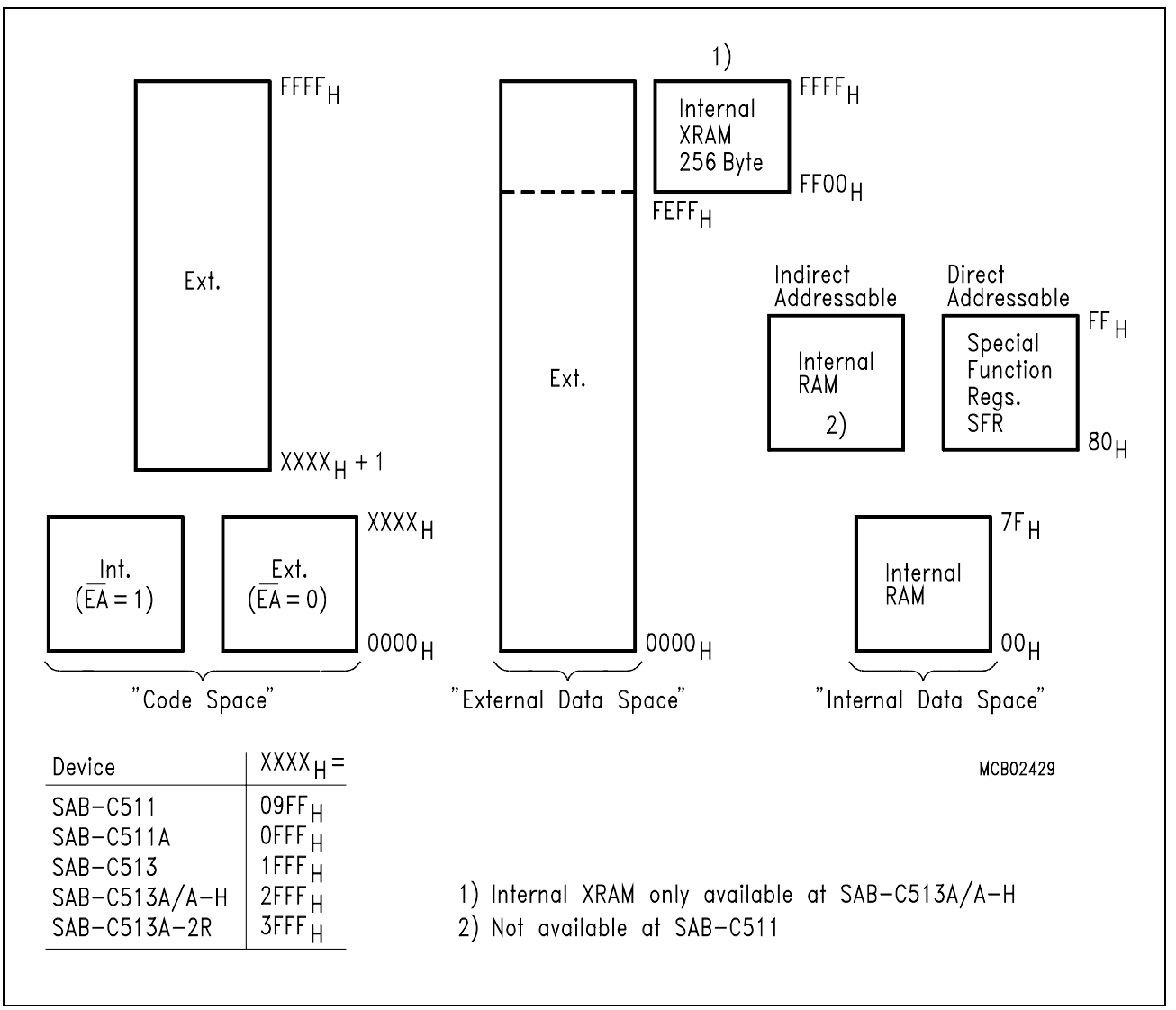


Figure 3-10
Memory Map

3.1 Program Memory

The C511/513 family members have up to 12 Kbytes of on-chip read-only program memory (depending on version). The program memory can be externally expanded up to 64 Kbytes. If the $\overline{EA}$ pin is held high, the C511/513 executes out of internal ROM unless the address exceeds the upper limit of the on-chip program memory (upper limit table see **figure 3-10**). Instructions at addresses above this limit are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the C511/513 fetches all instructions from the external program memory.

3.2 Data Memory

The data memory address space consists of an internal and an external memory space. The different versions of the C511/513 microcontroller family provide 128 (C511 only) or 256 byte (all other members) of internal data memory.

The C513A/A-H contain another 256 byte of on-chip RAM additional to its 256 byte internal RAM. This RAM is referenced as XRAM (‘e**X**tended **RAM**’).

3.2.1 Internal Data Memory

The internal data memory is divided into four physically separate and distinct blocks:

- the lower 128 byte of RAM including four register banks containing eight registers each
- the upper 128 byte of RAM
- the 128 byte special function register area
- a 256 byte XRAM area which is accessed like external RAM (MOVX-instructions), but is implemented on-chip (C513A/A-H only).

3.2.2 External Data Memory

Up to 64 Kbyte external data memory can be addressed by instructions that use 8-bit or 16-bit indirect addressing. A 16-bit external memory addressing mode is supported by the MOVX instructions using the 16-bit datapointer DPTR for addressing. For 8-bit addressing MOVX instructions in combination with registers R0 and R1 can be used.

Note that there are restrictions with the C513A/A-H and 8-bit addressing mode when accessing external data memory (details see **chapters 3.3.2 and 3.3.3**).

3.3 XRAM Operation (C513A/A-H only)

The XRAM is a memory area that is logically located in the external memory space, but is integrated on the chip. Because the XRAM is used in the same way as external data memory the same instruction types must be used for accessing the XRAM. The XRAM can be enabled and disabled by the XMAP bit in the SYSCON register (see **chapter 4.4**).

3.3.1 Reset Operation of the XRAM

The content of the XRAM is not affected by a reset. After power-up the content is undefined, while it remains unchanged during and after a reset as long as the power supply is not turned off.

If a reset occurs during a write operation to XRAM, the content of a XRAM memory location depends on the cycle which the reset is detected at (MOVX is a 2-cycle instruction):

Reset during 1st cycle : The new value will not be written to XRAM. The old value is not affected.
Reset during 2nd cycle : The old value in XRAM is overwritten by the new value.

After reset the XRAM is disabled.

3.3.2 Accesses to XRAM using the DPTR (16-bit Addressing Mode)

There are a read and a write instruction to access the XRAM which use the 16-bit DPTR for indirect addressing. The instructions are :

- MOVX A, @DPTR (Read)
- MOVX @DPTR, A (Write)

Use of these instructions normally implies, that port 0 is used as address low/data bus, port 2 for high address output and parts of port 3 for control to access upto 64 KB of external memory. If the XRAM is disabled, this will happen as with the other members of the C511/513 family. If it is enabled and if the effective address is in the range of 0000_H to $FEFF_H$, these instruction also will access external memory.

If XRAM is enabled and if the address is within $FF00_H$ to $FFFF_H$, the physically internal XRAM of the C513A/A-H will be accessed. Physically external memory in this address range cannot be accessed in this case, because no external bus cycles will generated. Therefore port 0, 2 and 3 can be used as general purpose I/O if only the XRAM memory space is addressed by the user program.

3.3.3 Accesses to XRAM using the Registers R0/R1 (8-bit Addressing Mode)

The C511/513 architecture provides also instructions for accesses to external data memory which use only an 8-bit address (indirect addressing with registers R0 or R1). These instructions are :

- MOVX A, @Ri (Read)
- MOVX @Ri, A (Write)

Use of these instructions implies, that port 0 is used as address/data bus, port 2 for high address output and parts of port 3 for control. Normally these instructions are used to access up to 256 byte of external memory.

If the XRAM is disabled, this will happen as with the other members of the C511/513 components and the external memory is accessed.

If the XRAM is enabled these instruction will only access the internal XRAM. External memory cannot be accessed in this case because no external bus cycle will be generated. Therefore, port 0 and 3 can be used as standard I/O, if only the internal XRAM is used.

3.4 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general purpose registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word PSW, RS1 and RS0, select the active register bank (see description of the PSW in **chapter 2.1**). This allows fast context switching, which is useful when entering subroutines or interrupt service routines. After reset register bank 0 is selected.

The 8 general purpose registers of the selected register bank may be accessed by register addressing. With register addressing the instruction opcode indicates which register is to be used. For indirect addressing R0 and R1 are used as pointer or index register to address internal or external memory (e.g. MOV @R0).

Reset initializes the stack pointer to location 07_H and increments it once to start from location 08_H which is also the first register (R0) of register bank 1. Thus, if one is going to use more than one register bank, the SP should be initialized to a different location of the RAM which is not used for data storage.

3.5 Special Function Registers

All registers except the program counter and the four general purpose register banks reside in the special function register area.

The 34 special function registers (SFR) include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. There are also 128 directly addressable bits within the SFR area.

All SFRs are listed in **table 3-1 and table 3-2**. In **table 3-1** they are organized in groups which refer to the functional blocks of the C511/513. **Table 3-2** illustrates the contents of the SFRs, e.g. the bits of the SFRs.

Table 3-2
Contents of the SFRs, SFRs in Numeric Order of their Addresses

Addr	Register	Content after Reset [1)]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80_H	P0	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
81_H	SP	07_H	.7	.6	.5	.4	.3	.2	.1	.0
82_H	DPL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
83_H	DPH	00_H	.7	.6	.5	.4	.3	.2	.1	.0

Table 3-1
SFRs - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	**$E0_H$** [1]	**00_H**
	B	B-Register	**$F0_H$** [1]	**00_H**
	DPH	Data Pointer, High Byte	83_H	00_H
	DPL	Data Pointer, Low Byte	82_H	00_H
	PSW	Program Status Word	**$D0_H$** [1]	**00_H**
	SP	Stack Pointer	81_H	07_H
	SYSCON	System Control Reg. C511/C511A	$B1_H$	$101X0XXX_B$ [3]
		C513/C513A/C513A-H	$B1_H$	$101X0XX0_B$ [3]
Interrupt System	IE	Interrupt Enable Register	**$A8_H$** [1]	**00_H**
	IP	Interrupt Priority Register	**$B8_H$** [1]	**$X0000000_B$** [3]
Ports	P0	Port 0	**80_H** [1]	**FF_H**
	P1	Port 1	**90_H** [1]	**FF_H**
	P2	Port 2	**$A0_H$** [1]	**FF_H**
	P3	Port 3	**$B0_H$** [1]	**FF_H**
SSC	SSCON	SSC Control Register	**$E8_H$** [1]	**07_H**
	STB	SSC Transmit Buffer	$E9_H$	XX_H [3])
	SRB	SSC Receive Register	EA_H	XX_H [3])
	SCF	SSC Flag Register	**$F8_H$** [1]	**$XXXXXX00_B$** [3]
	SCIEN	SSC Interrupt Enable Register	$F9_H$	$XXXXXX00_B$ [3]
	SSCMOD	SSC Mode Test Register	EB_H	00_H
USART	PCON [2]	Power Control Register	87_H	$0XXX0000_B$ [3]
	SBUF	Serial Channel Buffer Register	99_H	XX_H [3])
	SCON	Serial Channel 1 Control Register	**98_H** [1]	00_H
Timer 0 / Timer 1	TCON	Timer Control Register	**88_H** [1]	**00_H**
	TMOD	Timer Mode Register	89_H	00_H
	TL0	Timer 0, Low Byte	$8A_H$	00_H
	TL1	Timer 1, Low Byte	$8B_H$	00_H
	TH0	Timer 0, High Byte	$8C_H$	00_H
	TH1	Timer 1, High Byte	$8D_H$	00_H
Timer 2	T2CON	Timer 2 Control Register	**$C8_H$** [1]	00_H
	T2MOD	Timer 2 Mode Register	$C9_H$	$XXXXXXX0_B$ [3]
	RC2L	Timer 2 Reload/Capture Register, Low Byte	CA_H	00_H
	RC2H	Timer 2 Reload/Capture Register, High Byte	CB_H	00_H
	TL2	Timer 2 Low Byte	CC_H	00_H
	TH2	Timer 2 High Byte	CD_H	00_H
Power Save Mode	PCON [2]	Power Control Register	87_H	$0XXX0000_B$ [3]

[1] Bit-addressable special function registers
[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.
[3] X means that the value is indeterminate and the location is reserved

Table 3-2
Contents of the SFRs, SFRs in Numeric Order of their Addresses (cont'd)

Addr	Register	Content after Reset [1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
87_H	PCON	$0XXX\text{-}0000_B$	SMOD	–	–	–	GF1	GF0	PDE	IDLE
88_H	TCON	00_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89_H	TMOD	00_H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
$8A_H$	TL0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8B_H$	TL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8C_H$	TH0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8D_H$	TH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
90_H	P1	FF_H	–	–	$\overline{SLS}$	STO	SRI	SCLK	T2EX	T2
98_H	SCON	00_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99_H	SBUF	XX_H	.7	.6	.5	.4	.3	.2	.1	.0
$A0_H$	P2	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
$A8_H$	IE	00_H	EAL	ESSC	ET2	ES0	ET1	EX1	ET0	EX0
$B0_H$	P3	FF_H	RD	WR	T1	T0	INT1	INT0	TxD	RxD
$B1_H$	SYSCON	[2]	1	0	EALE	–	0	–	–	XMAP[2]
$B8_H$	IP	$X000\text{-}0000_B$	–	PSSC	PT2	PS	PT1	PX1	PT0	PX0
$C8_H$	T2CON	00_H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
$C9_H$	T2MOD	$XXXX\text{-}XXX0_B$	–	–	–	–	–	–	–	DCEN
CA_H	RC2L	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CB_H	RC2H	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CC_H	TL2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CD_H	TH2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$D0_H$	PSW	00_H	CY	AC	F0	RS1	RS0	OV	F1	P
$E0_H$	ACC	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E8_H$	SSCCON	07_H	SCEN	TEN	MSTR	CPOL	CPHA	BRS2	BRS1	BRS0
$E9_H$	STB	XX_H	.7	.6	.5	.4	.3	.2	.1	.0
EA_H	SRB	XX_H	.7	.6	.5	.4	.3	.2	.1	.0
EB_H	SSCMOD	00_H [3]	0	0	0	0	0	0	0	0
$F0_H$	B	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F8_H$	SCF	$XXXX\text{-}XX00_B$	–	–	–	–	–	–	WCOL	TC

Table 3-2
Contents of the SFRs, SFRs in Numeric Order of their Addresses (cont'd)

Addr	Register	Content after Reset [1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$F9_H$	SCIEN	XXXX-XX00$_B$	–	–	–	–	–	–	WCEN	TCEN

1) X means that the value is indeterminate and the location is reserved.
2) The availability of the XMAP bit and the reset value of SYSCON depends on the specific microcontroller :
C511/511A : $101X0XXX_B$ - bit XMAP is not available
C513/513A/513A-H $101X0XX0_B$ - bit XMAP is available
3) This register ist only used for test purposes and must not be written. Otherwise unpredictable results may occur.
Shaded registers are bit-addressable special function registers.

4 External Bus Interface

The SAB-C511/513 microcontrollers allow external memory expansion. The functionality and implementation of the external bus interface is identical to the common interface for the 8051 architecture with one exception : if the SAB-C511/513 is used in systems with no external memory the generation of the ALE signal can be suppressed. Resetting bit EALE in SFR SYSCON register, the ALE signal will be gated and no more generated externally. This feature reduces RFI emmisions of the system.

4.1 Accessing External Memory

It is possible to distinguish between accesses to external program memory and external data memory or other peripheral components respectively. This distinction is made by hardware. Accesses to external program memory use the signal $\overline{\text{PSEN}}$ (program store enable) as a read strobe. Accesses to external data memory use $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Port 0 and port 2 (with exceptions) are used to provide data and address signals. In this section only the port 0 and port 2 functions relevant to external memory accesses are described.

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

4.1.1 Role of P0 and P2 as Data/Address Bus

When used for accessing external memory, port 0 provides the data byte time-multiplexed with the low byte of the address. In this state, port 0 is disconnected from its own port latch and the address/ data signal drives both FETs in the port 0 output buffers. Thus, in this application the port 0 pins are not open-drain outputs and do not require external pullup resistors.

During any access to external memory, the CPU writes FF_H to the port 0 latch (the special function register), thus obliterating whatever information the port 0 SFR may have been holding.

Whenever a 16-bit address is used (MOVX @DPTR), the high byte of the address comes out on port 2, where it is held for the duration of the read or write cycle. During this time, the port 2 lines are disconnected from the port 2 latch (the special function register). Thus the port 2 latch does not have to contain 1s, and the contents of the port 2 SFR are not modified. If the XRAM is enabled (only SAB-C513A/A-H) at 16-bit address accesses with an address value within the XRAM address space, no external bus cycle will be seen, but the internal XRAM will be accessed.

If an 8-bit address is used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. It should be noted that, if a port 2 pin outputs an address bit that is a 1, strong pullups will be used for the entire read/write cycle and not only for two oscillator periods. If the XRAM is enabled at the SAB-C513A/A-H no external bus cycle will be seen regardless of the address.

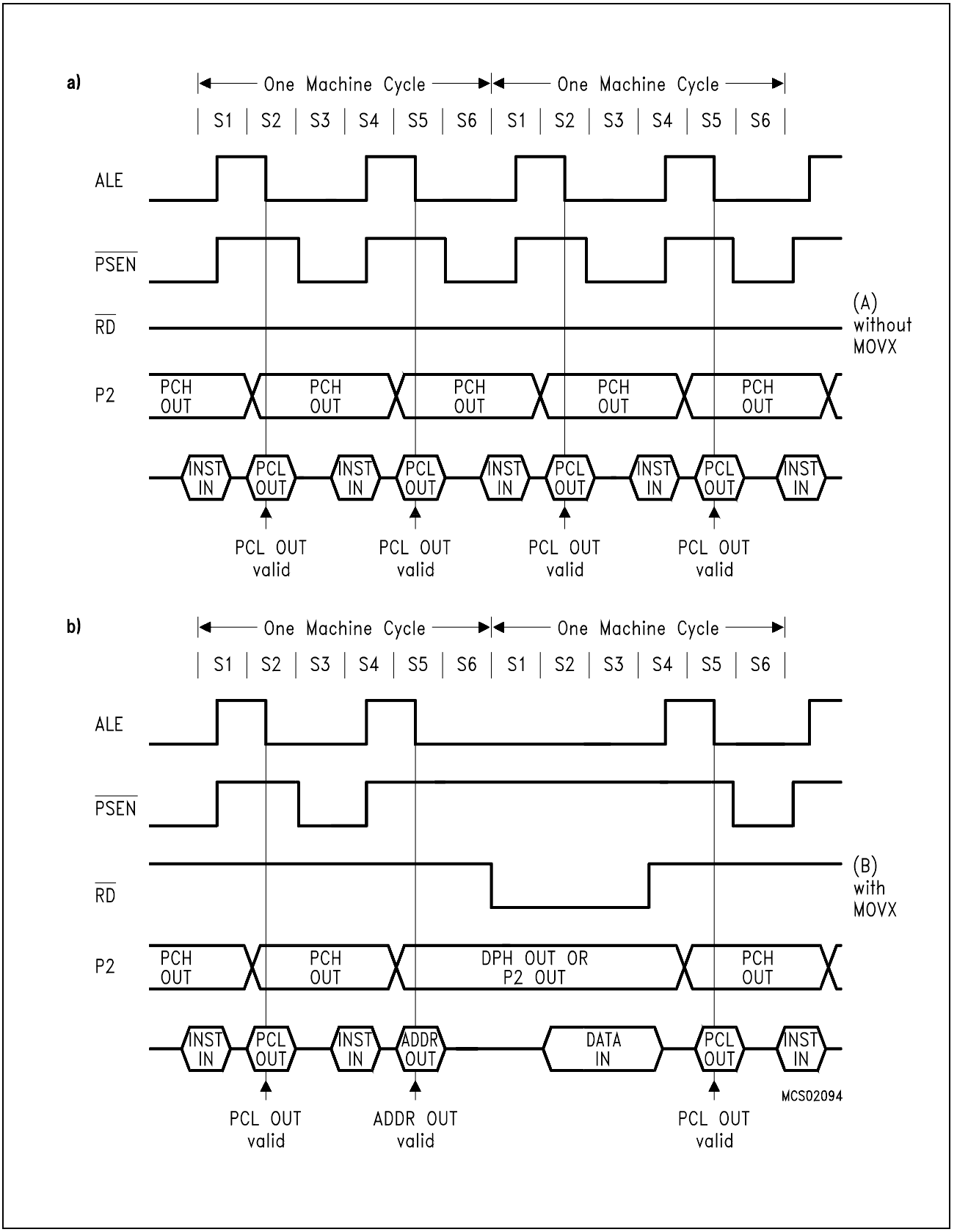


Figure 4-11
External Program Memory Execution

4.1.2 Timing

The timing of the external bus interface, in particular the relationship between the control signals ALE, $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$ and information on port 0 and port 2, is illustrated in **figure 4-11 a) and b)**.

Data memory: in a write cycle, the data byte to be written appears on port 0 just before WR is activated and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at port 0 before the read strobe is deactivated.

Program memory: Signal $\overline{PSEN}$ functions as a read strobe.

4.1.3 External Program Memory Access

The external program memory is accessed under two conditions:

- whenever signal $\overline{EA}$ is active or
- whenever $\overline{EA}$ is high and the program counter (PC) contains an address that is higher than the internal ROM size.

This requires ROM-less versions of the SAB-C511/513 family components to have $\overline{EA}$ wired to ground to allow the program to be fetched from external memory only.

When the CPU is executing out of external program memory, all 8 bits of port 2 are dedicated to an address output function and may not be used for general-purpose I/O. The contents of the port 2 SFR however is not affected. During external program memory fetches port 2 lines output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @DPTR or a MOVX @Ri).

4.2 $\overline{PSEN}$ - Program Store Enable

The read strobe for external fetches is $\overline{PSEN}$. $\overline{PSEN}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{PSEN}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as for $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 12 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$ takes 6 oscillator periods. The execution sequence for these two types of read cycles is shown in **figure 4-11 a) and b)**.

4.3 ALE - Address Latch Enable

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 into an external latch during fetches from external memory. The address byte is valid at the negative transition of ALE. For that purpose, ALE is activated twice every machine cycle. This activation takes place even if the cycle involves no external fetch. The only time no ALE pulse comes out is during an access to external data memory when $\overline{RD}$/$\overline{WR}$ signals are active. The first ALE of the second cycle of a MOVX instruction is missing (see **figure 4-11 b**). Consequently, in any system that does not use data memory, ALE is activated at a constant rate of 1/6 of the oscillator frequency and can be used for external clocking or timing purposes. In systems that do not use external memory at all and do not use ALE as clock, external ALE generation can be suppressed by resetting the EALE bit in the SYSCON register. This can help to reduce system RFI. Because ALE

can be enabled/disabled dynamically, it is also possible to enable ALE only when external memory is accessed. This can be useful, if the external memory is accessed very seldom only.

4.4 XRAM Access Enable (SAB-C513A/A-H only)

The SAB-C513A/A-H maps 256 bytes of the external data space into the on-chip XRAM. Especially when using the 8-bit addressing modes this could prevent access to the external memory extension and might induce problems when porting software. Therefore it is possible to enable and disable the on-chip XRAM. When the XRAM is disabled (default after reset) all accesses will go to the external memory/IO.

Special Function Registers SYSCON (Address $B1_H$)

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$B1_H$	1	0	EALE	–	0	–	–	XMAP	SYSCON

Bit	Function
–	Not implemented. Reserved for future use.
7, 6, 3	Reserved bits; these bits must be always written with the values shown above.
XMAP	Enable XRAM (SAB-C513A/A-H only). XMAP=0 : XRAM disabled. XMAP=1 : XRAM enabled. Note: This bit is don't care for the other members of the SAB-C511/513 family, but should be written with "0" for compatibility reasons when writing to the SYSCON register. When reading the bit in non-SAB-C513A versions, it will be undefined.
EALE	Enable ALE generation. If EALE=0, ALE signal will not be generated. If EALE=1, ALE signal will be generated.

Reset Value (C513/513A/A-H) : $101X0XX0_B$
Reset Value (C511/C511A/C513) : $101X0XXX_B$

4.5 Overlapping External Data and Program Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is used for storing data. In the SAB-C511/C513 the external program and data memory spaces can be combined by AND-ing $\overline{PSEN}$ and $\overline{RD}$. A positive logic AND of these two signals produces an active low read strobe that can be used for the combined physical memory. Since the $\overline{PSEN}$ cycle is faster than the $\overline{RD}$ cycle, the external memory needs to be fast enough to adapt to the $\overline{PSEN}$ cycle.

5 System Reset

5.1 Hardware Reset

The hardware reset function built in the SAB-C511/513 microcontrollers allows for an easy automatic start-up at a minimum of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation in order to restart the device. This is particularly done when the power-down mode is to be terminated (see power-down description **chapter 8**).

The reset input is an active high input. An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the RESET pin must be held high for at least two machine cycles (12 oscillator periods) while the oscillator is running. With the oscillator running the internal reset is executed during the second machine cycle and is repeated every cycle until RESET goes low again.

During RESET active, the pins ALE and $\overline{\text{PSEN}}$ are configured as inputs and should not be active driven externally. An external stimulation at these lines during reset activates several test modes which are reserved for test purposes. This in turn may cause unpredictable output operations at several port pins.

At the RESET pin a pulldown resistor is internally connected to V_{SS} to allow a power-up reset operation with an external capacitor only. An automatic reset can be obtained when power supply is applied by connecting the reset pin via an external capacitor to V_{CC}. After the power supply has been turned on, the capacitor must hold the voltage level at the reset pin for a specified time to effect a complete reset.

A correct reset leaves the processor in a defined state. The program execution starts at location 0000_H. After reset is internally accomplished the port latches of ports 0, 1, 2 and 3 default in FF_H. This leaves port 0 floating, since it is an open drain port when not used as data/address bus. All other I/O port lines (ports 1, 2 and 3) output a one (1).

The contents of the internal RAM (conventional and XRAM) of the SAB-C511/513 is not affected by a reset. After power-up the contents is undefined, while it remains unchanged during a reset if the power supply is not turned off.

Note:

For the SAB-C513A-H (EEPROM version) the RESET signal has to be activated for at least 10 ms if power is applied to the device. The reason for this is that the reference voltage generator of the EEPROM device needs some time to come up from power-down state. In the power-on behaviour there are no differences between the EEPROM and ROM versions.

This reset behaviour of the EEPROM version has to be taken into account for systems that also will uses ROM versions of the SAB-C511/513 family and that use its software power down features.

5.2 Hardware Reset Timing

This section describes the timing of the hardware reset signal.

The input pin RESET is sampled once during each machine cycle. This happens in state 5 phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (low level) the internal reset procedure is started. It needs two complete machine cycles to put the complete device to its correct reset state, i.e. all special function registers contain their default values, the port latches contain 1's etc. The RESET signal must be active for at least two machine cycles; after this time the SAB-C511/513 remains in its reset state as long as the signal is active. When the signal goes inactive this transition is recognized in the following state 5 phase 2 of the machine cycle. Then the processor starts its address output (when configured for external ROM) in the following state 5 phase 1. One phase later (state 5 phase 2) the first falling edge at pin ALE occurs.

Figure 5-12 shows this timing for a configuration with $\overline{EA}$ = 0 (external program memory). Thus, between the release of the RESET signal and the first falling edge at ALE there is a time period of at least one machine cycle but less than two machine cycles.

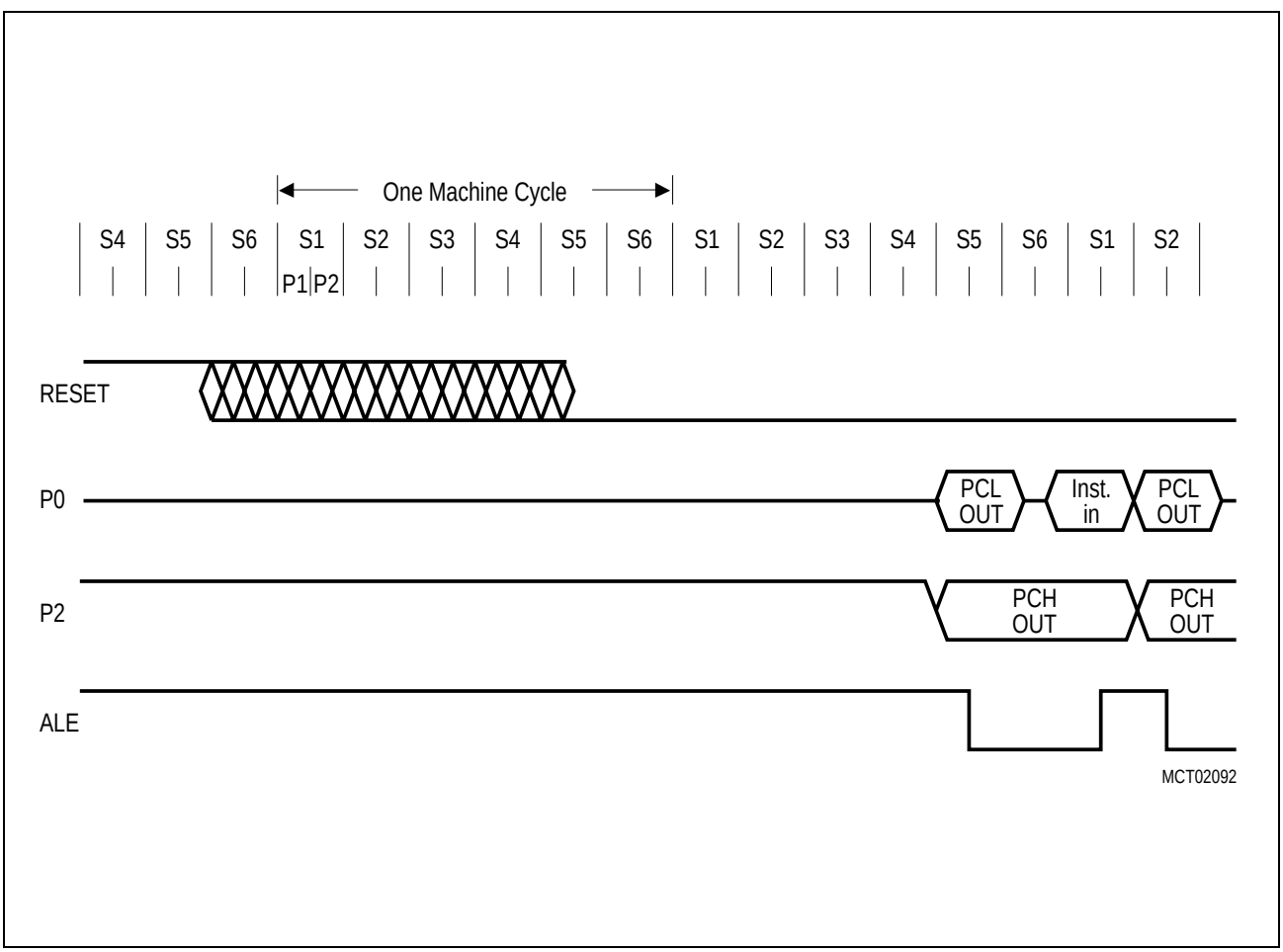


Figure 5-12
CPU Timing after Reset

6 On-Chip Peripheral Components

6.1 Parallel I/O

The SAB-C511/513 has four 8-bit I/O ports. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 3 are quasi-bidirectional I/O ports with internal pullup resistors. That means, when configured as inputs, ports 1 to 3 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

The output drivers of port 0 and 2 and the input buffers of port 0 are also used for accessing external memory. In this application, port 0 outputs the low byte of the external memory address, time multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the port 2 pins continue emitting the P2 SFR contents. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET.

Port 1 pins used for SSC outputs are true push-pull outputs. When used as SSC inputs they float (no pull-up).

6.1.1 Port Structures

Each port bit consists of a latch, an output driver(s) and an input buffer. Read and write accesses to the I/O ports P0, P1, P2 and P3 are performed via the corresponding special function registers.

Figure 6-13 shows a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the four I/O-ports. The bit latch (one bit in the port's SFR) is represented as a type-D flip-flop, which will clock in a value from the internal bus in response to a "write-to-latch" signal from the CPU. Both the output of the latch as well as the actual state of the port pins can be read, depending on the instruction used for accessing the port.

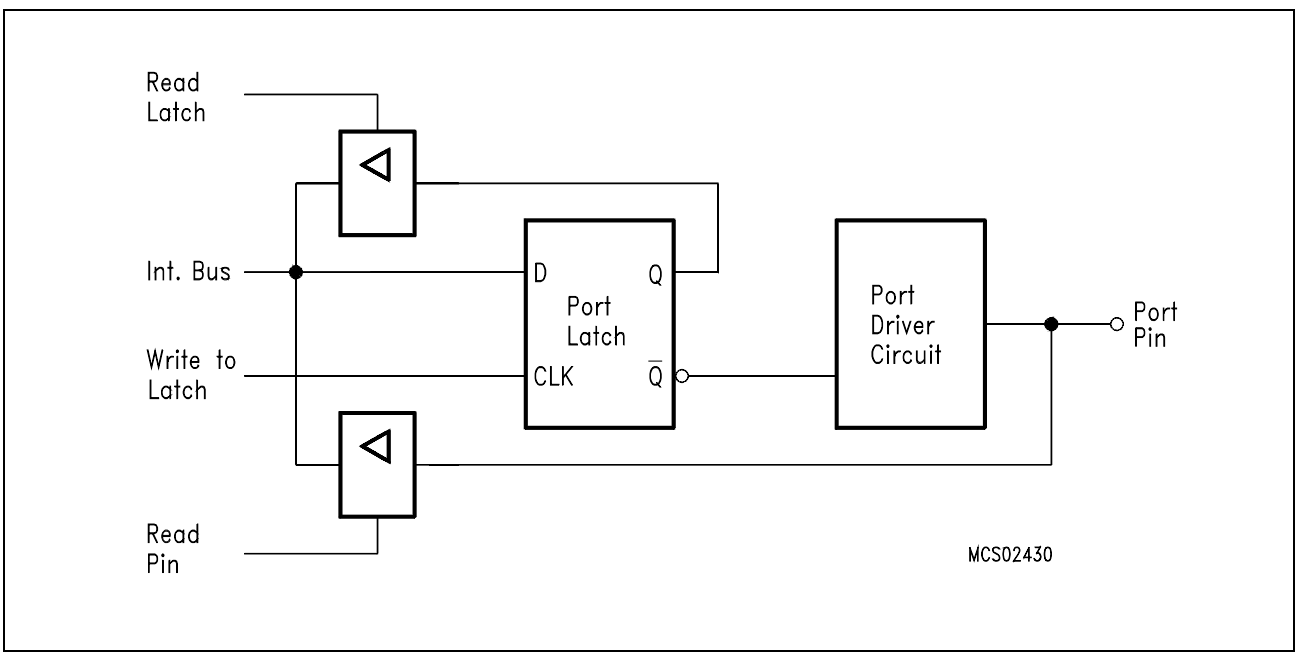


Figure 6-13
Basic Structure of a Port Circuitry

Port 1, 2 and 3 output drivers have internal pullup FET's (see **figure 6-6-14**). Each I/O line can be used independently as an input or output. To be used as an input, the port bit must contain a one (1) (that means for **figure 6-2**: $\overline{Q}$=0), which turns off the output driver FET n1. Then, for ports 1, 2 and 3, the pin is pulled high by the internal pullups, but can be pulled low by an external source. When externally pulled low the port pins source current (I_{IL} or I_{TL}). For this reason these ports are sometimes called "quasi-bidirectional".

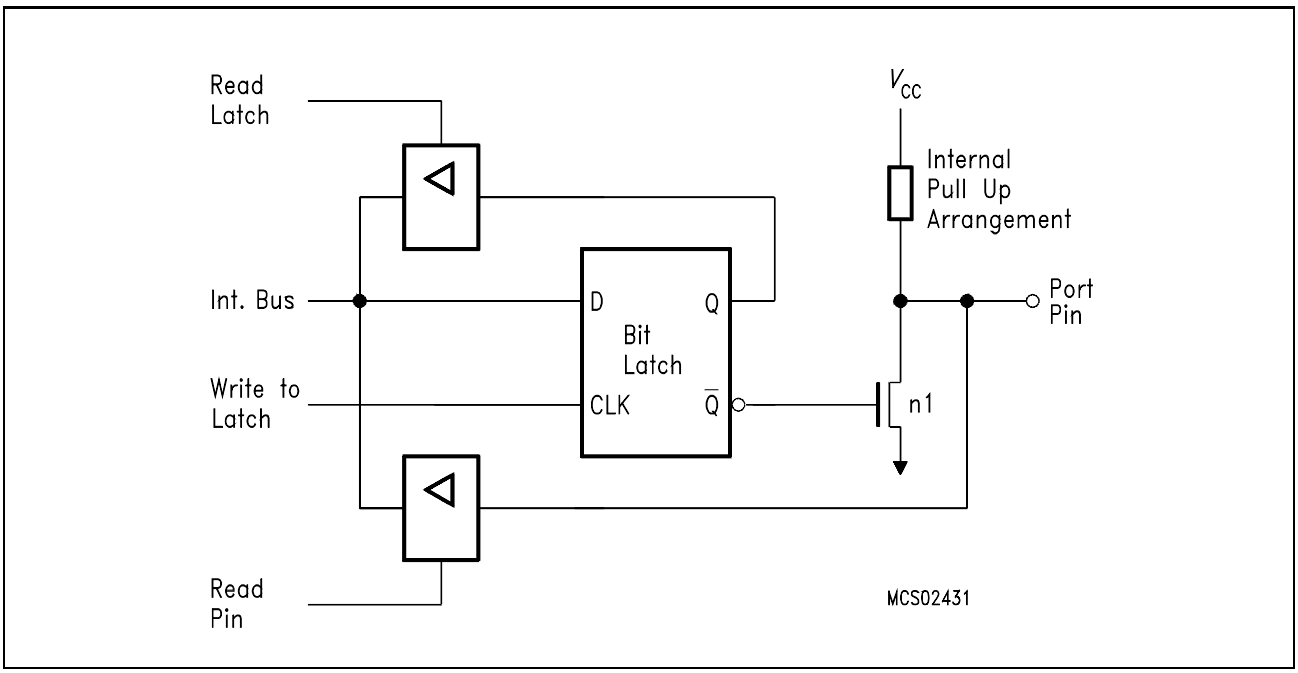


Figure 6-14
Basic Output Driver Circuit of Ports 1, 2 and 3

In fact, the pullups mentioned before and included in **figure 6-2** are pullup arrangements shown in **figure 6-6-15a**. One n-channel pulldown FET and three pullup FETs are used:

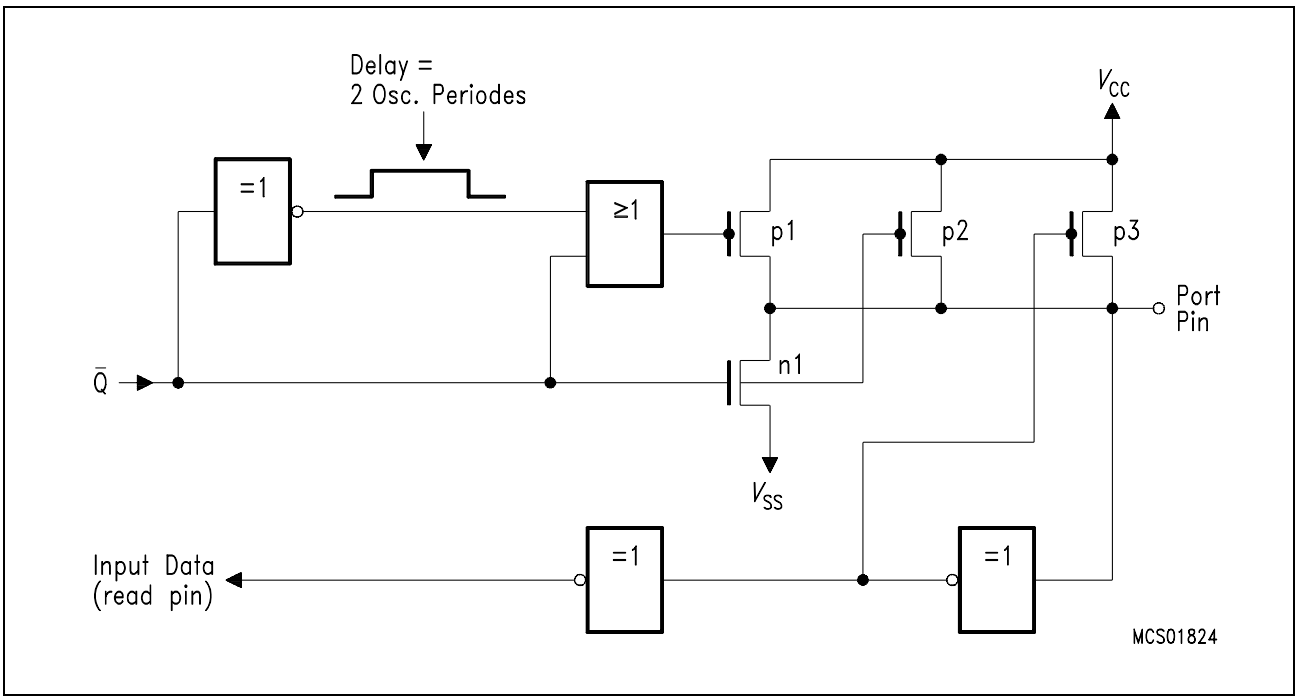


Figure 6-15a
Output Driver Circuit of Ports 1, 2 and 3 (except P1.2, P1.3, P1.4 and P1.5)

- The **pulldown FET n1** is of n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a "0" is programmed to the port pin. A short circuit to V_{CC} must be avoided if the transistor is turned on, since the high current might destroy the FET. This also means that no ”0“ must be programmed into the latch of a pin that is used as input.
- The **pullup FET p1** is of p-channel type. It is activated for two oscillator periods (S1P1 and S1P2) if a 0-to-1 transition is programmed to the port pin, i.e. a "1" is programmed to the port latch which contained a "0". The extra pullup can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.
- The **pullup FET p2** is of p-channel type. It is always activated when a "1" is in the port latch, thus providing the logic high output level. This pullup FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.
- The **pullup FET p3** is of p-channel type. It is only activated if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pullup current if a logic high level shall be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level, e.g when used as input. In this configuration only the weak pullup FET p2 is active, which sources the current I_{IL} . If, in addition, the pullup FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, an additional power consumption can be avoided if port pins are used as inputs with a low level applied. However, the driving capability is stronger if a logic high level is output.

The described activating and deactivating of the four different transistors results in four states which can be :

- input low state (IL), p2 active only
- input high state (IH) = steady output high state (SOH), p2 and p3 active
- forced output high state (FOH), p1, p2 and p3 active
- output low state (OL), n1 active

If a pin is used as input and a low level is applied, it will be in IL state, if a high level is applied, it will switch to IH state.

If the latch is loaded with "0", the pin will be in OL state.

If the latch holds a "0" and is loaded with "1", the pin will enter FOH state for two cycles and then switch to SOH state. If the latch holds a "1" and is reloaded with a "1" no state change will occur.

At the beginning of power-on reset the pins will be in IL state (latch is set to "1", voltage level on pin is below of the trip point of p3). Depending on the voltage level and load applied to the pin, it will remain in this state or will switch to IH (=SOH) state.
If it is is used as output, the weak pull-up p2 will pull the voltage level at the pin above p3's trip point after some time and p3 will turn on and provide a strong "1". Note, however, that if the load exceeds the drive capability of p2 (I_{IL}), the pin might remain in the IL state and provide a week "1" until the first 0-to-1 transition on the latch occurs. Until this the output level might stay below the trip point of the external circuitry.

The same is true if a pin is used as bidirectional line and the **external** circuitry is switched from output to input when the pin is held at "0" and the load then exceeds the p2 drive capabilities.

If the load exceeds I_{IL} the pin can be forced to “1“ by writing a “0“ followed by a “1“ to the port pin..

The driver and control structure of the port pins used for the alternate functions of the SSC have been modified to provide the following features:

- P1.2 when used as SSC clock output will become a true push-pull output
- P1.3 when used as SSC receiver input will become an input without pullups.
- P1.4 when used as SSC transmitter output will become a true push-pull output with tristate capability
- P1.5 when used as SSC slave select input will directly control the tristate condition of P1.4.

The modified port structure is illustrated in **figures 6-b** and **6-c**.

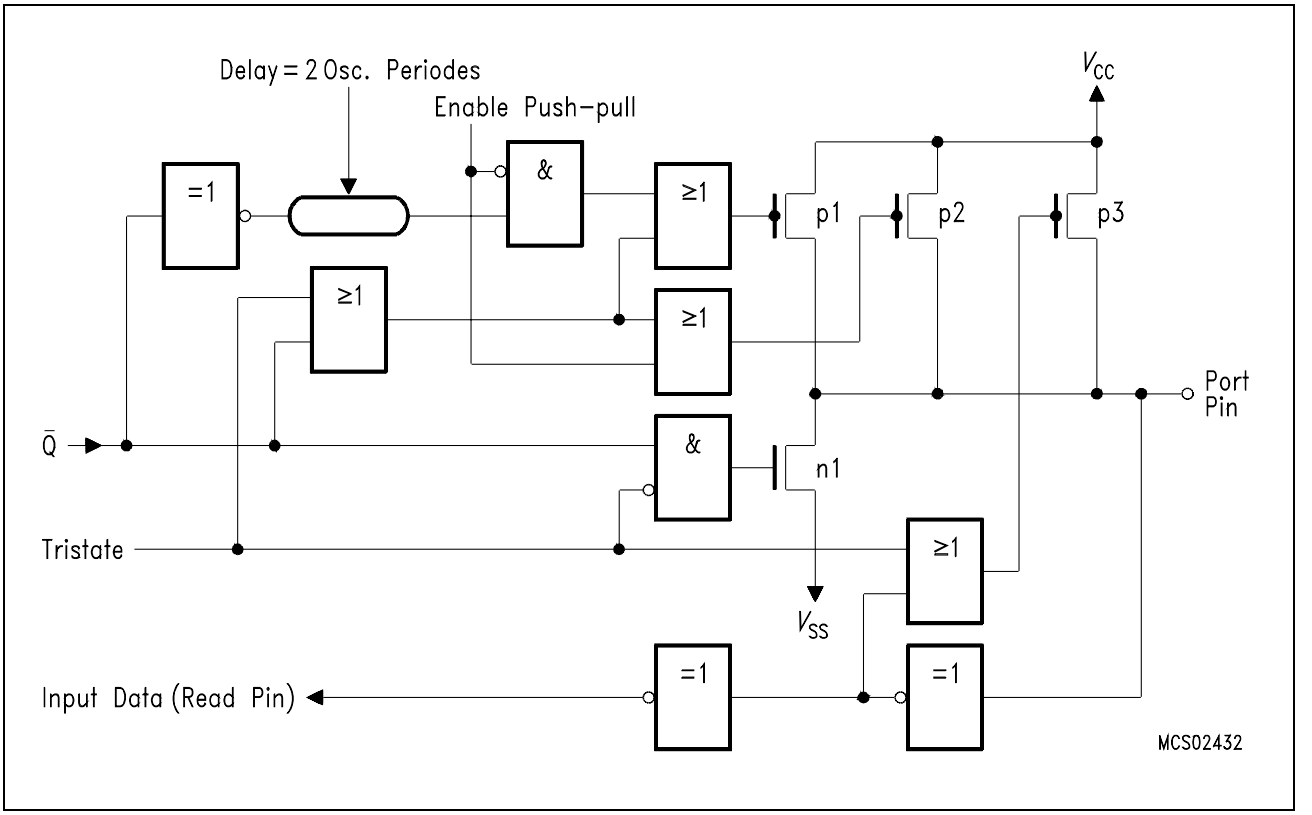


Figure 6-15b
Driver Circuit of Port 1 pins P1.2 and P1.4 (when used for SLCK and STO)

Pin Control for SCLK

When the SSC is disabled, both Enable Push-pull and Tristate will be inactive, the pin behaves like a standard IO pin.

In master mode and with SSC enabled, Enable Push-pull will be active and Tristate will be inactive.

In slave mode and with SSC enabled, Enable Push-pull will be inactive and Tristate will be active.

Pin Control for STO

When the SSC is disabled, both Enable Push-pull and Tristate will be inactive.

In master mode and SSC enabled, Enable Push-pull will be active and Tristate will be inactive.

In slave mode and SSC enabled, Enable Push-pull will be active.
If the transmitter is enabled ($\overline{SLS}$ and TEN active), Tristate will be inactive.
If the transmitter is disabled (either $\overline{SLS}$ or TEN inactive), Tristate will be active.

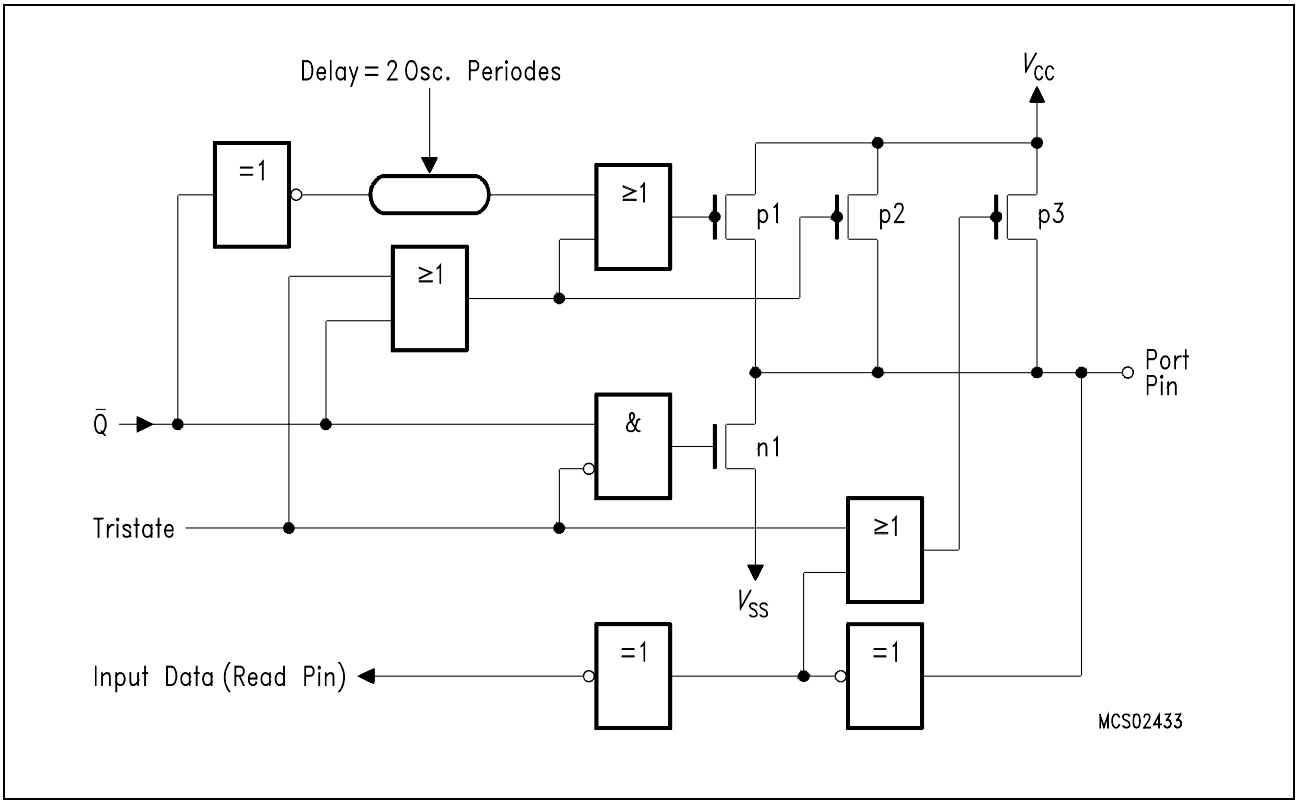


Figure 6-15c
Driver Circuit of Port 1 pins P1.3 and P1.5 (when used for SRI and $\overline{SLS}$)

When enabling the SSC, inputs used for the SSC will be switched into a high-impedance mode.

For P1.3/SRI, Tristate will be enabled, when the SSC is enabled.

For P1.5/$\overline{SLS}$, Tristate will be enabled, when the SSC is enabled and is switched to slave mode. In master mode this pin will remain a regular I/O pin.

Port 0, in contrast to ports 1, 2 and 3, is considered as "true" bidirectional, because the port 0 pins float when configured as inputs. Thus, this port differs in not having internal pullups. The pullup FET in the P0 output driver (see **figure 6-16a**) is used only when the port is emitting 1 s during the external memory accesses. Otherwise, the pullup is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a "1" to the port latch leaves both output FETs off and the pin floats. In that condition it can be used as high-impedance input. If port 0 is configured as general I/O port and has to emit logic high-level (1), external pullups are required.

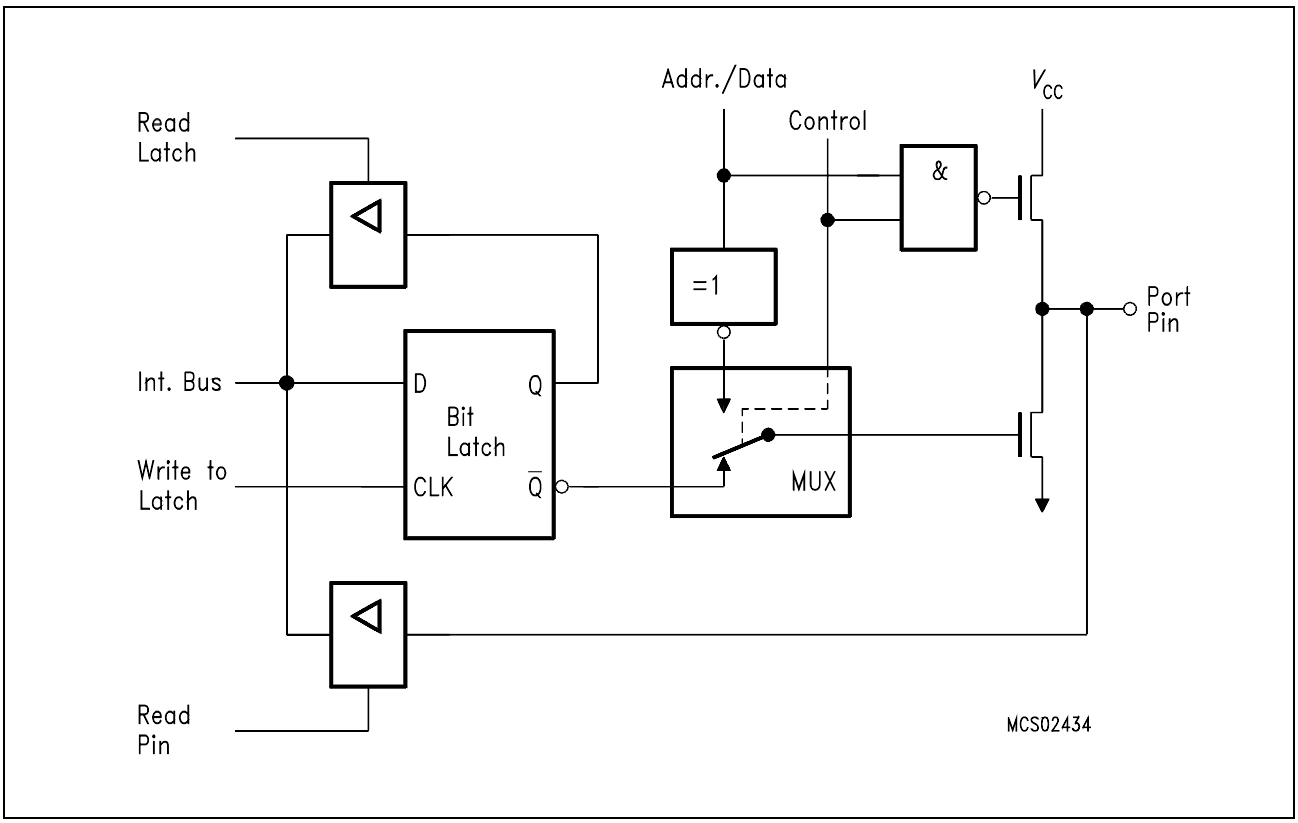


Figure 6-16a
Port 0 Circuitry

6.1.2 Port 0 and Port 2 used as Address/Data Bus

As shown in **figure 6-16a** and below in **figure 6-16b**, the output drivers of ports 0 and 2 can be switched to an internal address or address/data bus for use in external memory accesses. In this application they cannot be used as general purpose I/O, even if not all address lines are used externally. The switching is done by an internal control signal dependent on the input level at the $\overline{EA}$ pin and/or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P2 SFR remains unchanged while the P0 SFR has 1's written to it. Being an address/data bus, port 0 uses a pullup FET as shown in **figure 6-16a**. When a 16-bit address is used, port 2 uses the additional strong pullups p1 to emit 1's for the entire external memory cycle instead of the weak ones (p2 and p3) used during normal port activity.

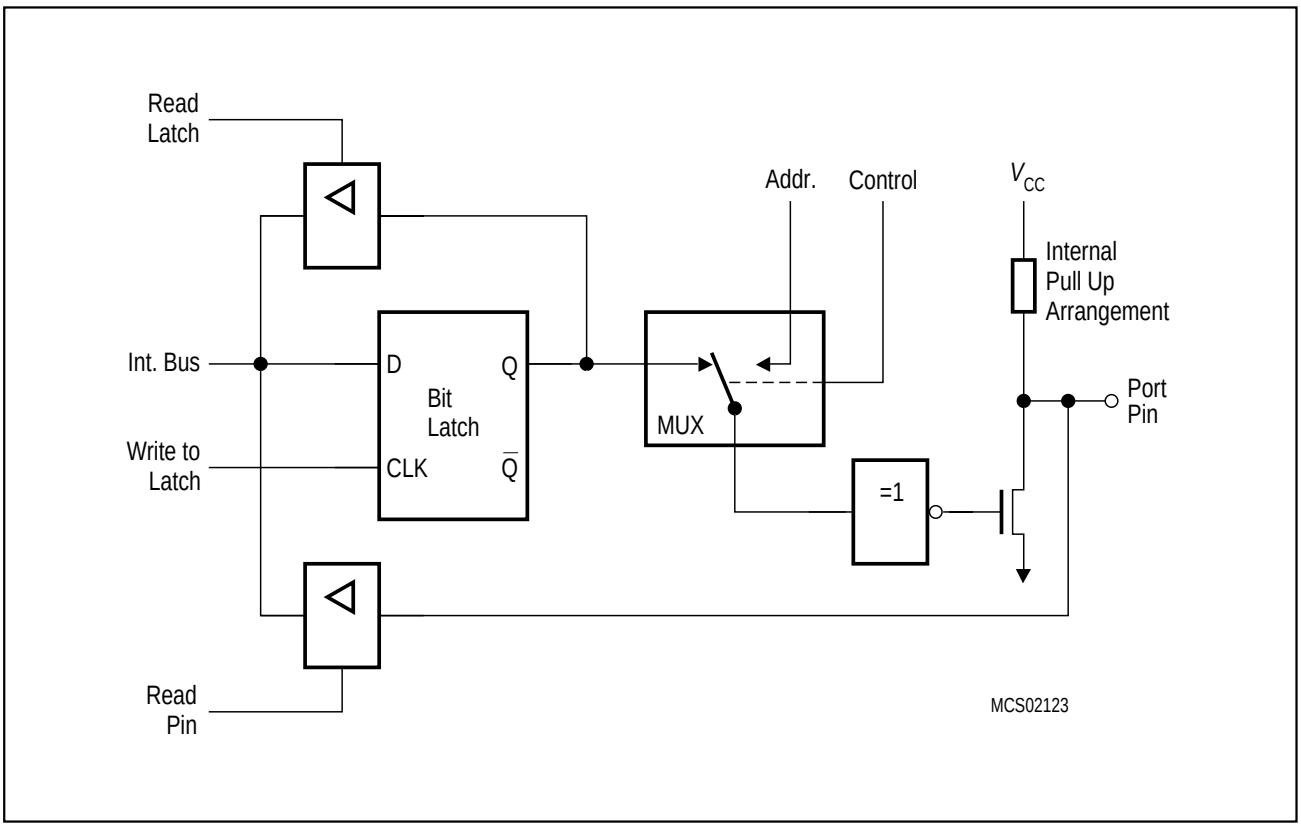


Figure 6-16b
Port 2 Circuitry

6.1.3 Alternate Functions

The pins of ports 1 and 3 are multifunctional. They are port pins and also serve to implement alternate functions (special inputs/outputs for on-chip peripherals) as listed in **table 6-6-3**.

Figure 6-17a shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise the pulldown FET is on and the port pin is stuck at 0. After reset all port latches contain ones (1).

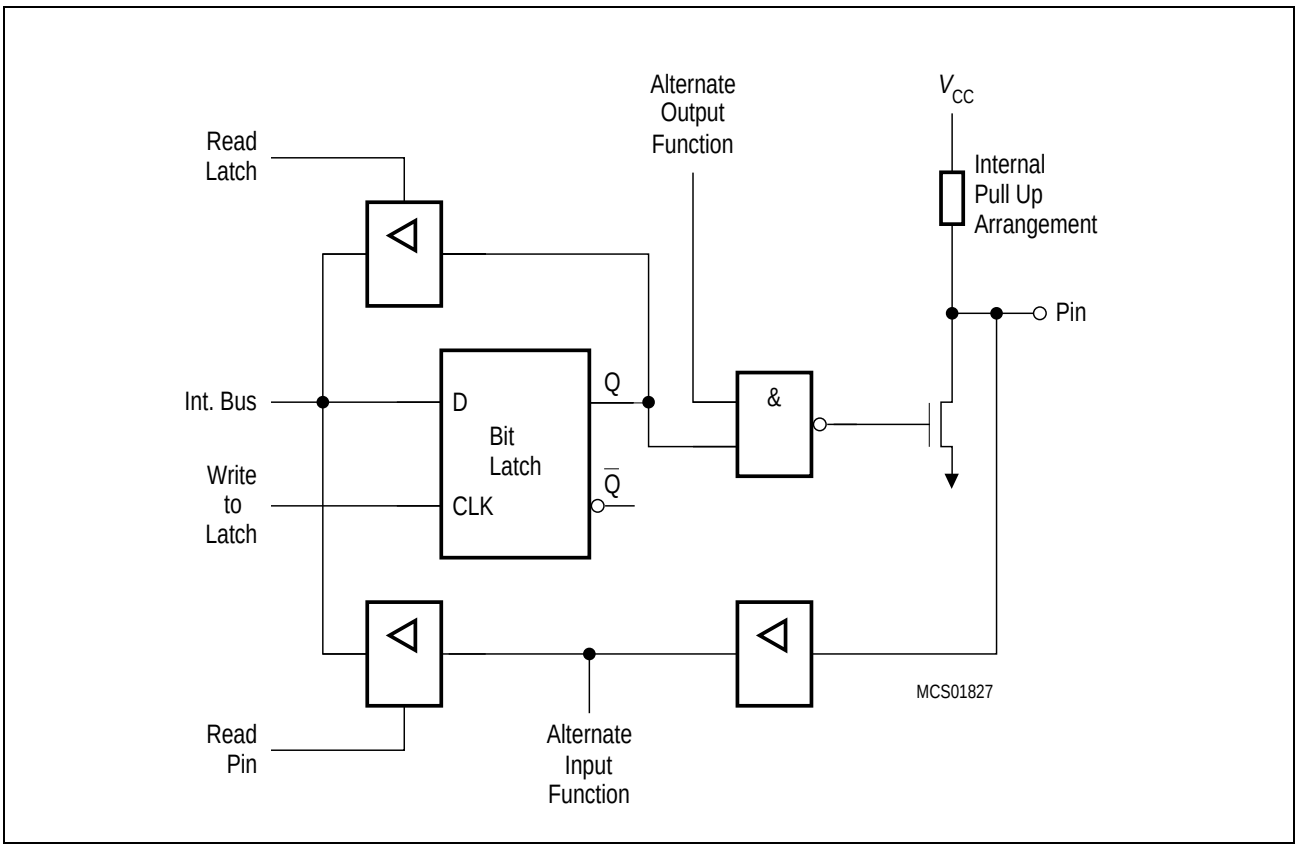


Figure 6-17a
Ports 1 and 3

For port pins P1.2 to P1.5 different structures apply, see **figures 6-17b** and **6-17c**.

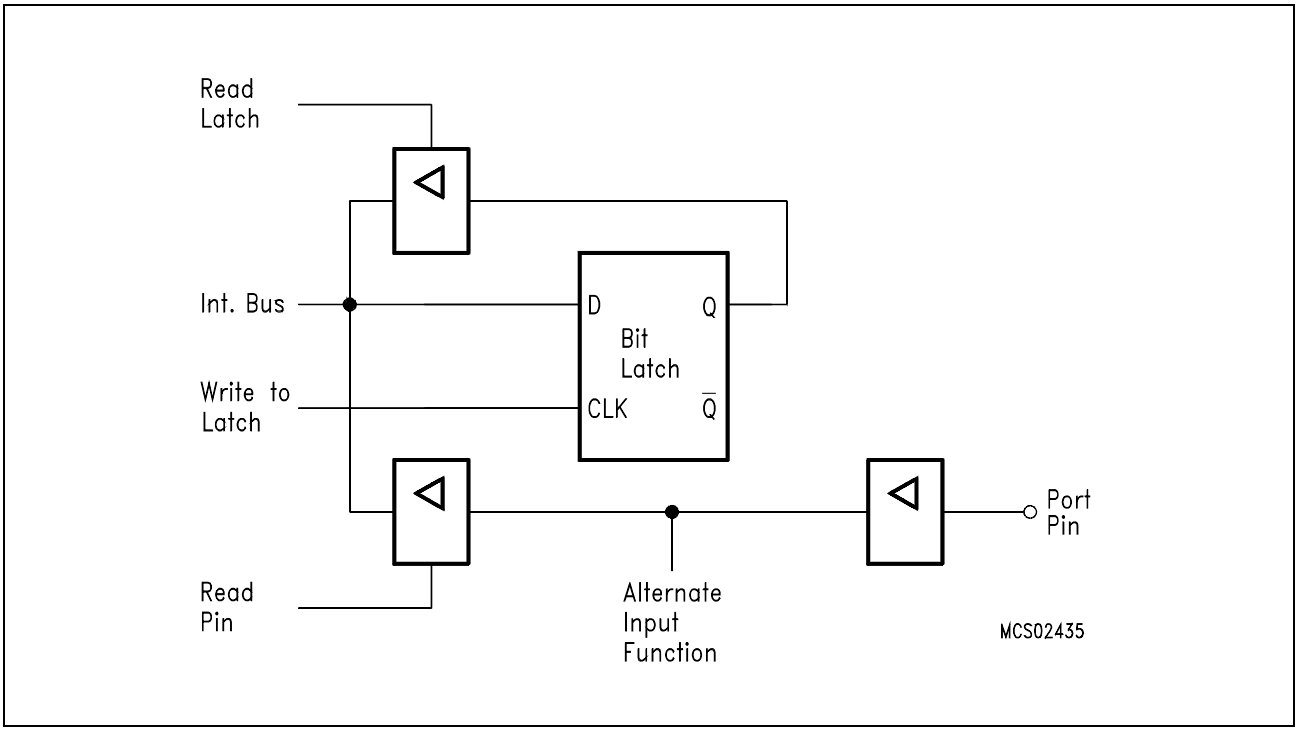


Figure 6-17b
Port pins P1.2, P1.3 and P1.5 (when used as inputs to SSC)

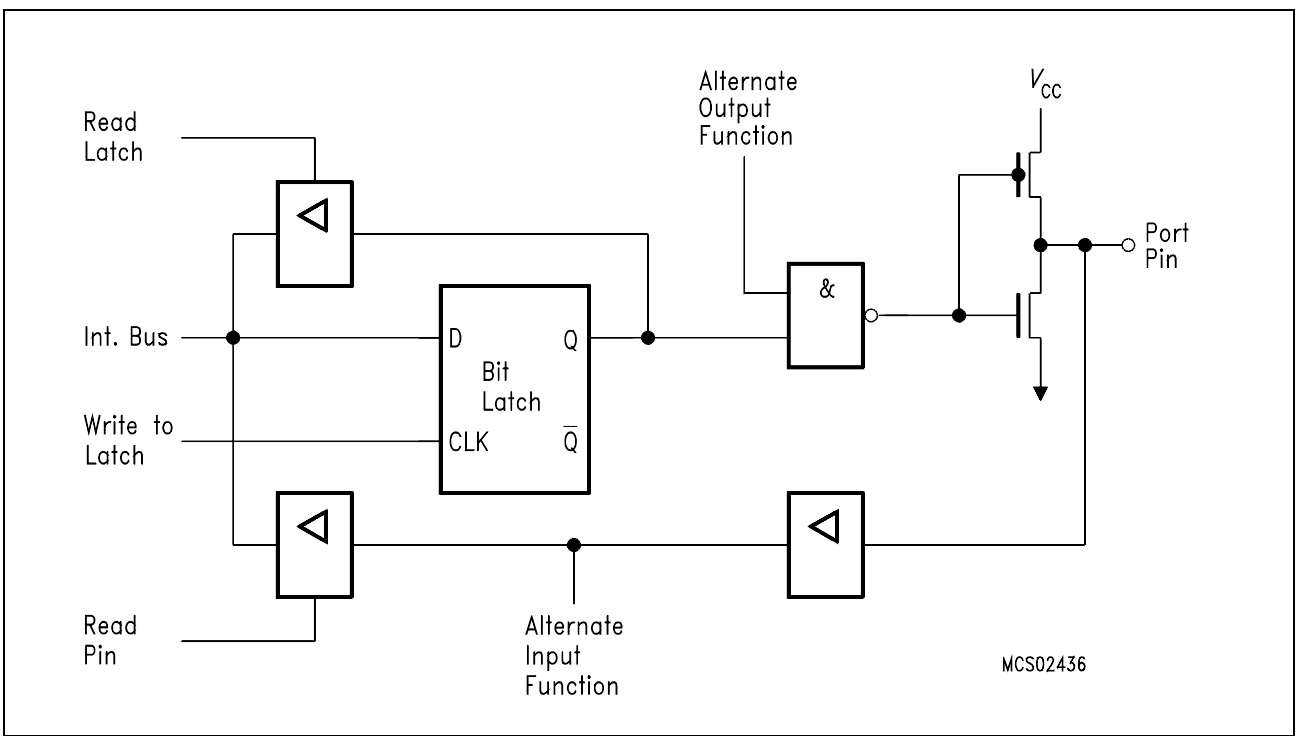


Figure 6-17c
Port pins P1.2 and P1.4 (when used as outputs by SSC)

Ports 1 and 3 provide several alternate functions as listed in **table 6-3**:

Table 6-3
Alternate Functions of Port 1 and 3

Port	Symbol	Function
P1.0	T2	Input to counter 2 (SAB-C513/513A/C513A-H only)
P1.1	T2EX	Capture-reload trigger of timer 2 / up-down count (SAB-C513/513A/C513A-H only)
P1.2	SCLK	SSC master clock output, slave clock input
P1.3	SRI	SSC serial data in
P1.4	STO	SSC serial data out
P1.5	$\overline{\text{SLS}}$	SSC slave select
P3.0	RXD	Serial port's receiver data input (asynchronous) or data input/output (synchronous, not available at SAB-C511/C511A)
P3.1	TXD	Serial port's transmitter data output (asynchronous) or data clock output (synchronous, not available at SAB-C511/C511A)
P3.2	$\overline{\text{INT0}}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe

6.1.4 Port Handling

6.1.4.1 Port Timing

When executing an instruction that changes the value of a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are only sampled by their output buffers during phase 1 of any clock period (during phase 2 the output buffer holds the value it noticed during the previous phase 1). Consequently, the new value in the port latch will not appear at the output pin until the next phase 1, which will be at S1P1 of the next machine cycle.

When an instruction reads a value from a port pin (e.g. MOV A, P1) the port pin is actually sampled in state 5 phase 1 or phase 2 depending on port and alternate functions. **Figure 6-18** illustrates this port timing. It must be noted that this mechanism of sampling once per machine cycle is also used if a port pin is to detect an "edge", e.g. when used as counter input. In this case an "edge" is detected when the sampled value differs from the value that was sampled the cycle before. Therefore, there must be met certain requirements on the pulse length of signals in order to avoid signal "edges" not being detected. The minimum time period of high and low level is one machine cycle, which guarantees that this logic level is noticed by the port at least once.

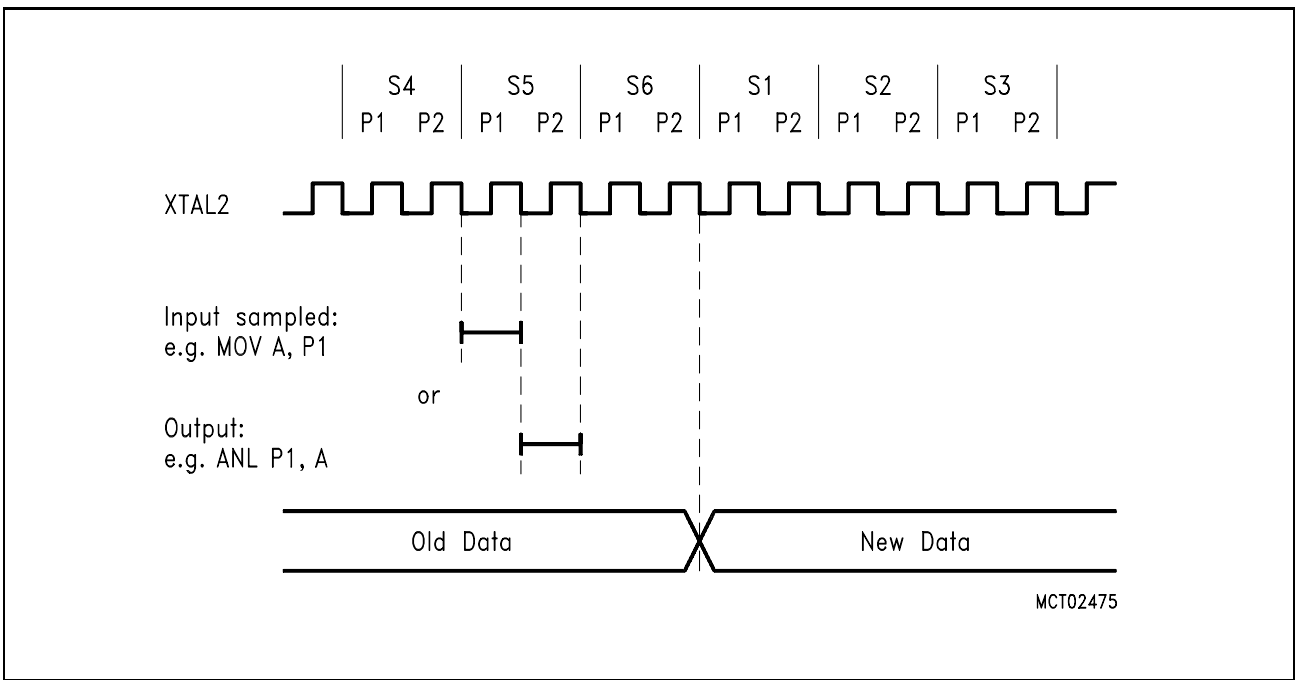


Figure 6-18
Port Timing

6.1.4.2 Port Loading and Interfacing

The output buffers of ports 1, 2 and 3 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can be be seen in the Data Sheet of the SAB-C511/513 family members. The corresponding DC parameters are V_{OL} and V_{OH}.

The same applies to port 0 output buffers. They do, however, require external pullups to drive floating inputs, except when being used as the address/data bus.

When used as inputs it must be noted that the ports 1, 2 and 3 are not floating but have internal pullup transistors. The driving devices must be capable of sinking a sufficient current if a logic low level shall be applied to the port pin (the parameters I_{TL} and I_{IL} in the DC characteristics specify these currents). Port 0 has floating inputs when used for digital input.

6.1.4.3 Read-Modify-Write Feature of Ports 1, 2 and 3

Some port-reading instructions read the latch and others read the pin (see **figure 6-13**). The instructions reading the latch rather than the pin read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write"- instructions, which are listed in **table 6-4**. If the destination is a port or a port pin, these instructions read the latch rather than the pin. Note that all other instructions which can be used to read a port, exclusively read the port pin. In any case, reading from latch or pin, resp., is performed by reading the SFR P0, P1, P2 and P3; for example, "MOV A, P1" reads the value from port 3 pins, while "ANL P1, #0AAH" reads from the latch, modifies the value and writes it back to the latch.

It is not obvious that the last three instructions in **table 6-4** are read-modify-write instructions, but they are. The reason is that they read the port byte, all 8 bits, modify the addressed bit, then write the complete byte back to the latch.

Table 6-4
"Read-Modify-Write"-Instructions

Instruction	Function
ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic exclusive OR; e.g. XRL P3, A
JBC	Jump if bit is set and clear bit; e.g. JBC P1.1, LABEL
CPL	Complement bit; e.g. CPL P3.0
INC	Increment byte; e.g. INC P1
DEC	Decrement byte; e.g. DEC P1
DJNZ	Decrement and jump if not zero; e.g. DJNZ P3, LABEL
MOV Px.y,C	Move carry bit to bit y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

The reason why read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a "1" is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor (approx. 0.7 V, i.e. a logic low level!) and interpret it as "0". For example, when modifying a port bit by a SETB or CLR instruction, another bit in this port with the above mentioned configuration might be changed if the value read from the pin were written back to the latch. However, reading the latch rater than the pin will return the correct value of "1".

6.2 Timers/Counters

The SAB-C511/513 microcontrollers contains two (SAB-C511/C511A) or three (SAB-C513/C513A/C513A-H) 16-bit timers/counters which are useful in many applications for timing and counting functions.

In timer function, the register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the counter rate is 1/12 of the oscillator frequency.

In counter function, the register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 and P3.5, resp.). In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the counter is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 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 must be held for at least one full machine cycle.

6.2.1 Timer/Counter 0 and 1

Timer / counter 0 and 1 of the SAB-C511/513 family components are fully compatible with timer / counter 0 and 1 of the 8051 microcontroller and can be used in the same four operating modes:

Mode 0: 8-bit timer/counter with a divide-by-32 prescaler

Mode 1: 16-bit timer/counter

Mode 2: 8-bit timer/counter with 8-bit auto-reload

Mode 3: Timer/counter 0 is split into one 8-bit timer/counter and one 8-bit timer when programmed to this mode. Timer/counter 1 set to this mode will simply hold its count. The effect is the same as setting TR1 = 0, disabling the counter.

The external inputs $\overline{INT0}$ and $\overline{INT1}$ can be programmed to function as a gate for timer/counters 0 and 1 to facilitate pulse width measurements.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer/counter 0, TH1 and TL1 for timer/counter 1) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD.

In the following descriptions the symbols TH0 and TL0 are used to specify the high-byte and the low-byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are described and shown for timer 0. If not explicity noted, this applies also to timer 1.

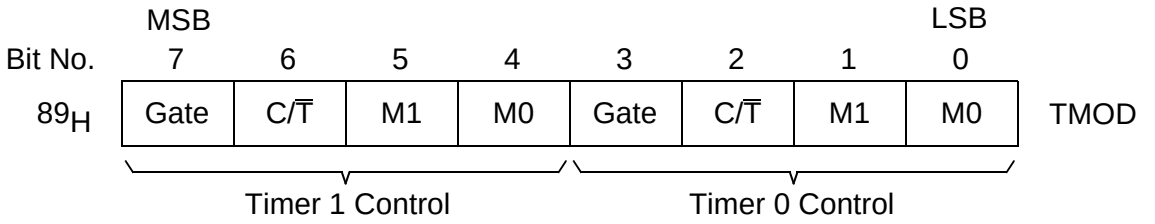
Special Function Register TCON (Address 88_H) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
88_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

These bits are not used in controlling timer/counter 0 and 1.

Bit	Function
TR0	Timer 0 run control bit. Set/cleared by software to turn timer/counter 0 ON/OFF.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 run control bit. Set/cleared by software to turn timer/counter 1 ON/OFF.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

Special Function Register TMOD (Address 89_H) **Reset Value : 00_H**



Bit		Function
Gate		Gating control. When set, timer/counter "x" is enabled only while "INTx" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
C/$\overline{T}$		Counter or timer select bit. Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).
M1	M0	Timer modes select bits
0	0	8-bit timer/counter. "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops

6.2.1.1 Mode 0

Putting either timer/counter 0,1 into mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. **Figure 6-19** shows the mode 0 operation.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1's to all 0's, it sets the timer overflow flag TF0. The overflow flag TF0 then can be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or $\overline{INT0}$ = 1 (setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements). TR0 is a control bit in the special function register TCON; Gate is in TMOD.

The 13-bit register consists of all 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR0, TF0, TH0, TL0 and $\overline{INT0}$ for the corresponding timer 1 signals in **figure 6-19**. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

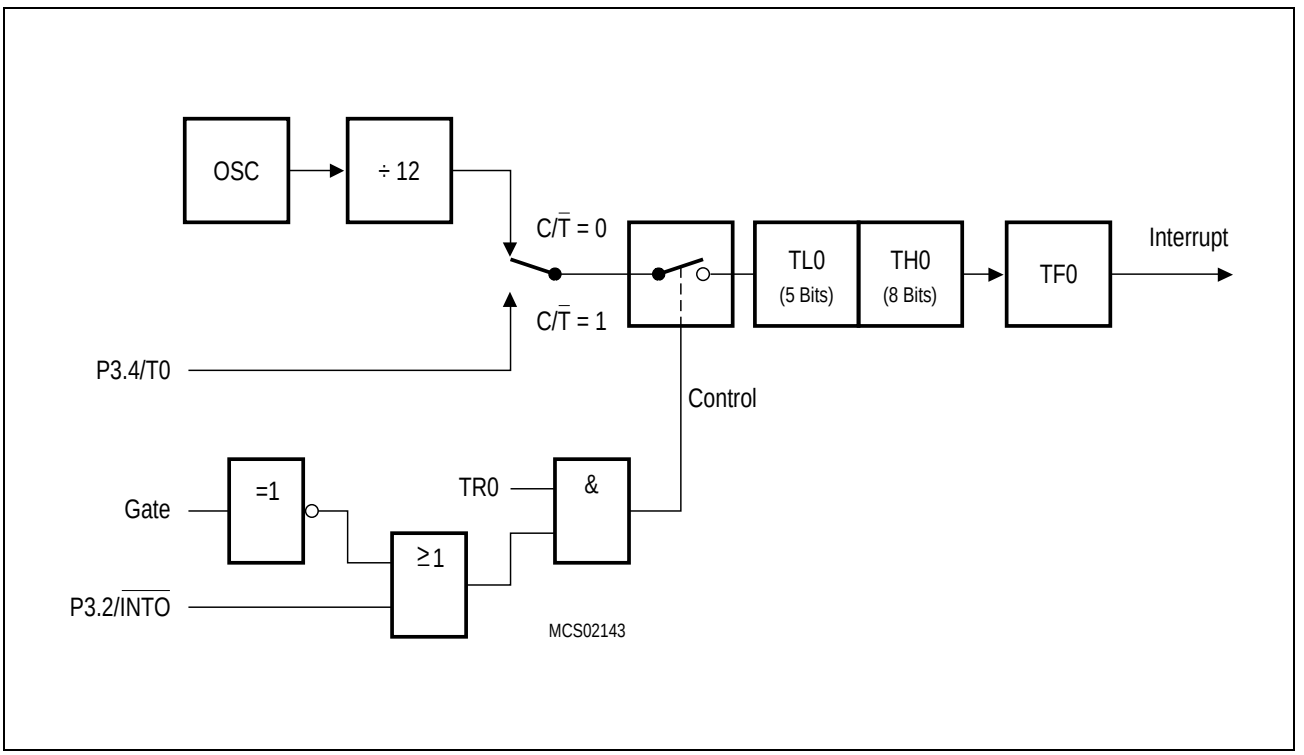


Figure 6-19
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

6.2.1.2 Mode 1

Mode 1 is the same as mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in **figure 6-20**.

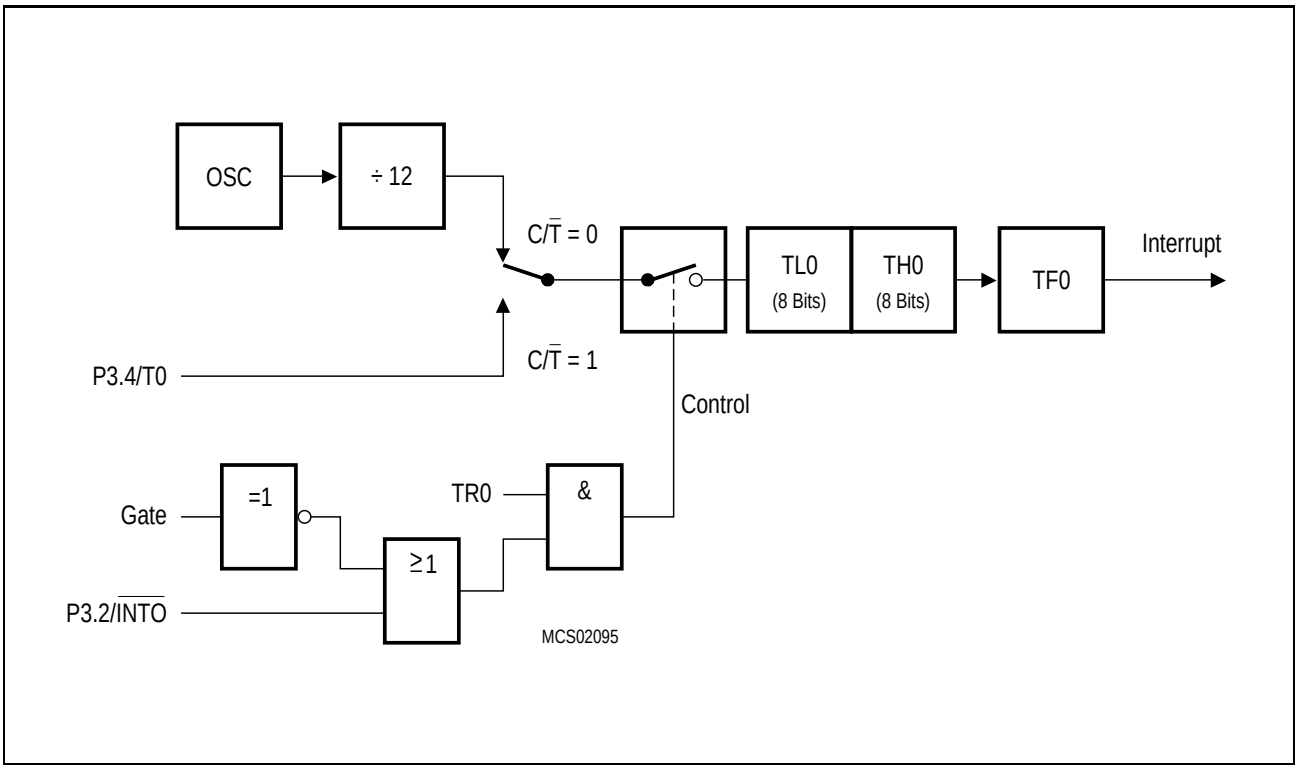


Figure 6-20
Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

6.2.1.3 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in **figure 6-21**. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

Mode 2 operation is the same for timer 0 as for timer 1. Substitute TR0, TF0, TH0, TL0 and $\overline{\text{INT0}}$ for the corresponding timer 1 signals in **figure 6-21**. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

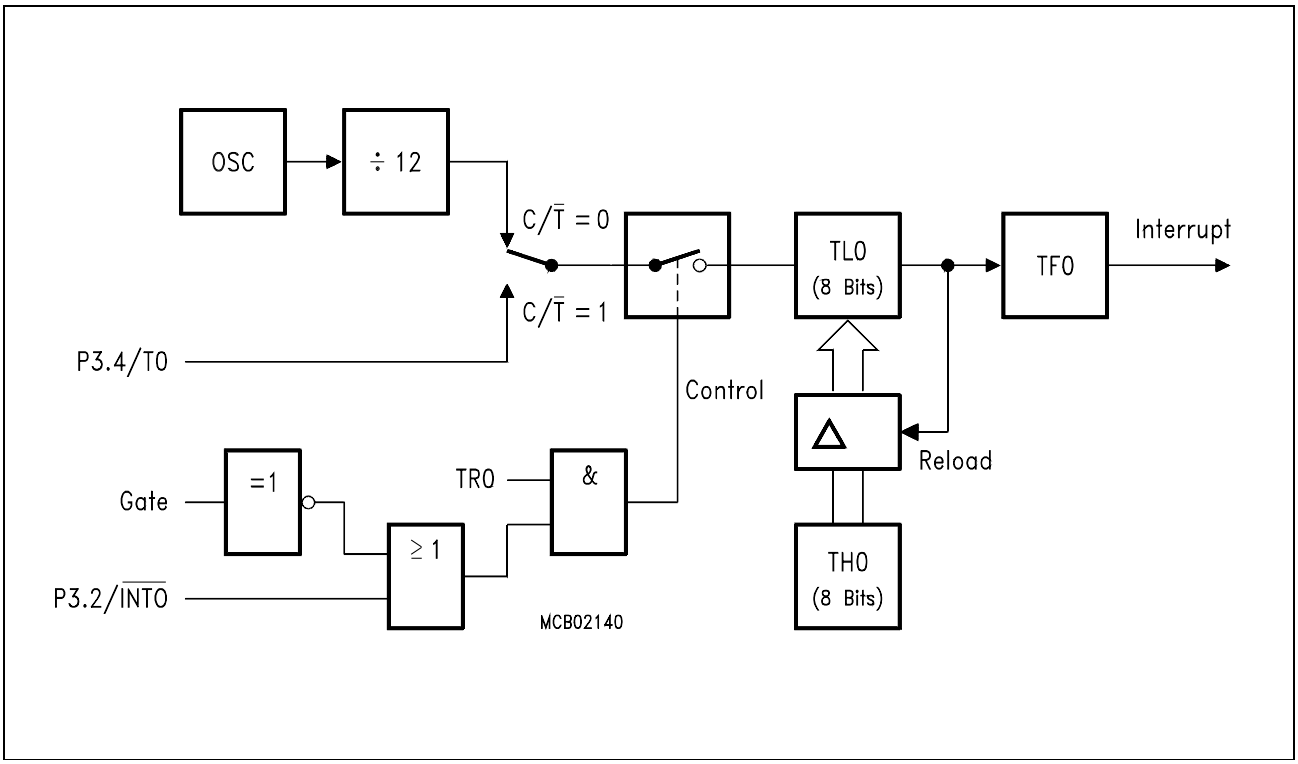


Figure 6-21
Timer/Counter 0, Mode 2: 8-Bit Timer/Counter with Auto-Reload

6.2.1.4 Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count and stops counting. The effect is the same as setting TR1=0. Timer 0 in mode 3 establishes TL0 and TH0 as two seperate counters. The logic for mode 3 on timer 0 is shown in **figure 6-22**. TL0 uses the timer 0 control bits: C/$\overline{T}$, Gate, TR0, $\overline{INT0}$ and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial channel as a baud rate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

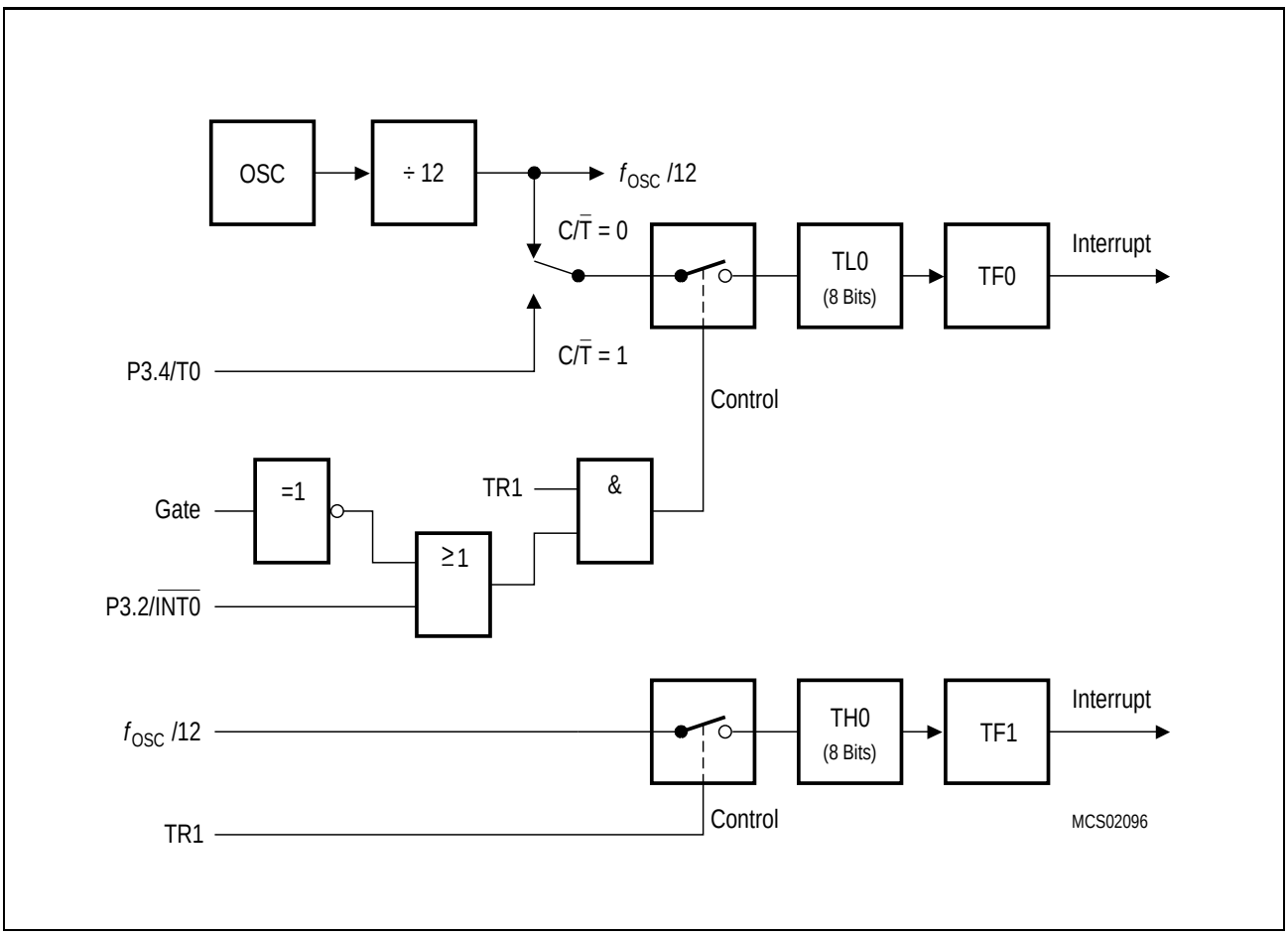


Figure 6-22
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

6.2.2 Timer/Counter 2 (SAB-C513/C513A/C513A-H only)

Timer 2 is a 16-bit timer / counter which can operate as timer or counter. It has 3 operating modes :

- 16-bit auto-reload mode (up or down counting)
- 16-bit capture mode
- Baudrate generator (see **chapter 6.3.3.2**)

The modes are selected by bits in the SFR T2CON ($C8_H$) as shown in **table 6-5**.

Table 6-5
Timer/Counter 2 - Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	Mode
0	0	1	16-bit auto-reload
0	1	1	16-bit capture
1	X	1	Baud rate generator
X	X	0	(OFF)

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.0). In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscilllator frequency. To ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

Special Function Register T2CON (Address $C8_H$) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$C8_H$	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	T2CON

Bit	Function
TF2	Timer 2 Overflow Flag. Set by a timer 2 overflow. Must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 External Flag. Set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1, SFR T2MOD)
RCLK	Receive Clock Enable. When set, causes the serial port to use timer 2 overflow pulses for its receive clock in serial port modes 1 and 3. RCLK = 0 causes timer 1 overflows to be used for the receive clock.
TCLK	Transmit Clock Enable. When set, causes the serial port to use timer 2 overflow pulses for its transmit clock in serial port modes 1 and 3. TCLK = 0 causes timer 1 overflow to be used for the transmit clock.
EXEN2	Timer 2 External Enable. When set, allows a capture or reload to occur as a result of a negative transition on pin T2EX (P1.1) if timer 2 is not being used to clock the serial port. EXEN2 = 0 causes timer 2 to ignore events at T2EX.
TR2	Start / Stop Control for Timer 2. TR2 = 1 starts timer 2.
C/$\overline{T2}$	Timer or Counter Select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture / Reload Select. CP/$\overline{RL2}$ = 1 causes captures to occur an negative transitions at pin T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when timer 2 overflows or negative transitions occur at pin T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on timer 2 overflow.

6.2.2.1 Auto-Reload Mode (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by a bit named DCEN (Down Counter Enable), which is located in SFR T2MOD. When DCEN is set, timer 2 can count up or down depending on the value of pin T2EX (P1.1).

Special Function Register T2MOD (Address $C9_H$) **Reset Value : $XXXXXXX0_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$C9_H$	–	–	–	–	–	–	–	DCEN	T2MOD

Bit	Function
–	Not implemented, reserved for future use.
DCEN	When set, this bit allows timer 2 to be configured as an up/down counter. With T2EX=0 down-counting is selected, with DCEN=1 up-counting is enabled.

Figure 6-23 shows timer 2 automatically counting up when DCEN=0. In this mode there are two options selectable by bit EXEN2 in SFR T2CON.

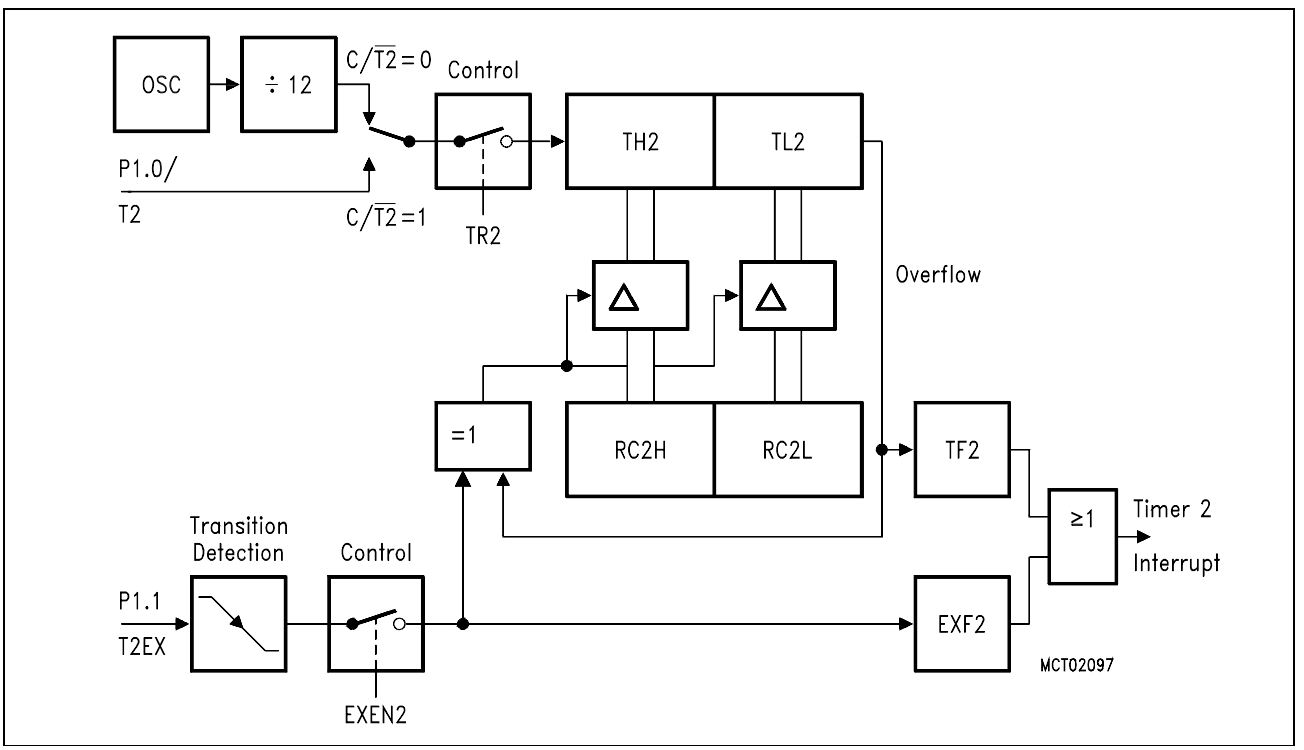


Figure 6-23
Timer 2 Auto-Reload Mode (DCEN = 0)

If EXEN2 = 0, timer 2 counts up to $FFFF_H$ and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RC2H and RC2L. The values in RC2H and RC2L are preset by software.

If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at the external input T2EX (P1.1). This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an timer 2 interrupt if enabled.

Setting the DCEN bit enables timer 2 to count up or down as shown in **figure 6-24**. In this mode the T2EX pin controls the direction of count.

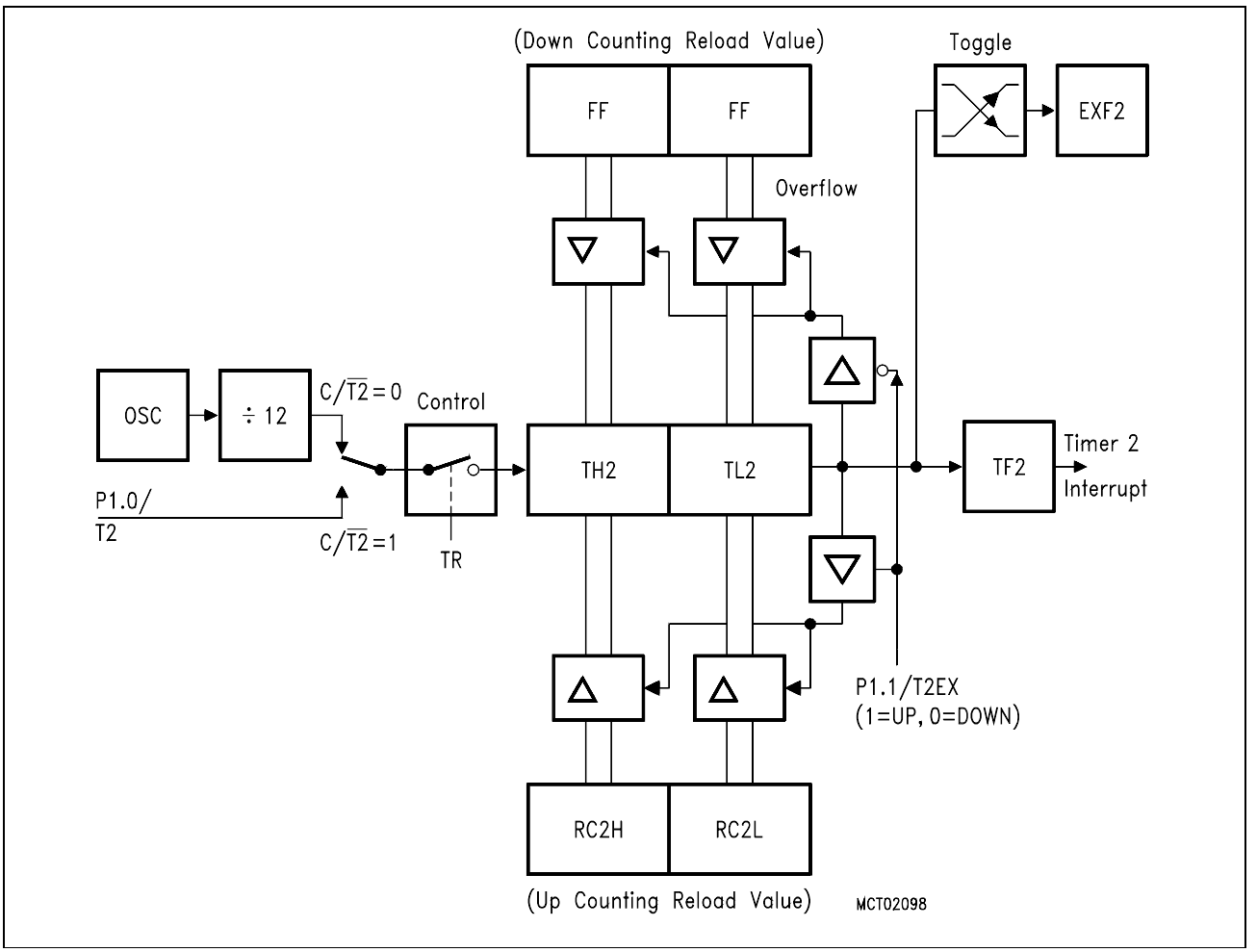


Figure 6-24
Timer 2 Auto-Reload Mode (DCEN = 1)

A logic 1 at T2EX makes timer 2 count up. The timer will overflow at $FFFF_H$ and set the TF2 bit. This overflow also causes the 16-bit value in RC2H and RC2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes timer 2 count down. Now the timer underflows when TH2 and TL2 equal the values stored in RC2H and RC2L. The underflow sets the TF2 bit and causes $FFFF_H$ to be reloaded into the timer registers. The EXF2 bit toggles whenever timer 2 overflows or underflows. This bit can be used as a 17th bit of resolution if desired. In this operating mode, EXF2 does not flag an interrupt.

Note: P1.1/T2EX is sampled during S5P2 of every machine cycle. The next increment/decrement of timer 2 will be done during S3P1 in the next cycle.

6.2.2.2 Capture Mode

In the capture mode there are two options selected by bit EXEN2 in SFR T2CON.

If EXEN2 = 0, timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in SFR T2CON. This bit can be used to generate an interrupt.

If EXEN2 = 1, timer 2 still does the above, but with added feature that a 1-to-0 transition at external input T2EX causes the current value in TH2 and TL2 to be captured into RC2H and RC2L, respectively. In addition, the transition at T2EX causes bit EXF2 in SFR T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in **figure 6-25**.

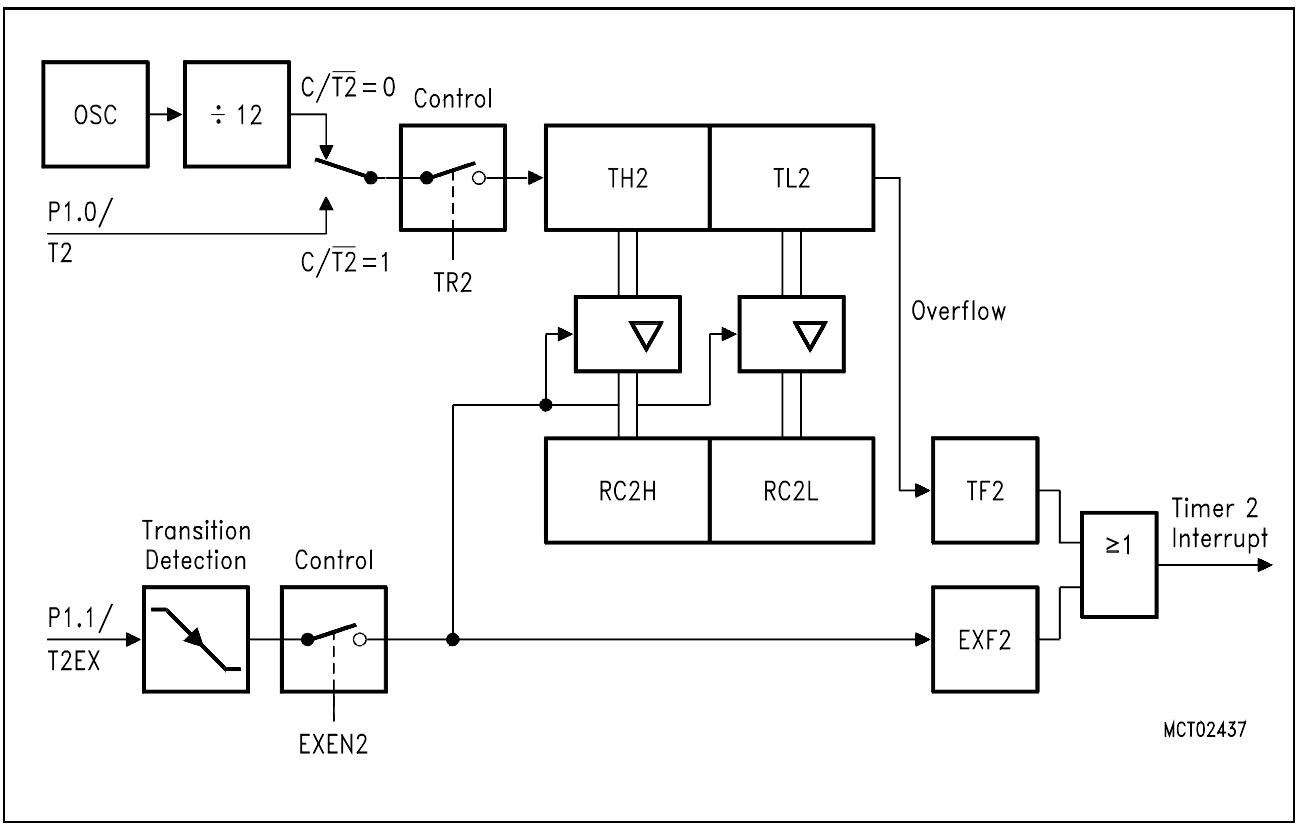


Figure 6-25
Timer 2 in Capture Mode

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1 in SFR T2CON. It will be described in conjunction with the serial port.

6.3 General Purpose Serial Interface USART (SAB-C513/C513AC513A-H only)

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, the first byte will be lost. The serial port receive and transmit registers are both accessed at special function register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes (one synchronous mode, three asynchronous modes):

Mode 0, Shift Register (Synchronous) Mode:

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 data bits are transmitted/ received (LSB first). The baud rate is fixed at $^1/_{12}$ of the oscillator frequency. See section 6.3.4 for more detailed information.

Mode 1, 8-Bit USART, Variable Baud Rate:

10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in special function register SCON. The baud rate is variable. See section 6.3.5 for more detailed information.

Mode 2, 9-Bit USART, Fixed Baud Rate:

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. The baud rate is programmable to either $^1/_{32}$ or $^1/_{64}$ of the oscillator frequency. See section 6.3.6 for more detailed information.

Mode 3, 9-Bit USART, Variable Baud Rate:

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable. See section 6.3.6 for more detailed information.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incomming start bit if REN = 1.

6.3.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th data bit goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is beeing addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the incoming data bytes.

SM2 has no effect in mode 0, and in mode 1 can be used to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

6.3.2 Serial Port Control Register

The serial port control and status register is the special function register SCON. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8) and the serial port interrupt bits (TI and RI).

Special Function Register SCON (Address 98_H) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
98_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

Bit		Function
SM0	SM1	Serial Mode Select Bits
0	0	Serial mode 0: Shift register, fixed baud rate (f_{OSC}/12)
0	1	Serial mode 1: 8-bit UART, variable baud rate
1	0	Serial mode 2: 9-bit UART, fixed baud rate (f_{OSC}/64 or f_{OSC}/32)
1	1	Serial mode 3: 9-bit UART, variable baud rate
SM2		Multiprocessor Communication Select Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN		Receiver Enable Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8		Transmit Bit 8 Is the 9th data bit that will be transmitted in modes 2 and 3. Set or clear by software as desired.
RB8		Receive Bit 8 In modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI		Transmit Interrupt Flag Is transmit interupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI		Receive Interrupt Flag Is receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

6.3.3 Baud Rates

There are several possibilities to generate the baud rate clock for the serial interface depending on the mode in which it is operated.

To clearify the terminology, something should be mentioned about the differences between "baud rate clock" and "baud rate".

The serial interface requires a clock rate which is 16 times the baud rate for the internal synchronization. Therefore, the baud rate generators have to provide a "baud rate clock" to the serial interface which - there divided by 16 - results in the actual "baud rate".

However, all formulas given in the following chapter already includes the factor and calculate the final baud rate.

Mode 0

The baud rate in mode 0 is fixed:

Mode 0 baud rate = oscillator frequency/12 = $f_{OSC}/12$

Mode 2

The baud rate in mode 2 depends on the value of bit SMOD in special function register PCON (87_H). If SMOD = 0 (which is the value on reset), the baud rate is $f_{OSC}/64$. If SMOD = 1, the baud rate is $f_{OSC}/32$.

Mode 2 baud rate = $2^{SMOD}/64 \times (f_{OSC})$

Modes 1 and 3

The baud rates in mode 1 and 3 are determined by the timer overflow rate. These baud rates can be determined by timer 1 or by timer 2 or by both (one for transmit and the other for receive).

6.3.3.1 Using Timer 1 to Generate Baud Rates

When timer 1 is used as the baud rate generator, the baud rates in modes 1 and 3 are determined by the timer 1 overflow rate and the value of SMOD as follows:

Modes 1,3 baud rate = $2^{SMOD}/32 \times$(timer 1 overflow rate)

The timer 1 interrupt should be disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD=0010_B). In that case, the baud rate is given by the formula

Modes 1,3 baud rate = $(2^{SMOD}/32) \times (f_{OSC}/(12 \times (256 - TH1)))$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, and configuring the timer to run as a 16-bit timer (high nibble of TMOD = 0001_B), and using the timer 1 interrupt to do a 16-bit software reload.

Table 6-7 lists commonly used baud rates and how they can be obtained from timer 1.

Table 6-7
Timer 1 Generated Commonly Used Baud Rates

Baud Rate	f_{osc}	SMOD	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 max: 640 kHz	7.68 MHz	X	X	X	X
Mode 2 max: 240 K	7.68 MHz	1	X	X	X
Modes 1, 3: 40 K	7.68 MHz	1	0	2	FF_H
19.2 K	7.68 MHz	0	0	2	FF_H
9.6 K	7.68 MHz	0	0	2	FE_H
4.8 K	7.68 MHz	0	0	2	FC_H
2.4 K	7.68 MHz	0	0	2	$F8_H$
1.2 K	7.68 MHz	0	0	2	EA_H
Mode 0 max : 1 MHz	12 MHz	X	X	X	X
Mode 2 max : 375 K	12 MHz	1	X	X	X
Mode 1,3 : 62.5 K	12 MHz	1	0	2	FF_H
19,2 K	11.059 MHz	1	0	2	FD_H
9.6 K	11.059 MHz	0	0	2	FD_H
4.8 K	11.059 MHz	0	0	2	FA_H
2.4 K	11.059 MHz	0	0	2	$F4_H$
1.2 K	11.059 MHz	0	0	2	$E8_H$
110	6 MHz	0	0	2	72_H
110	12 MHz	0	0	1	$FEEB_H$

6.3.3.2 Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts timer 2 into its baud rate generator mode, as shown in **figure 6-6-26**.

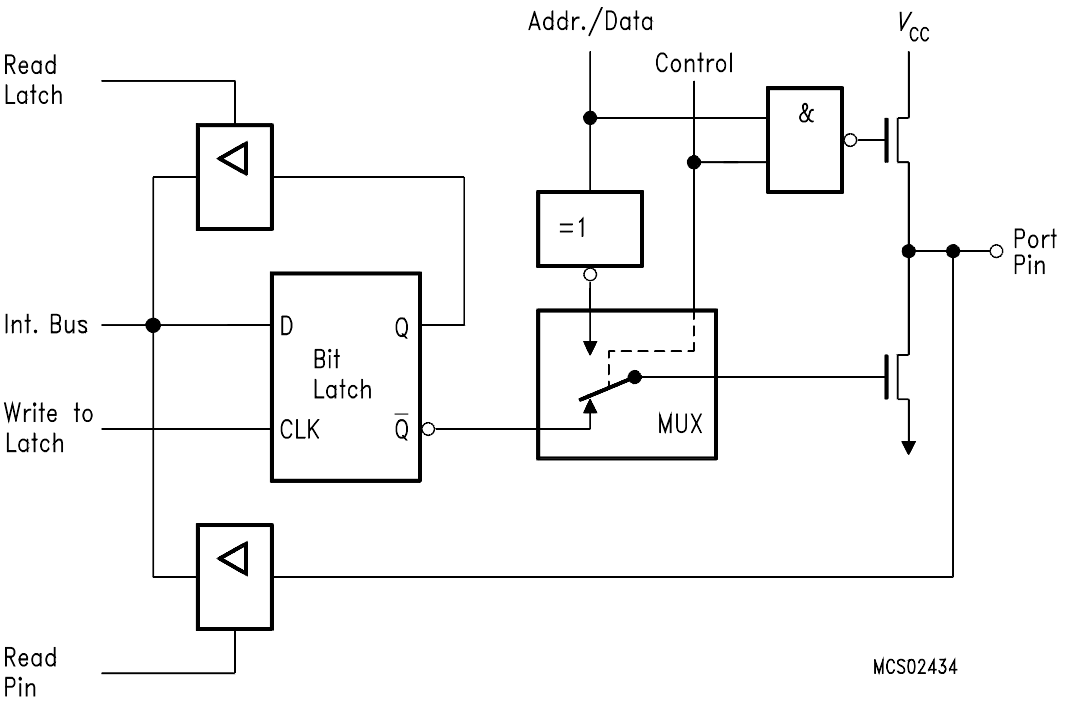


Figure 6-26
Timer 2 in Baud Rate Generator Mode

The baud rate generator mode is similar to the auto-reload mode, in that rollover in TH2 causes the timer 2 registers to be reloaded with the 16-bit value in registers RC2H and RC2L, which are preset by software.

Now the baud rates in modes 1 and 3 are determined by timer 2's overflow rate as follows:

Modes 1, 3 baud rate = timer 2 overflow rate/16

The timer can be configured for either "timer" or "counter" operation: In the most typical applications, it is configured for "timer" operation (C/$\overline{T2}$ = 0). "Timer" operation is a little different for timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at f_{OSC}/12). As a baud rate generator, however, it increments every state time (f_{OSC}/2). In that case the baud rate is given by the formula :

Modes 1,3 baud rate = $f_{OSC}/32 \times [65536 - (RC2H, RC2L)]$

where (RC2H, RC2L) is the content of RC2H and RC2L taken as a 16-bit unsigned integer.

Note that the rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the timer 2 interrupt does not have to be disabled when timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX can be used as an extra external interrupt, if desired.

It should be noted that when timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the timer is being incremented every state time, and the results of a read or write may not be accurate. The RC2H/RC2L registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the timer off (clear TR2) before accessing the timer 2 or RC2H/RC2L registers, in this case.

6.3.4 Details about Mode 0

Serial data enters and exists through RXD. TXD outputs the shift clock. 8 data bits are transmitted/ received: (LSB first). The baud rate is fixed at $f_{OSC}/12$.

Figure 6-15a shows a simplyfied functional diagram of the serial port in mode 0. The associated timing is illustrated in **figure 6-b**.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "WRITE to SBUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "WRITE to SBUF", and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0, and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1 and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted to the right one position.

As data bits shift out to the right, zeroes come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initialy loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "WRITE to SBUF".

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX control unit writes the bits 1111 1110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bit comes in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

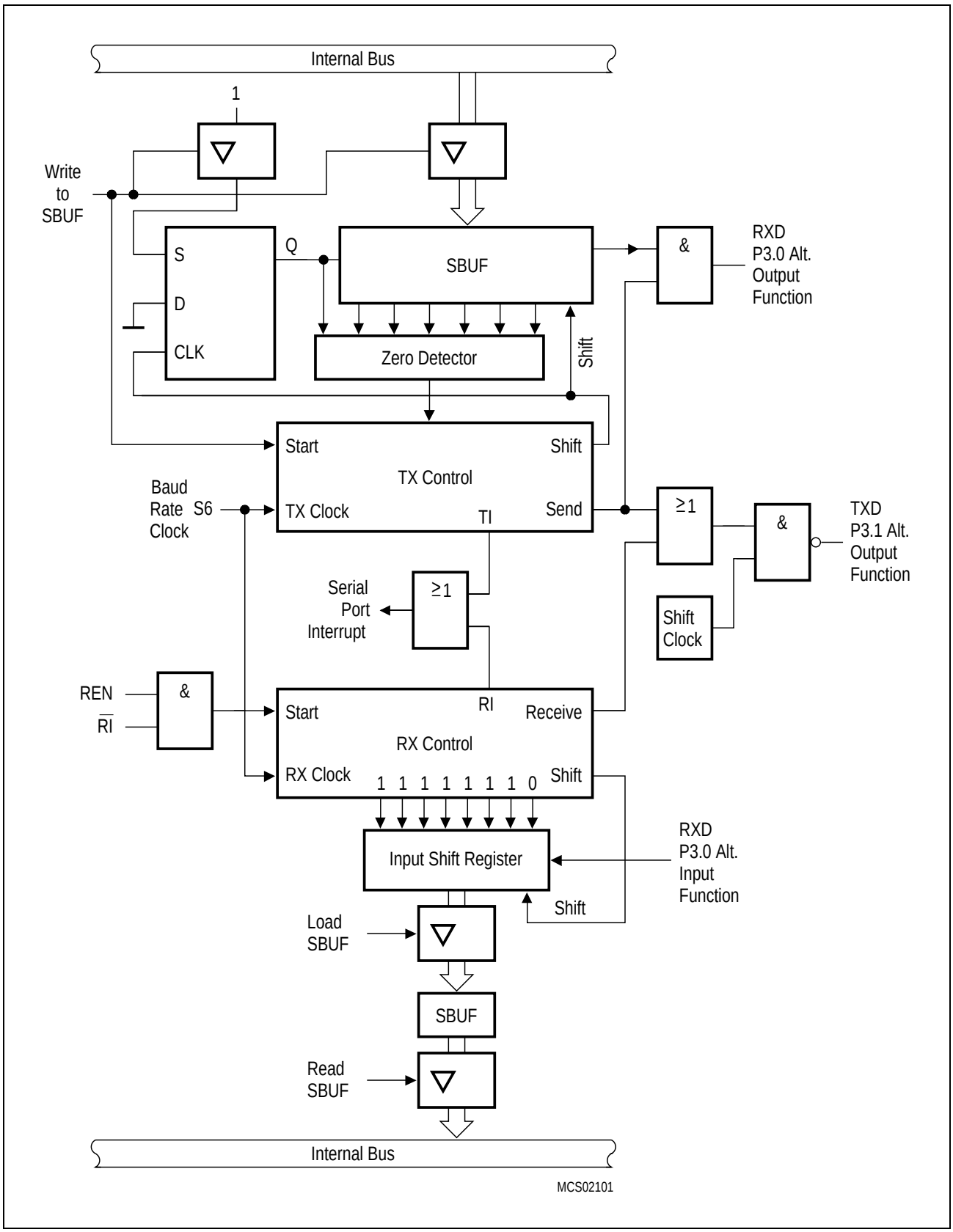


Figure 6-27a
Serial Interface, Mode 0, Functional Diagram

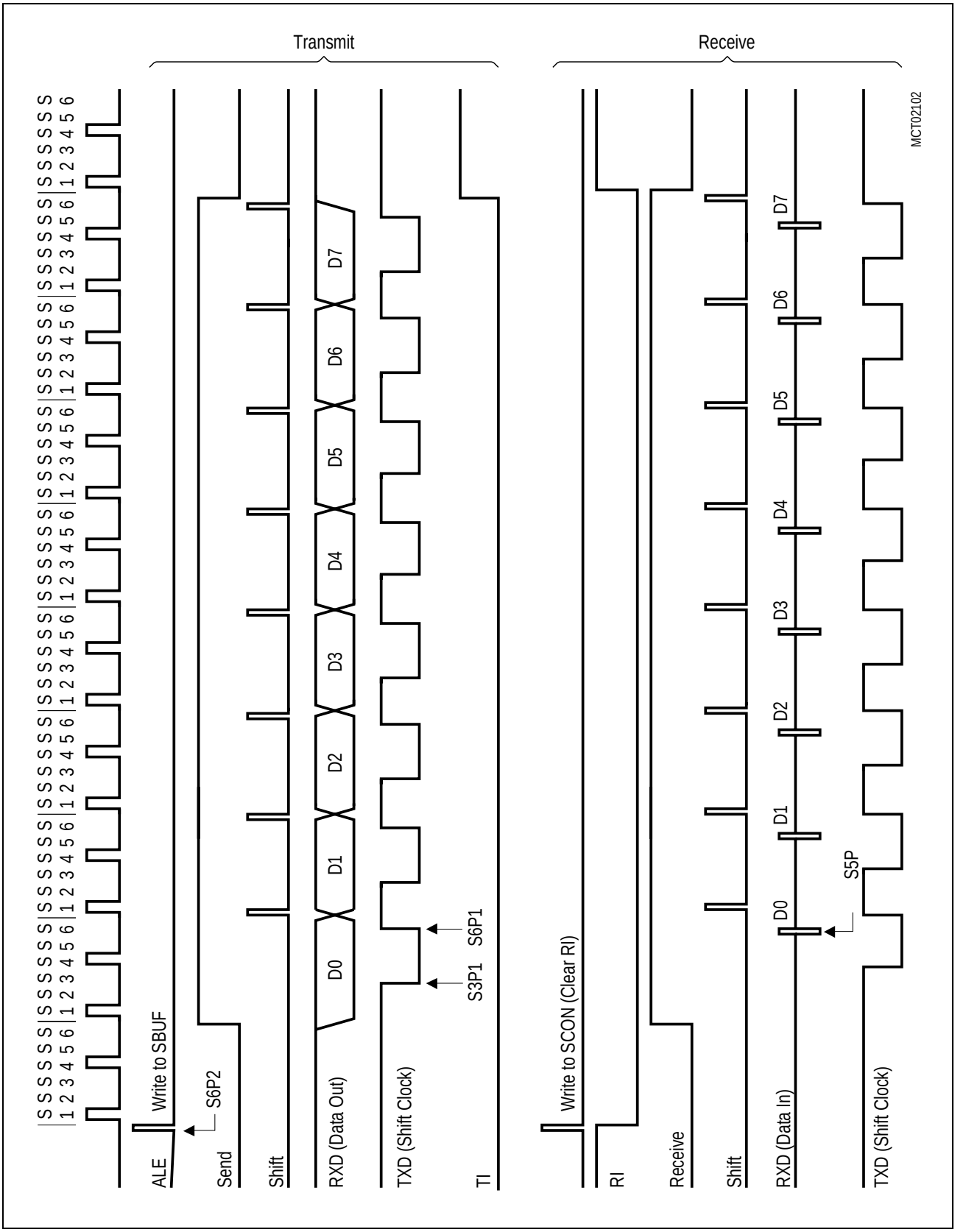


Figure 6-27b
Serial Interface, Mode 0, Timing

6.3.5 Details about Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. The baud rate is determined either by the timer 1 overflow rate, or the timer 2 overflow rate, or both (one for transmit and the other for receive).

Figure 6-6-28a shows a simplified functional diagram of the serial port in mode 1. The assiociated timings for transmit receive are illustrated in **figure 6-b**.

Transmission is initiated by an instruction that uses SBUF as a destination register. The "WRITE to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "WRITE to SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeroes are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 10th divide-by-16 rollover after "WRITE to SBUF".

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and $1FF_H$ is written into the input shift register, and reception of the rest of the frame will proceed.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at latest 2 of the 3 samples. This is done for the noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection or false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register, (which in mode 1 is a 9-bit register), it flags the RX control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0, and
2) Either SM2 = 0, or the received stop bit = 1

If either of these two condtions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bit goes into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RXD.

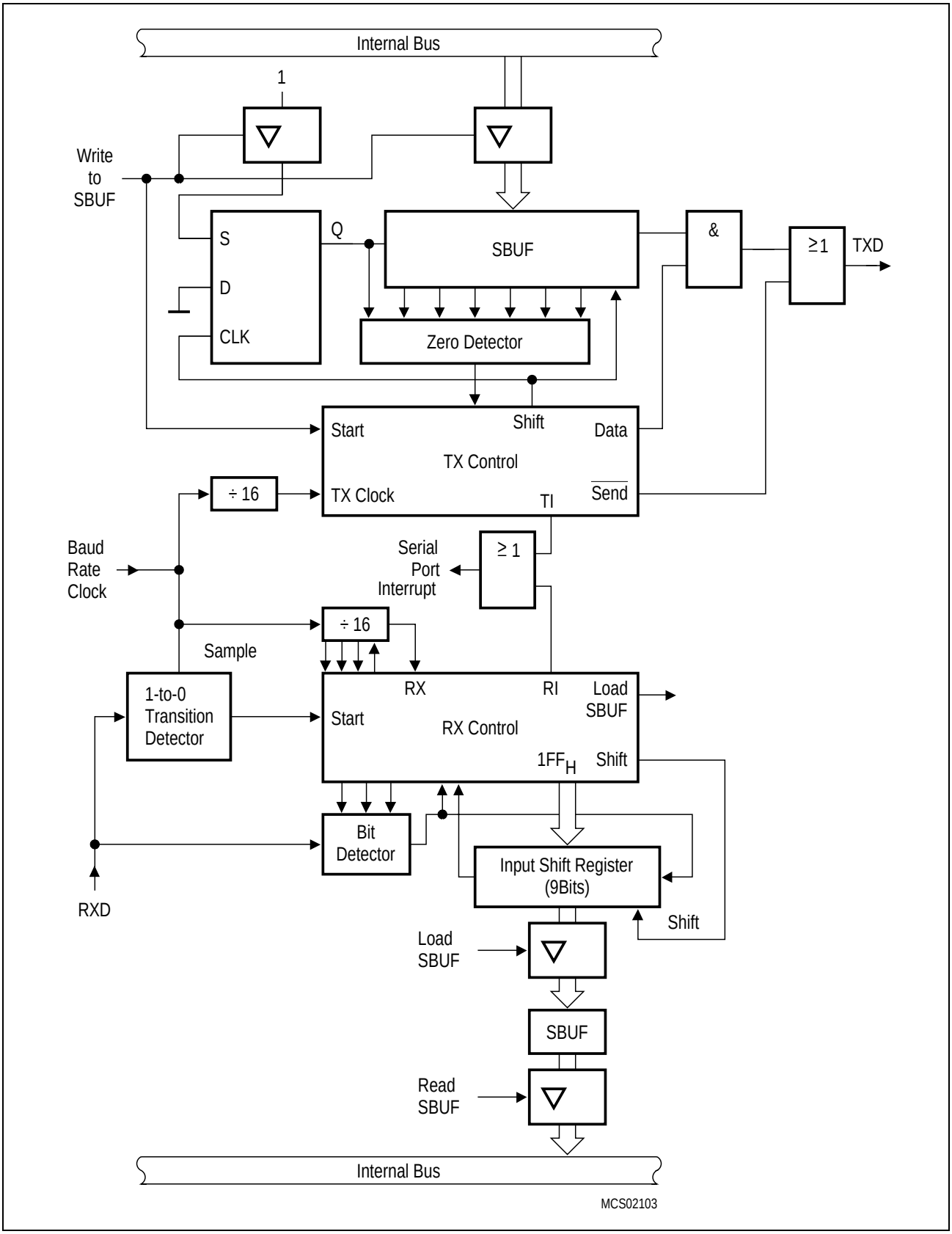


Figure 6-28a
Serial Interface, Mode 1, Functional Diagram

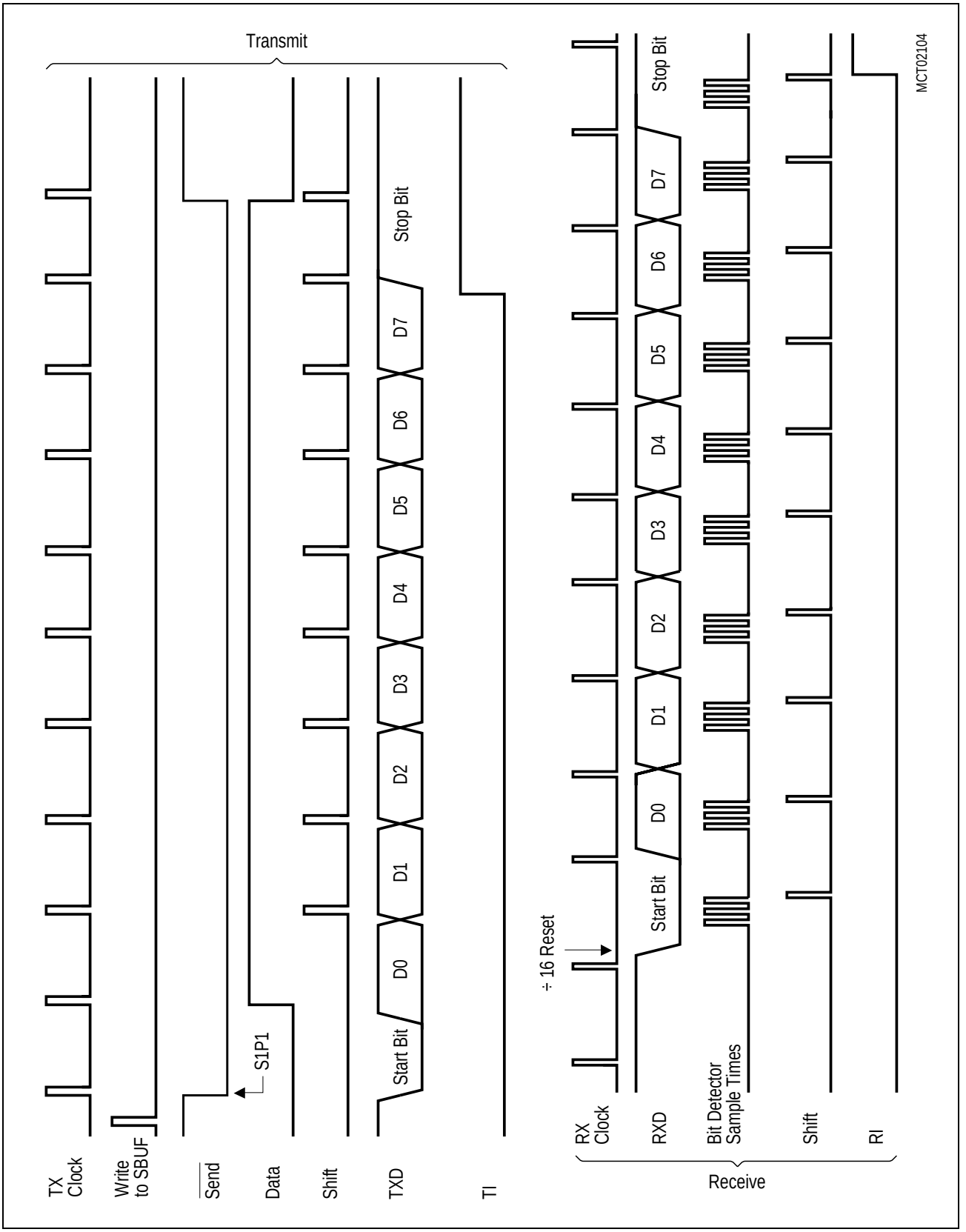


Figure 6-28b
Serial Interface, Mode 1, Timing

6.3.6 Details about Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in mode 2 (When bit SMOD in SFR PCON (87_H) is set, the baud rate is $f_{OSC}/32$). Mode 3 may have a variable baud rate generated from either timer 1 or 2 depending on the state of TCLK and RCLK (SFR T2CON).

Figure 6-6-29a shows a functional diagram of the serial port in modes 2 and 3. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register. The associated timings for transmit/receive are illustrated in **figure 6-b**.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "WRITE to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "WRITE to SBUF" signal.)

The transmision begins with activation of SEND, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeroes are clocked in. Thus, as data bits shift out to the right, zeroes are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeroes. This conditon flags the TX control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "WRITE to SBUF".

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and $1FF_H$ is written to the input shift register.

At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bit come from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in modes 2 and 3 is a 9-bit register), it flags the RX control block to do one last shift, load SBUF and RB8, and to set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and
2) Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bit goes into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8 or RI.

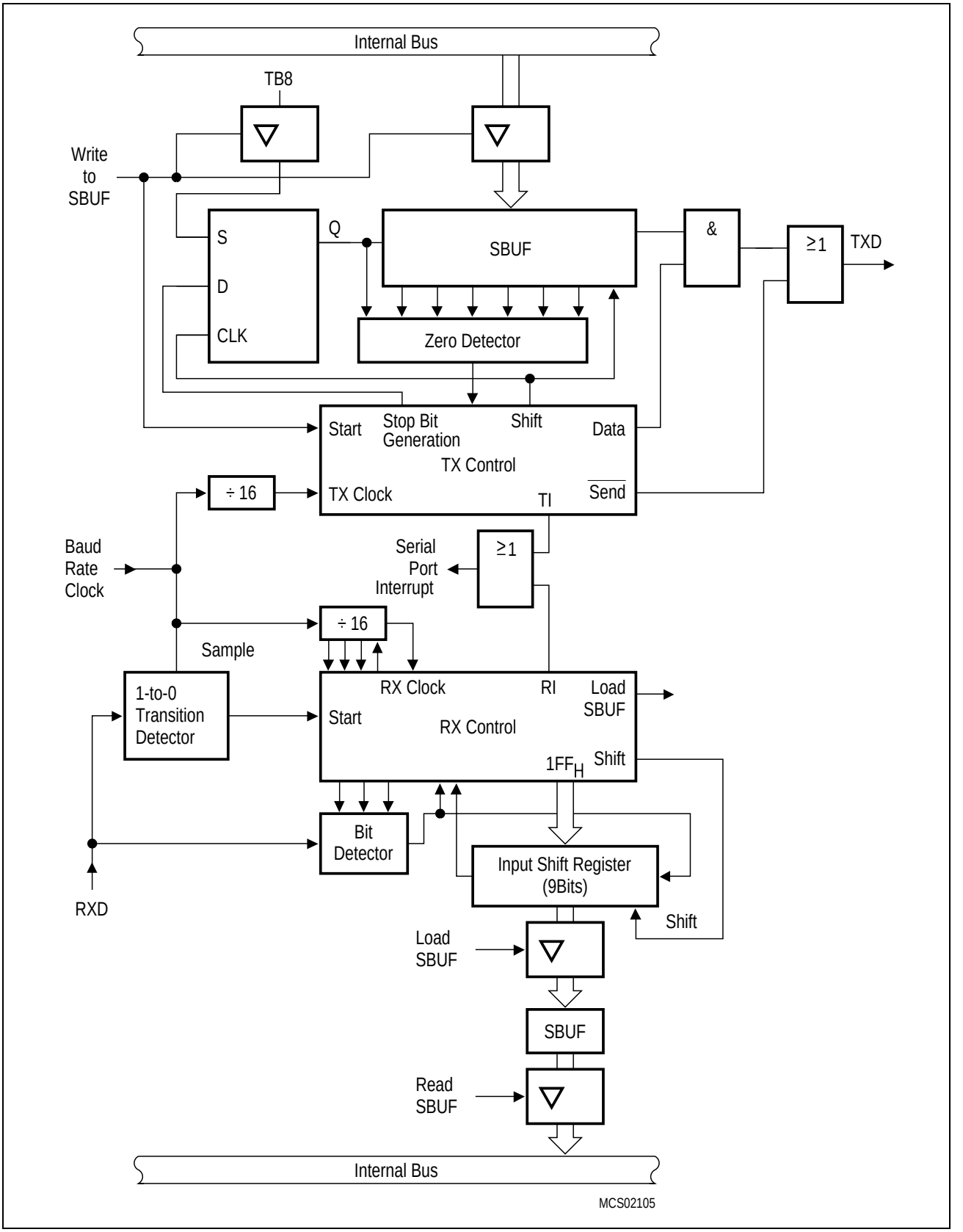


Figure 6-29a
Serial Interface, Mode 2 and 3, Functional Diagram

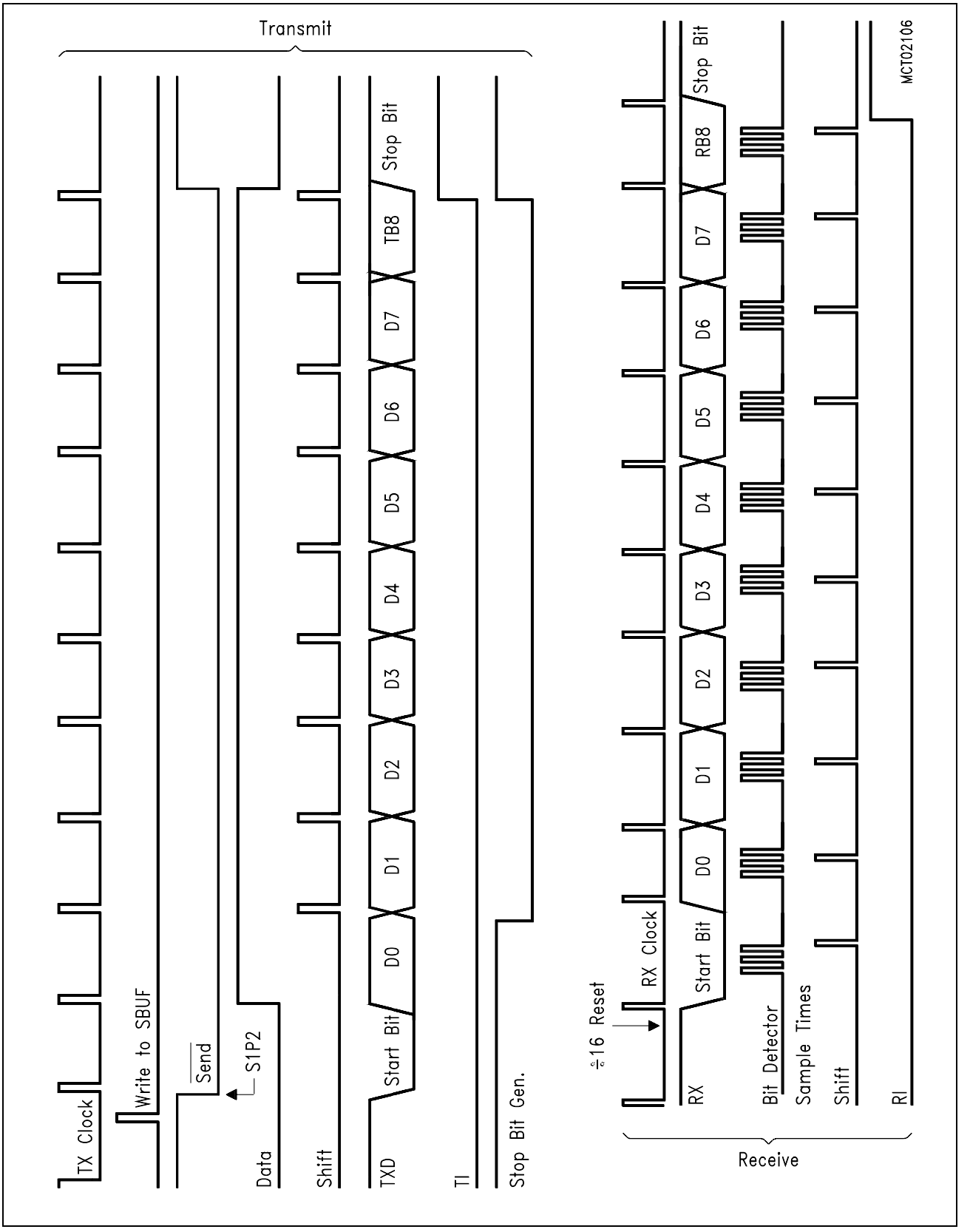


Figure 6-29b
Serial Interface, Mode 2 and 3, Timing

6.4 Synchronous Serial Channel (SSC)

The SAB-C511/513 microcontrollers provide a Synchronous Serial Channel unit, the SSC. This interface is compatible to the popular SPI serial bus interface. It can be used for simple I/O expansion via shift registers, for connection of a variety of peripheral components, such as A/D converters, EEPROMs etc., or for allowing several microcontrollers to be interconnected in a master/slave structure. It supports full-duplex or half-duplex operation and can run in a master or a slave mode.

6.4.1 SSC Block Diagram

Figure 6-6-30 shows the block diagram of the SSC. The central element of the SSC is an 8-bit shift register. The input and the output of this shift register are each connected via a control logic to the pin P1.3/SRI (SSC Receiver In) and P1.4/STO (SSC Transmitter Out). This shift register can be written to (SFR STB) and can be read through the Receive Buffer Register SRB.

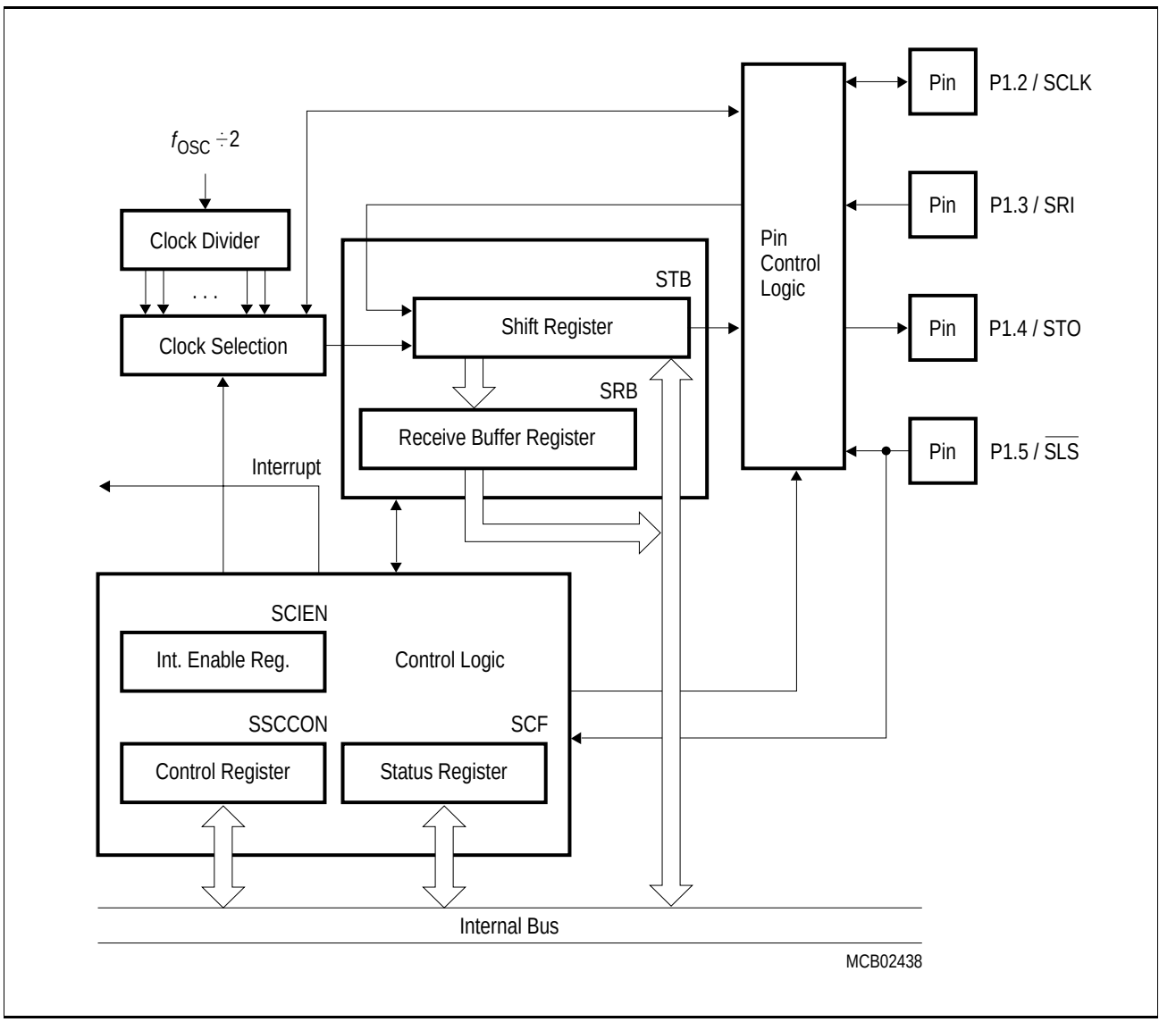


Figure 6-30
SSC Blockdiagram

As the SSC is a synchronous serial interface, for each transfer a dedicated clock signal sequence must be provided. The SSC has implemented a clock control circuit, which can generate the clock via a baud rate generator in the master mode, or receive the transfer clock in the slave mode. The clock signal is fully programmable for clock polarity and phase. The pin used for the clock signal is P1.2 / SCLK.

When operating in slave mode, a slave select input $\overline{SLS}$ is provided which enables the SSC interface and also will control the transmitter output. The pin used for this is P1.5 / $\overline{SLS}$. In addition to this there is an additional option for controlling the transmitter output by software.

The SSC control block is responsible for controlling the different modes and operation of the SSC, checking the status, and generating the respective status and interrupt signals.

6.4.2 General Operation of the SSC

After initialization of the SSC, the data to be transmitted has to be written into the shift register STB.

In master mode this will initiate the transfer by resetting the baudrate generator and starting the clock generation. The control bits CPOL and CPHA in the SSCCON register determine the idle polarity of the clock (polarity between transfers) and which clock edges are used for shifting and sampling data (see **figure 6-6-32**).
While the transmit data in the shift register is shifted out bit per bit starting with the MSB, the incoming receive data are shifted in, synchronized with the clock signal at pin SCLK. When the eight bits are shifted out (and the same number is of course shifted in), the contents of the shift register is transferred to the receive buffer register SRB, and the transmission complete flag TC is set. If enabled an interrupt request will be generated.
After the last bit has been shifted out and was stable for one bit time, the STO output will be switched to "1" (forced "1"), the idle state of STO. This allows connection of standard asynchronous receivers to the SSC in master mode.

In slave mode the device will wait for the slave select input $\overline{SLS}$ to be activated (=low) and then will shift in the data provided on the receive input according to the clock provided at the SCLK input and the setting of the CPOL ad CPHA bits. After eight bits have been shifted in, the content of the shift register is transferred to the receive buffer register and the transmission complete flag TC is set. If the transmitter is enabled in slave mode (TEN bit set to 1), the SSC will shift out at STO at the same time the data currently contained in the shift register. If the transmitter is disabled, the STO output will remain in the tristate state. This allows more than one slave to share a common select line.

If $\overline{SLS}$ is inactive the SSC will be inactive and the content of the shift register will not be modified.

6.4.3 Enable/Disable Control

Bit SSCEN of the SSCCON register globally enables or disables the synchronous serial interface. Setting SSCEN to “0” stops the baud rate generator and all internal activities of the SSC. Current transfers are aborted. The alternate output functions at pins P1.3/SRI, P1.4/STO, P1.5/$\overline{SLS}$, and P1.2/SCLK return to their primary I/O port function. These pins can now be used for general purpose I/O.

When the SSC is enabled and in master mode, pins P1.3/SRI, P1.4/STO, and P1.2/SCLK will be switched to the SSC control function. P1.4/STO and P1.2/SCLK actively will drive the lines. P1.5/$\overline{SLS}$ will remain a regular I/O pin.

The output latches of port pins dedicated to alternate functions must be programmed to logic 1 (= state after reset).

In slave mode all four control pins will be switched to the alternate function. However, STO will stay in the tristate state until the transmitter is enabled by $\overline{SLS}$ input being low and the TEN control bit is set to 1. This allows for more than one slave to be connected to one select line and the final selection of the slave will be done by a software protocol.

6.4.4 Baudrate Generation (Master Mode only)

The baudrate clock is generated out of the processor clock (fosc divided by 2). This clock is fed into a resetable divider with seven outputs for different baudrate clocks (fosc/8 to fosc/512). One of this eight clocks is selected by the bits BRS2,1, 0 in SSCCON and provided to the shift control logic.

Whenever the shift register is loaded with a new value, the baudrate generation is restarted with the trailing edge of the write signal to the shift register. In the case of CPHA = 0 the baudrate generator will be restarted in a way, that the first SCLK clock transisition will not occur before one half transmit clock cycle time after the register load. This ensures that there is sufficient setup time between MSB valid on the data output and the first sample clock edge and that the MSB has the same length than the other bits. (No special care is necessary in case of CPHA=1, because here the first clock edge will be used for shifting).

6.4.5 Write Collision Detection

When an attempt is made to write data to the shift register while a transfer is in progress, the WCOL bit in the status register will be set. The transfer in progress continues uninterrupted, the write will not access the shift register and will not corrupt data.
However, the data written erroneously will be stored in a shadow register and can be read by reading the STB register.

Depending on the operation mode there are different definitions for a transfer being considered to be in progress:

Master Mode

CPHA=0: from the trailing edge of the write into STB until the last sample clock edge
CPHA=1: from the first SCLK clock edge until the last sample clock edge

Note, that this also means, that writing new data into STB immediately after the transfer complete flag has been set (also initiated with the last sample clock edge) will not generate a write collision. However, this may shorten the lenght of the last bit (especially at slow baudrates) and prevent STO from switching to the forced "1" between transmissions.

Slave Mode

CPHA=0: while $\overline{SLS}$ is active
CPHA=1: from the first SCLK clock edge until the last sample clock edge

6.4.6 Master/Slave Mode Selection

The selection whether the SSC operates in master mode or in slave mode has to be made depending on the hardware configuration before the SSC will be enabled.

Normally a specific device will operate either as master or as slave unit. The SSC has no on-chip support for multimaster configurations (switching between master and slave mode operation). Operating the SSC as a master in a multimaster environment requires external circuitry for swapping transmit and receive lines.

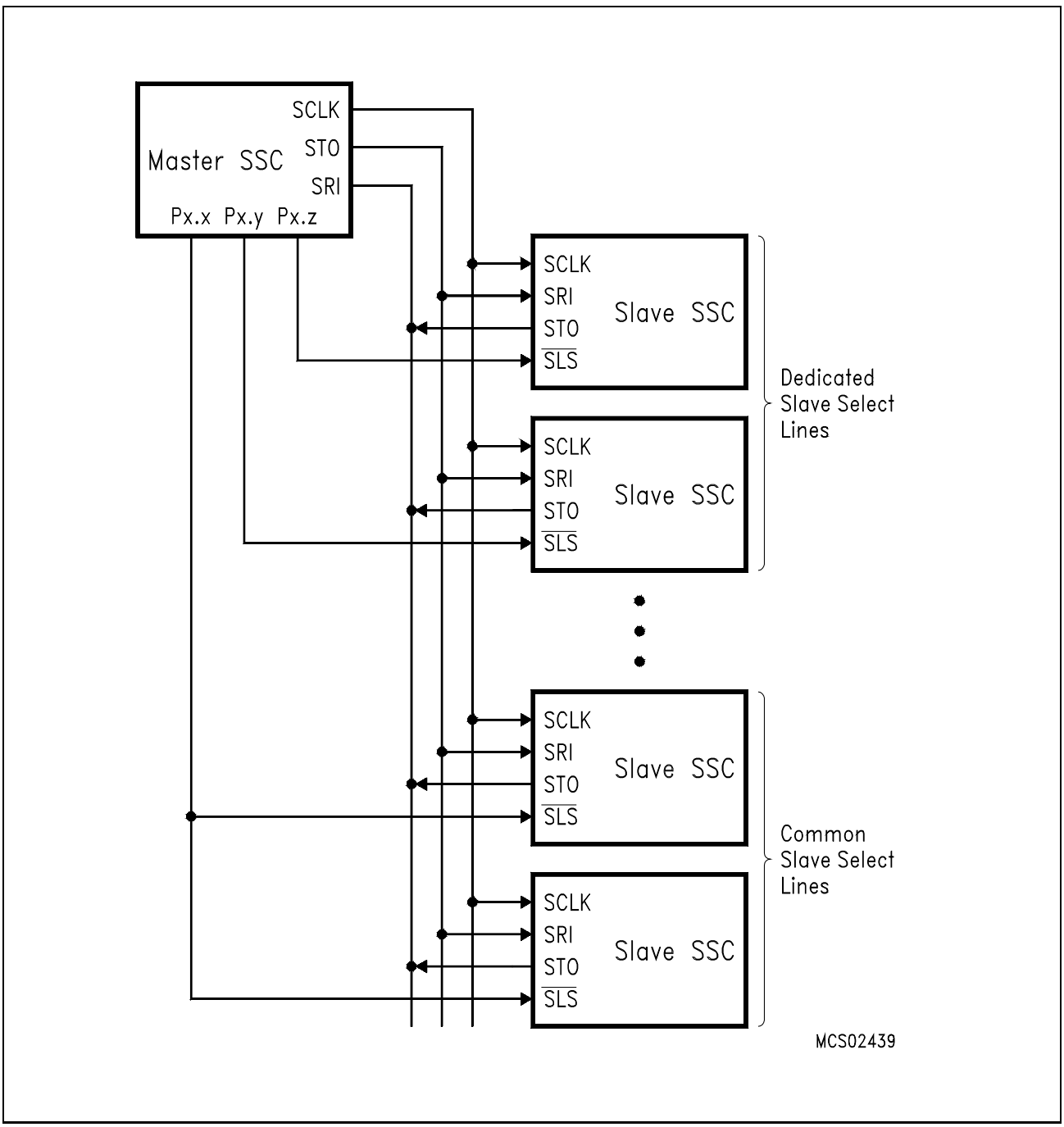


Figure 6-31
Typical SSC System Configuration

6.4.7 Data/Clock Timing Relationships

The SSC provides four different clocking schemes for clocking the data in and out of the shift register. Controlled by two bits in SSCCON, the clock polarity (idle state of the clock, control register bit CPOL) and the clock/data relationship (phase control, control register bit CPHA), i.e. which clock edges will be used for sample and shift. The following figures show the various possibilities.

6.4.7.1 Master Mode Operation

Figure 6-6-32 shows the clock/data/control relationship of the SSC in master mode. When CPHA is set to 1, the MSB of the data that was written into the shift register will be provided on the transmitter output after the first clock edge, the receiver input will sample with the next clock edge. The direction (rising or falling) of the respective clock edge is depending on the clock polarity selected. After the last bit has been shifted out, the data output STO will go to the high output level (logic 1) and remain there until the next transmission is started. However, when enabling the SSC after reset, the logic level of STO will be undefined, until the first transmission starts.

When CPHA is 0, the MSB will output immediately after the data was written into the shift register. The first clock edge of SCLK will be used for sampling the input data, the next to shift out the next bit. Between transmissons the data output STO will be "1".

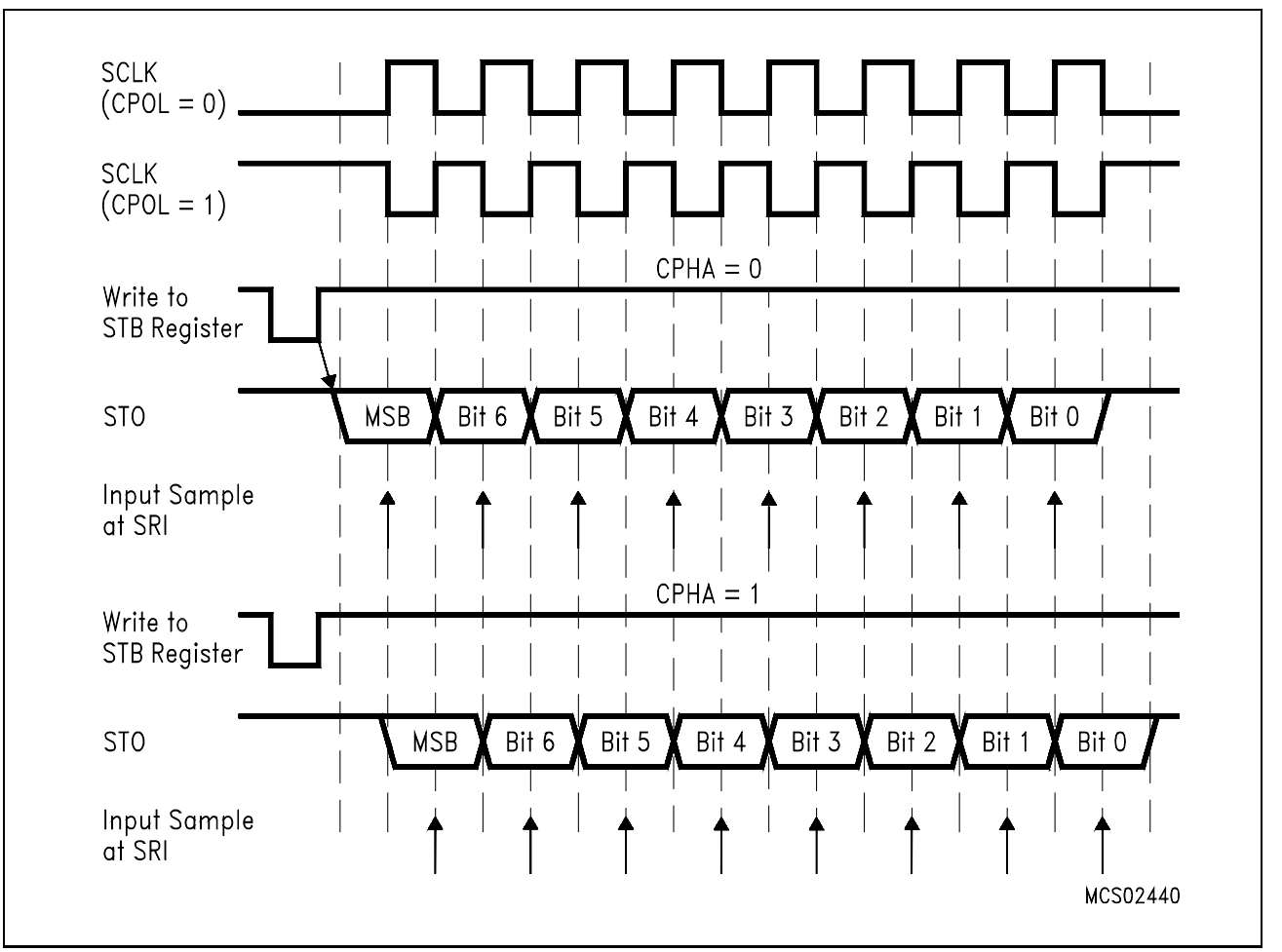


Figure 6-32
Master Mode Operation of SSC

6.4.7.2 Slave Mode Operation

Figure 6-6-33 shows the clock/data/control relationship of the SSC in slave mode. When $\overline{SLS}$ is active (low) and CPHA is 1, the MSB of the data that was written into the shift register will be provided on the transmitter output after the first clock edge (if the transmitter was enabled by setting the TEN bit to 1), the receiver input will sample the input data with the next clock edge. The direction (rising or falling) of the respective clock edge is depending on the clock polarity selected. In this case (CPHA = 1) the $\overline{SLS}$ input may stay active during the transmission of consecutive bytes.

When CPHA = 0 and the transmitter is enabled, the MSB of the shift register is provided immediately after the $\overline{SLS}$ input is pulled to active state (low). The receiver will sample the input with the first clock edge, and the transmitter will shift out the next bit with the following clock edge. If the transmitter is disabled the output will remain in the high impedancec state. In this case (CPHA=0), correct operation requires that the $\overline{SLS}$ input to go inactive between consecutive bytes.

When $\overline{SLS}$ is inactive the internal shift clock is disabled and the content of the shift register will not be modified. This also means that $\overline{SLS}$ must stay active until the transmission is completed.
If during a transmission $\overline{SLS}$ goes inactive before all eight bits are received, the reception process will be aborted and the internal frame counter will be reset. TC will not be set in this case. With the next activation of $\overline{SLS}$ a new reception process will be started.

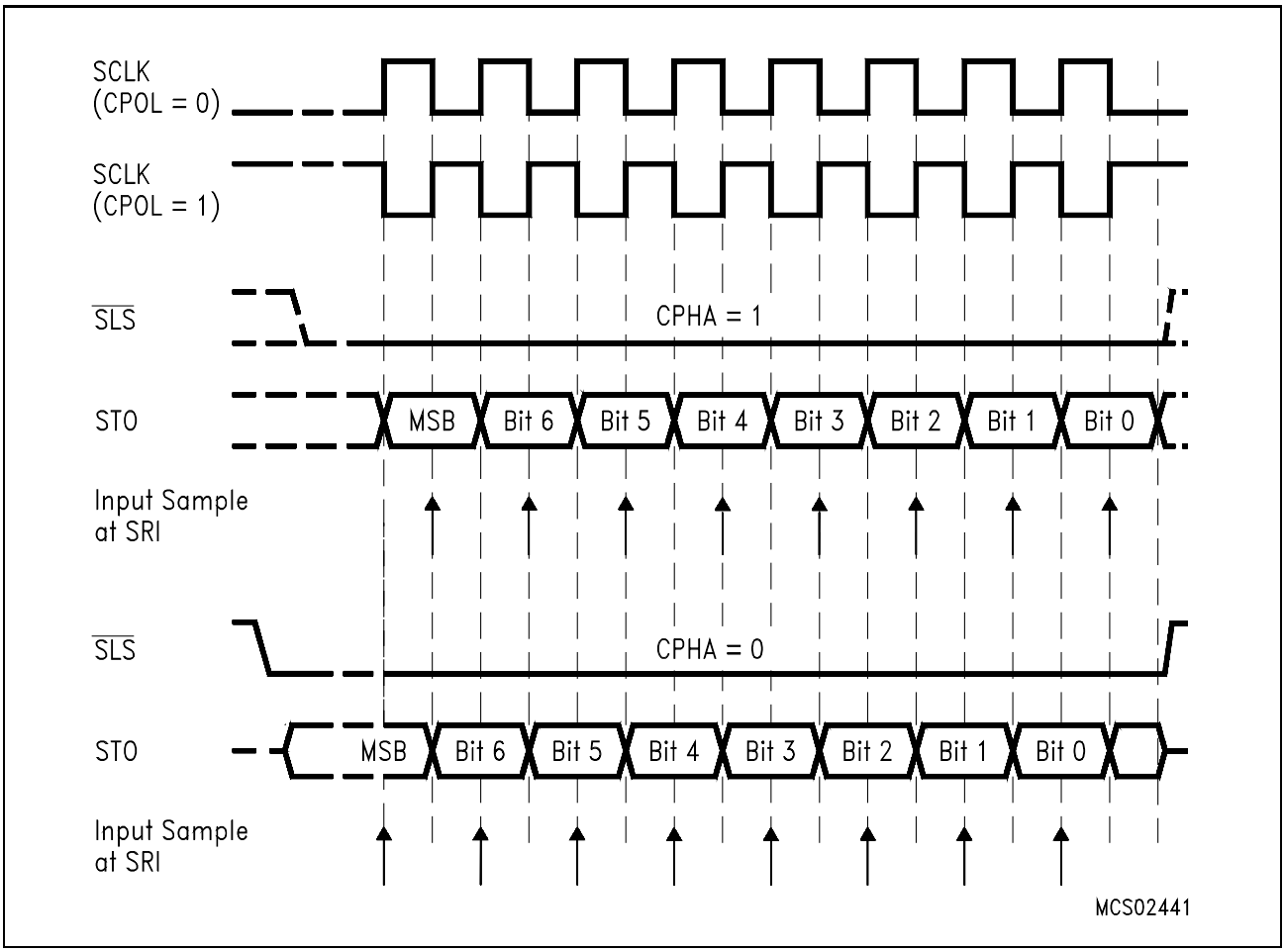


Figure 6-33
Slave Mode Operation of SSC

6.4.8 Register Description

6.4.8.1 SSC Control Register SSCCON

The bit addressable register SSCCON provides the basic control of the SSC functions like general enable/disable, mode selections and transmitter control.

Special Function Registers SSCCON (Address $E8_H$) **Reset Value : 07_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$E8_H$	SCEN	TEN	MSTR	CPOL	CPHA	BRS2	BRS1	BRS0	SSCCON

Bit	Function
SCEN	SSC System Enable SCEN =0 : SSC subsystem is disabled, related pins are available as general I/O. SCEN=1 : SSC subsystem is enabled.
TEN	Slave Mode - Transmitter enable TEN =0 : Transmitter output STO will remain in tristate state, regardless of the state of $\overline{SLS}$. TEN=1 and $\overline{SLS}$ =0 : Transmitter will drive the STO output. In master mode the transmitter will be enabled all the time, regardless of the setting of TEN.
MSTR	Master Mode Select MSTR=0 : Slave mode is selected MSTR=1 : Master mode is selected This bit has to be set to the correct value depending on the hardware setup of the system before the SSC will be enabled. It must not be modified afterwards. There is no on-chip support for dynamic switching between master and slave mode operation.
CPOL	Clock Polarity This bit controls the polarity of the shift clock and in conjunction with the CPHA bit which clock edges are used for sample and shift. CPOL=0 : SCLK idle state is low. CPOL=1 : SCLK idle state is high.

Bit	Function
CPHA	Clock Phase This bit controls in conjunction with the CPOL bit controls which clock edges are used for sample and shift CPHA=0 : The first clock edge of SCLK is used to sample the data, the second to shift the next bit out at STO. In master mode the transmitter will provide the first data bit on STO immediately after the data was written into the STB register. In slave mode the transmitter (if enabled via TEN) will shift out the first data bit with the falling edge of $\overline{SLS}$. CPHA=1 : The first data bit is shifted out with the first clock edge of SCLK and sampled with the second clock edge
BRS2, BRS1, BRS0	Baudrate Selection Bits These bits select one of the possible divide factors for generating the baudrate out of the micrcontroller clock rate f_{osc} . The baudrate is defined by . $$Baudrate = \frac{fosc}{Dividefactor} = \frac{fosc}{4 \bullet 2^{BRS(2-0)}}$$ for BRS (2-0) ≠ 0

BRS(2-0)	Divide Factor	Example: Baudrate for fosc = 7.68 MHz
0	reserved	reserved
1	8	960 kBaud
2	16	480 kBaud
3	32	240 kBaud
4	64	120 kBaud
5	128	60 kBaud
6	256	30 kBaud
7	512	15 kBaud

Note: SSCCON must be programmed only when the SSC is idle. Modifying the contents of SSCCON while a transmission is in progress will corrupt the current transfer and will lead to unpredictable results.

6.4.8.2 SSC Interrupt Enable Register SCIEN

This byte addressable register enables or diasables interrupt request for the status bits.
This register must only be written when the SSC interrupts are disabled in the general interrupt enable register IE ($A8_H$), otherwise unexpected interrupt requests may occur.

Special Function Registers SCIEN (Address $F9_H$) **Reset Value : $XXXXXX00_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$F9_H$	–	–	–	–	–	–	WCEN	TCEN	SCIEN

Bit	Function
–	Not implemented. Reserved for future use. During reads these bits will be undefined.
WCEN	Write Collision Interrupt Enable WCEN =0 : No interrupt request will be generated if the WCOL bit in the status register SCF is set. WCEN=1 : An interrupt is generated if the WCOL bit in the status register SCF is set.
TCEN	Transfer Completed Interrupt Enable TCEN =0 : No interrupt request will be generated if the TC bit in the status register SCF is set. TCEN=1 : An interrupt is generated if the TC bit in the status register SCF is set.

Note: The SSC interupt behaviour is in addition affected by bit ESSC in the interrupt enable register IE and by bit PSSC of the interrupt priority register IP.

6.4.8.3 Status Register SCF

This bit addressable register contains the status bits.

Special Function Registers SCF (Address $F8_H$) **Reset Value : $XXXXXX00_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$F8_H$	–	–	–	–	–	–	WCOL	TC	SCF

Bit	Function
–	Not implemented. Reserved for future use. During reads these bits will be undefined.
WCOL	Write Collision Detect If WCOL is set it indicates that an attempt was made to write to the shift register STB while a data transfer was in progress and not fully completed. This bit will be set at the trailing edge of the write signal during the erronous write attempt. This bit can be reset in two different ways : 1. writing a "0" to the bit (bit access, byte access or read-modify-write access); 2. by reading the bit or the status register, followed by a write access to STB. If bit WCEN in the SCIEN register is set, an interrupt request will be generated if WCOL is set.
TC	Transfer Completed If TC is set it indicates that the last transfer has been completed. It is set with the last sample clock edge of a reception process. This bit can be reset in two different ways: 1. writing a "0" to the bit (bit access, byte access or read-modify-write access) after the receive buffer register SRB has been read; 2. by reading the bit or the status register, followed by a read access to SRB. If bit TCEN in the SCIEN register is set, an interrupt request will be generated if TC is set.

6.4.8.4 Data Registers STB and SRB

The register STB holds the data to be transmitted while SRB contains the data which was received during the last transfer. A write to the STB places the data directly into the shift register for transmission. Only in master mode this also will initiate the transmission/reception process. When a write collision occurs STB will hold the value written erroneously. This value can be read by reading from STB.

A read from the receive buffer register SRB will transfer the data of the last transfer completed. This register must be read before the next transmission completes or the data will be lost. There is no indication for this overrun condition.

After reset the contents of the shift register and the receive buffer register are undefined.

6.4.8.5 Mode Test Register SSCMOD

The register SSCMOD is used to enable test modes during factory test. It must not be written or modified during normal operation of the SAB-C511/513.

Special Function Registers SSCMOD (Address EB_H) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
EB_H	0	0	0	0	0	0	0	0	SSCMOD

Bit	Function
7-0	All bits of this register are set to 0 after reset and must not be modified !

6.4.8.6 Location of Bitaddressable Control and Status Bits

Table 6-8 shows the addresses of the control and status bits in the bit-addressable special function registers used for the SSC.

Table 6-8
Location of SSC Control and Status Bits

Control or Status Bit		SFR	Register Address	Bit-Addr.
Baudrate Select 0,	BRS0	SSCCON	$E8_H$	$E8_H$
Baudrate Select 1,	BRS1	SSCCON	$E8_H$	$E9_H$
Baudrate Select 2,	BRS2	SSCCON	$E8_H$	EA_H
Clock Phase Select,	CPHA	SSCCON	$E8_H$	EB_H
Clock Polarity Control,	CPOL	SSCCON	$E8_H$	EC_H
Master/Slave Mode Select,	MSTR	SSCCON	$E8_H$	ED_H
Transmitter Enable,	TEN	SSCCON	$E8_H$	EE_H
SSC System Enable,	SCEN	SSCCON	$E8_H$	EF_H
Transmit Complete Flag	TC	SCF	$F8_H$	$F8_H$
Write Collision Flag	WCOL	SCF	$F8_H$	$F9_H$

7 Interrupt System

The C511/513 provides 7 interrupt sources with two priority levels. Five interrupts can be generated by the on-chip peripherals (timer 0, timer 1, timer 2, USART and SSC) and three interrupts may be triggered externally (P1.1/T2EX, P3.2/INT0 and P3.3/INT1).

Short Description of the Interrupt Structure for Advanced C511/1513 Users

The interrupt structure of the C500 architecture has been mainly adapted from the 8052 microcontroller. Thus, each interrupt source has its dedicated interrupt vector and can be enabled/ disabled individually. There are also two priority levels available.

Figure 7-1 gives a general overview of the interrupt sources and illustrates the request and control flags described in the next sections.

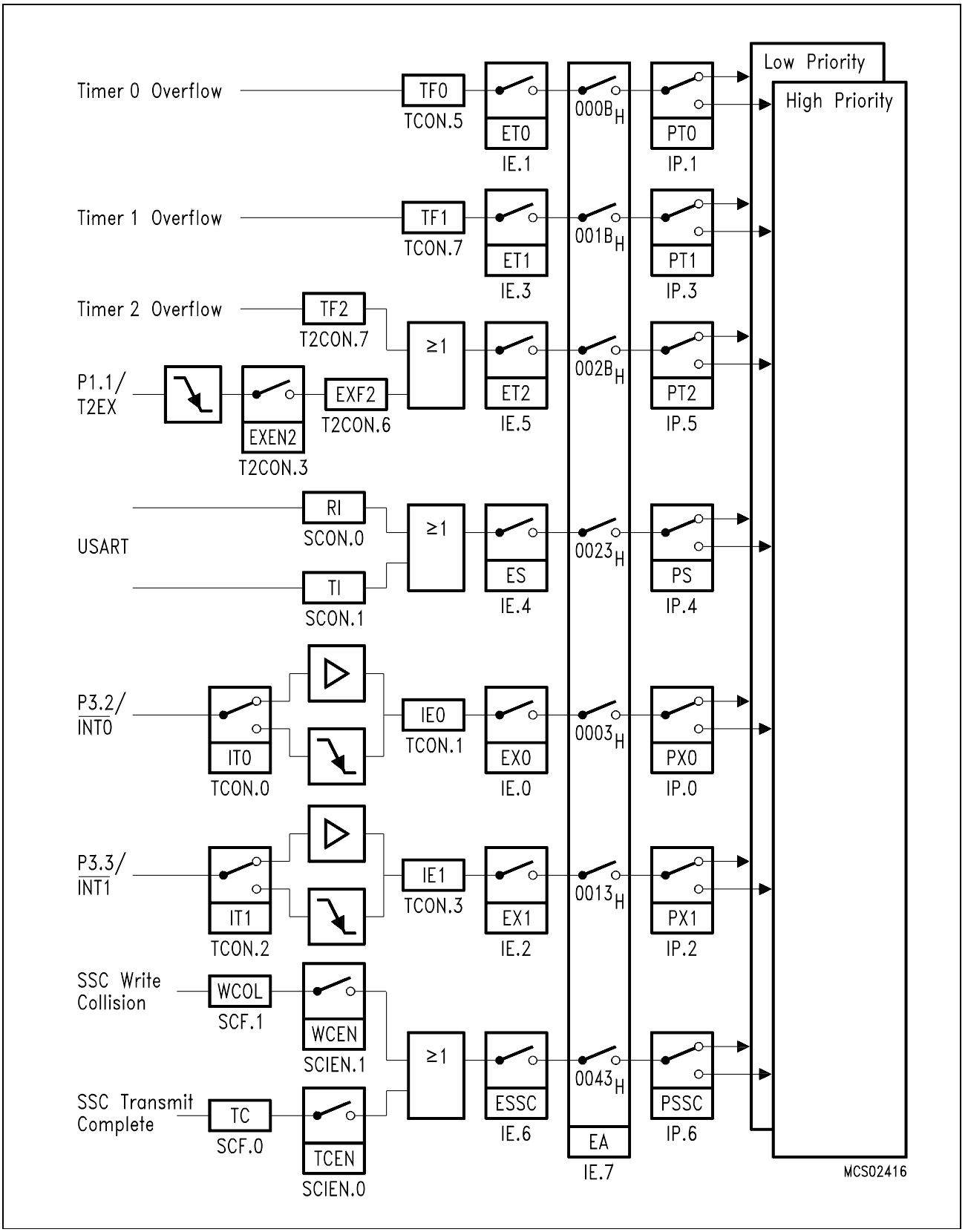


Figure 7-1
Interrupt Request Sources

7.1 Interrupt Structure

A common mechanism is used to generate the various interrupts, each source having its own request flag(s) located in a special function register (e.g. TCON, T2CON, SCON). When the peripheral or external source meets the condition for an interrupt, the dedicated request flag is set, whether an interrupt is enabled or not. For example, each timer 0 overflow sets the corresponding request flag TF0. If it is already set, it retains a one (1). But the interrupt is not necessarily serviced.

Now each interrupt requested by the corresponding flag can individually be enabled or disabled by the enable bits in SFR IE. This determines whether the interrupt will actually be performed. In addition, there is a global enable bit for all interrupts which, when cleared, disables all interrupts independent of their individual enable bits.

Table 8-1
Interrupt Sources and Vectors

Source (Request Flags)	Vector	Vector Address
IE0	External interrupt 0	0003_H
TF0	Timer 0 interrupt	$000B_H$
IE1	External interrupt 1	0013_H
TF1	Timer 1 interrupt	$001B_H$
RI + TI	USART serial port interrupt, (C513/C513A/C513A-H only)	0023_H
TF2 + EXF2	Timer 2 interrupt (C513/C513A/C513A-H only)	$002B_H$
SSCI	Synchronous serial channel interrupt (SSC)	0043_H

7.2 Interrupt Control Bits

7.2.1 Interrupt Enables

Each interrupt vector can be individually enabled or disabled by setting or clearing the corresponding bit in the SFR IE (Interrupt Enable). This register also contains the global disable bit EA, which can be cleared/set to disable/enable all interrupts at once.

Special Function Registers IE (Address $A8_H$) **Reset Value : 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$A8_H$	EA	ESSC	ET2	ES	ET1	EX1	ET0	EX0	IE

Bit	Function
EA	Disables all Interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
ESSC	SSC Interrupt Enable If ESSC = 0 the interrupt of the synchronous serial channel is disabled
ET2	Timer 2 Interrupt Enable. If ET2 = 0, the timer 2 interrupt is disabled.
ES	USART Serial Channel Interrupt Enable (C513/C513A/C513A-H only) If ES = 0, the serial channel interrupt is disabled.
ET1	Timer 1 Overflow Interrupt Enable. If ET1 = 0, the timer 1 interrupt is disabled.
EX1	External Interrupt 1 Enable. If EX1 = 0, the external interrupt 1 is disabled.
ET0	Timer 0 Overflow Interrupt Enable. If ET0 = 0, the timer 0 interrupt is disabled.
EX0	External Interrupt 0 Enable. If EX0 = 0, the external interrupt 0 is disabled.

7.2.2 Interrupt Priorities

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR IP (Interrupt Priority, 0: low priority, 1: high priority).

Special Function Registers IP (Address $B8_H$) **Reset Value : $X0000000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$B8_H$	–	PSSC	PT2	PS	PT1	PX1	PT0	PX0	IP

Bit	Function
–	Not implemented, reserved for future use. Will be read as “1“.
PSSC	SSC Priority Level If PSSC = 0, the SSC interrupt has a low priority.
PT2	Timer 2 Interrupt Priority Level. If PT2 = 0, the timer 2 interrupt has a low priority.
PS	Serial Channel (USART) Interrupt Priority Level (C513//C513A/C513A-H only) If PS = 0, the serial channel interrupt has a low priority.
PT1	Timer 1 Overflow Interrupt Priority Level. If PT1 = 0, the timer 1 interrupt has a low priority.
PX1	External Interrupt 1 Priority Level. If PX1 = 0, the external interrupt 1 has a low priority.
PT0	Timer 0 Overflow Interrupt Priority Level. If PT0 = 0, the timer 0 interrupt has a low priority.
PX0	External Interrupt 0 Priority Level. If PX0 = 0, the external interrupt 0 has a low priority.

If two interrupt requests of different priority level are received simultaneously, the request of higher priority is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as shown in **table 7-1**.

Table 7-1
Priority-within-Level Structure

Interrupt Source		Priority
External Interrupt 0,	IE0	High
Synchronous Serial Channel	SSC	
Timer 0 Interrupt,	TF0	
External Interrupt 1,	IE1	↓
Timer 1 Interrupt,	TF1	
Universal Serial Channel,	RI or TI	
Timer 2 Interrupt,	TF2 or EXF2	Low

A low-priority interrupt in service can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt in service cannot be interrupted by any other interrupt source.

The interrupt request flags are located in bit-addressable SFR's as listed in **table 7-2**.

Table 7-2
Location of Interrupt Sources Request Flags

Interrupt Request Flag		SFR	Address	Bit-Addr.
External Interrupt 0,	IE0	TCON	88_H	89_H
Timer 0 Interrupt,	TF0	TCON	88_H	$8D_H$
External Interrupt 1,	IE1	TCON	88_H	$8B_H$
Timer 1 Interrupt,	TF1	TCON	88_H	$8F_H$
Universal Serial Channel,	RI (C513/C513A/ C513A-H only)	SCON	98_H	98_H
Universal Serial Channel,	TI (C513/C513A/ C513A-H only)	SCON	98_H	99_H
Timer 2 Interrupt,	TF2	T2CON	$C8_H$	CF_H
Timer 2 Interrupt,	EXF2	T2CON	$C8_H$	CE_H
Synchr. Serial Channel	TC	SCF	$F8_H$	$F8_H$
Synchr. Serial Channel	WCOL	SCF	$F8_H$	$F9_H$

7.3 How Interrupts are Handled

The interrupt flags are sampled at S5P2 in each machine cycle. The sampled flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceeding cycle, the polling cycle will find it and the interrupt system will generate a LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1) An interrupt of equal or higher priority is already in progress.
2) The current (polling) cycle is not in the final cycle of the instruction in progress.
3) The instruction in progress is RETI or any write access to registers IE or IP.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress is completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write access to registers IE or IP, then at least one more instruction will be executed before any interrupt is vectored too; this delay guarantees that changes of the interrupt status can be observed by the CPU.

The polling cycle is repeated with each machine cycle and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if any interrupt flag is active but not being responded to for one of the conditions already mentioned, or if the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle interrogates only the pending interrupt requests.

The polling cycle/LCALL sequence is illustrated in **figure 7-2**.

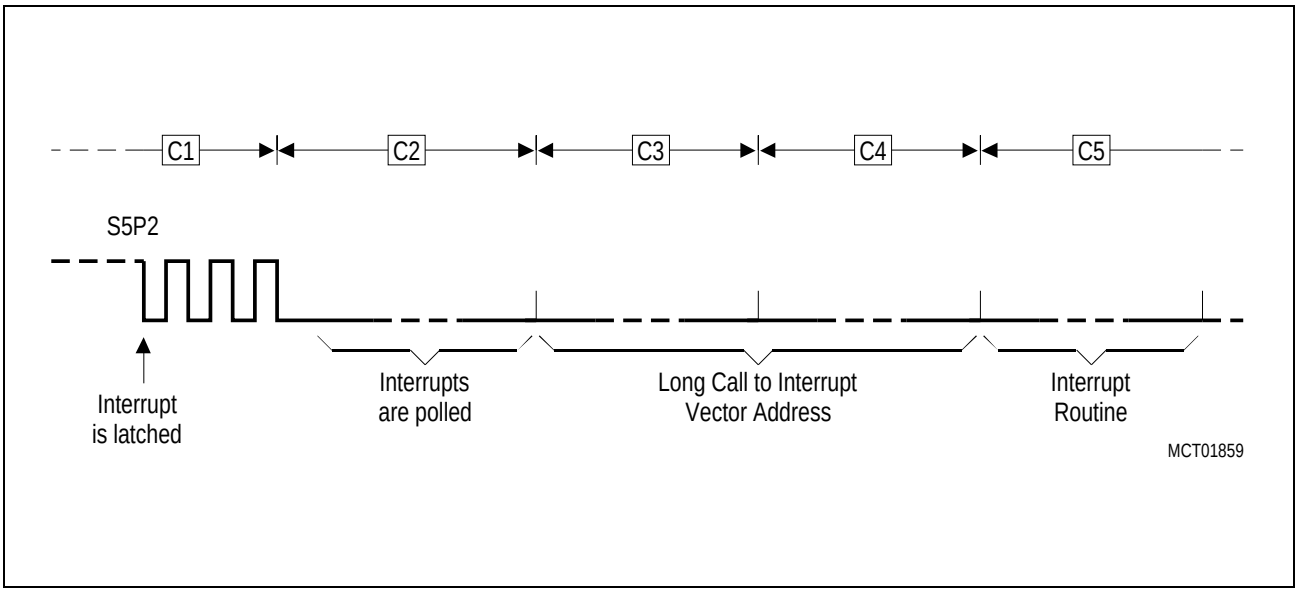


Figure 7-2
Interrupt Response Timing Diagram

Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in **figure 7-2** then, in accordance with the above rules, it will be vectored to during C5 and C6 without any instruction for the lower priority routine to be executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not. Then this has to be done by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored too.

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the two top bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt priority level. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress. In this case no interrupt of the same or lower priority level would be acknowledged.

7.4 External Interrupts

The external interrupts 0 and 1 can be programmed to be level-activated or negative-transition activated by setting or clearing bit IT0 or IT1, respectively, in register TCON. If ITx = 0 (x = 0 or 1), external interrupt x is triggered by a detected low level at the INTx pin. If ITx = 1, external interrupt x is negative edge-triggered. In this mode, if successive samples of the INTx pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

If the external interrupt 0 or 1 is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

The external timer 2 reload trigger interrupt request flag EXF2 will be activated by a negative transition at pin P1.1/T2EX but only if bit EXEN2 is set.

Since the external interrupt pins are sampled once in each machine cycle, an input low should be held for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set (see **figure 7-2**). The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

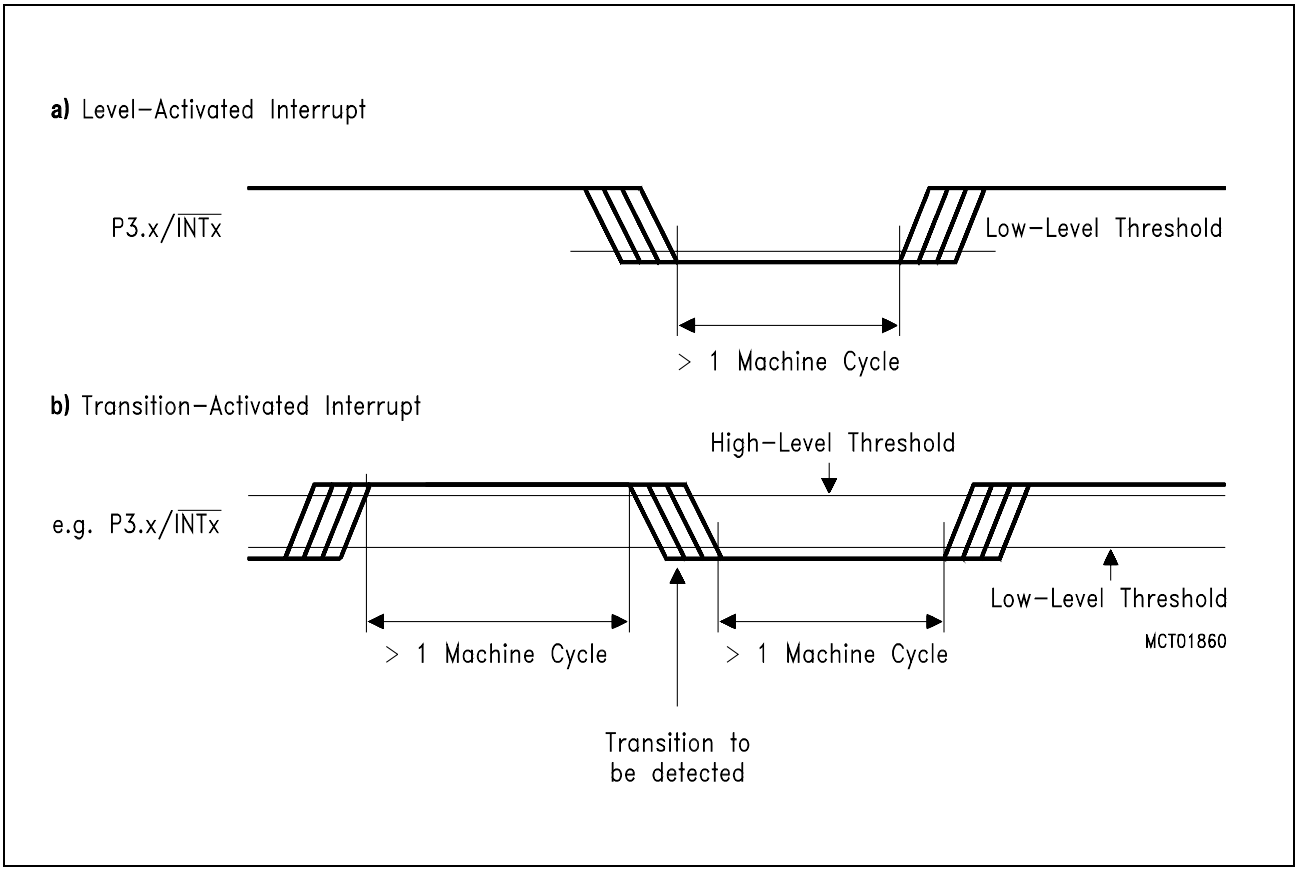


Figure 7-3
External Interrupt Detection

7.5 Response Time

If an external interrupt is recognized, its corresponding request flag is set at S5P2 in every machine cycle. The value is not polled by the circuitry until the next machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be next instruction to be executed. The call itself takes two cycles. Thus a minimum of three complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

A longer response time would be obtained if the request was blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or a write access to registers IE or IP the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction, if the instruction is MUL or DIV).

Thus a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

8 Power Saving Modes

The C511/513 microcontroller provide two power saving modes:

- Idle mode
- Power-down mode.

In the idle mode only the CPU will be deactivated, whereas in the power-down mode all operations will be stopped.

The bits PDE and IDLE, located in SFR PCON, select the power-down mode or the idle mode, respectively.

If the power-down mode and the idle mode are set at the same time, power-down takes precedence.

Furthermore, register PCON contains two general purpose flags. For example, the flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during the idle mode. An instruction that activates the idle mode also can set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

Special Function Register (Address 87_H) **Reset Value : $0XXX0000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
87_H	SMOD	–	–	–	GF1	GF0	PDE	IDLE	PCON

Symbol	Position	Function
SMOD	PCON.7	Baud rate doubled When set, the baud rate of the serial channel in mode 1,2,3 is doubled.
–	PCON.6-4	Not implemented. Reserved for future use.
GF1	PCON.3	General purpose flag
GF0	PCON.2	General purpose flag
PDE	PCON.1	Power-down enable bit When set, power-down mode is entered.
IDLE	PCON.0	Idle mode enable bit When set, idle mode is entered.

8.1 Idle Mode

In the idle mode the oscillator of the C511/513 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, and all timers are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption, which can be achieved by this feature depends on the number of peripherals running.

If all timers are stopped and the serial interface is not running, the maximum power reduction can be achieved. This state is also the test condition for the idle mode I_{CC}.

So the user has to take care which peripheral should continue to run and which has to be stopped during idle mode. Also the state of all port pins – either the pins controlled by their latches or controlled by their secondary functions – depends on the status of the controller when entering idle mode.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to drive the output during idle mode if the assigned function is on. This applies also to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and PSEN are held at logic high levels.

Table 8-1
Status of External Pins During Idle and Power-Down Mode

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	**Power-Down**	**Idle**	**Power-Down**
ALE	High	Low	High	Low
$\overline{\text{PSEN}}$	High	Low	High	Low
Port 0	Data	Data	Address	Float
Port 1	Data/alternate outputs	Data/alternate outputs	Data/alternate outputs	Data/alternate outputs
Port 2	Data	Data	Address	Data
Port 3	Data/alternate outputs	Data/last output	Data/alternate outputs	Data/last output

As in normal operation mode, the ports can be used as inputs during idle mode. Thus a capture or reload operation can be triggered, the timers can be used to count external events, and external interrupts will be detected.

The idle mode is a useful feature which makes it possible to "freeze" the processor's status - either for a predefined time, or until an external event reverts the controller to normal operation, as discussed below.

The idle mode is entered by setting the IDLE bit. Because PCON is not a bit-addressable register, the setting of the bit has to be obtained by byte-handling instructions, e.g. by:

```
ORL     PCON,#00000001B     ;Set IDLE bit
```

The instruction that sets bit IDLE is the last instruction executed before going into idle mode.

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and the instruction to be executed following the RETI instruction will be the one following the instruction that sets the bit IDLE.
- The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active at least for two machine cycles for a complete reset.

8.2 Power-Down Mode

In the power-down mode, the on-chip oscillator is stopped. Therefore all functions are stopped; only the contents of the on-chip RAM and the SFR's are maintained. The port pins controlled by their port latches output the values that are held by their SFR's. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode. ALE and PSEN hold at logic low level (see **table 8-1**).

The power-down mode is entered by setting the PDE bit. Because PCON is not a bit-addressable register, the setting of the bit has to be obtained by byte-handling instructions, e.g. by:

```
ORL     PCON,#00000010B     ;Set PDE bit
```

The instruction that sets bit PDE is the last instruction executed before going into power-down mode.

The only way to exit from power-down mode is a hardware reset. This reset will redefine all SFR's, but will not change the contents of the internal RAM.

In the power-down mode, V_{CC} can be reduced up to a minimum of 2 V to minimize power consumption. It must be ensured, however, that V_{CC} is not reduced before the power-down mode is invoked, and that V_{CC} is restored to its normal operating level, before the power-down mode is terminated. The reset signal that terminates the power-down mode also restarts the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (as with power-on reset).

9 EEPROM Programming Interface of the SAB-C513A-H

The SAB-C513A-H is the EEPROM MCU version in the SAB-C511/513 microcontroller family. Except the type of program memory, its functionality is equal to the SAB-C513A functionality. Therefore, the SAB-C513A-H can be used in a prototype system design as a replacement for all SAB-C511/513 family ROM-based microcontrollers.

This chapter describes in detail how the SAB-C513A-H is handled in the programmimg mode.

9.1 Programming Configuration

During normal program execution the SAB-C513A-H behaves like the SAB-C513A. For programming of the device, the SAB-C513A-H must be put into a programming mode. This typically is done not in-system but in a special programming hardware. In the programming mode the SAB-C513A-H operates as a typical peripheral device which allows the user to access it with its specific control and data registers.

In the programming mode the programming interface is accessed via a multiplexed address data bus (port 0) and via control lines at port 3. All other units of the microcontroller (CPU, timers, serial interfaces etc.) except the clock generator are disabled in the programming mode. The clock is needed for the operation of the charge pump which generates on-chip the necessary programming voltage for the EEPROM.

Figure 9-9-1 shows the pins of the SAB-C513A-H which are required to control the device in the programming mode. **Figure 9-9-2** shows the detailed pin configuration of the SAB-C513A-H.

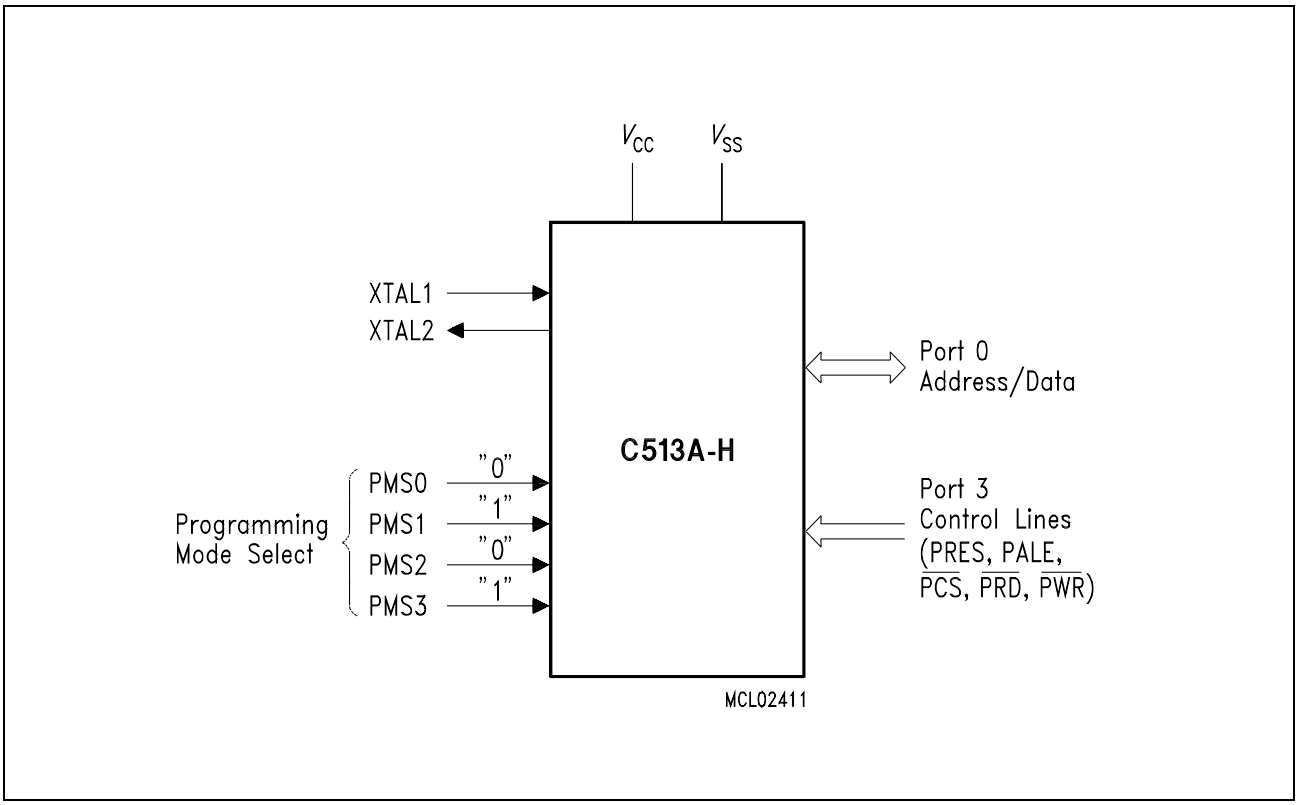


Figure 9-1
SAB-C513A-H Logic Symbol in Programming Mode

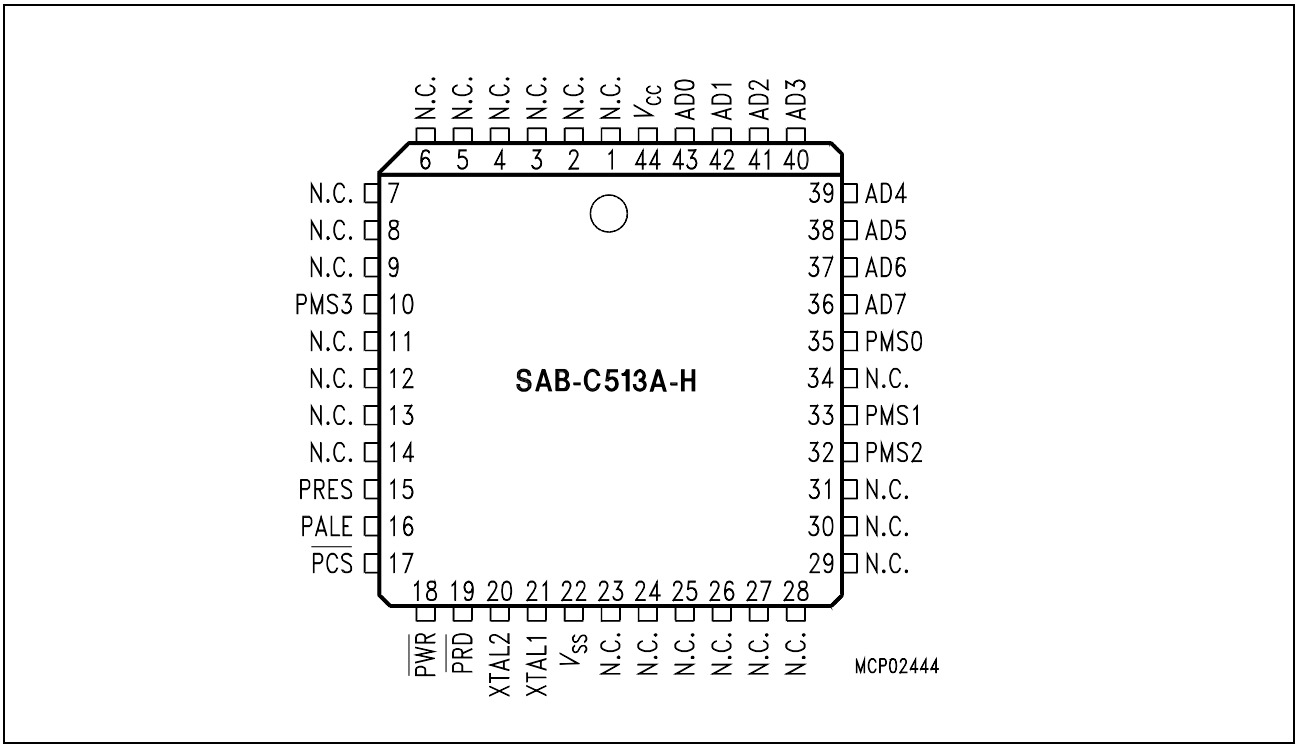


Figure 9-2
SAB-C513A-H Pin Configuration in Programming Mode

Table 9-1
Pin Definitions and Functions in Programming Mode

Symbol	Pin Number P-LCC-44	I/O*)	Function
PRES	15	I	**Programming Interface Reset** A high level on this input resets the programming interface and its registers to their initial state.
AD0 - AD7	43 - 36	I/O	**Bidirectional Address/Data Bus** AD0-7 is used to transfer data to and from the registers of the programming interface and to read the data of the memory field during EEPROM verification.
PALE	16	I	**Programming Address Latch Enable** This input is used to latch address information at AD0-7. The trailing edge of PALE is used to latch the register addresses. Each read or write access in programming mode must be initiated by a PALE high pulse.
$\overline{PRD}$	18	I	**$\overline{Programming\ Read\ Control}$** A low level at this pin (and $\overline{PCS}$=low) enables the AD0-7 buffers for reading of the data or control registers of the programming interface.

*) I = Input
O = Output

Table 9-1
Pin Definitions and Functions in Programming Mode (cont'd)

Symbol	Pin Number P-LCC-44	I/O*)	Function
$\overline{PWR}$	19	I	**$\overline{Programming\ Write\ Control}$** A low level at this pin (and $\overline{PCS}$=low and $\overline{PRD}$=high) causes the data at AD0-7 to be written into the data or control registers of the programming interface.
$\overline{PCS}$	17	I	**$\overline{Programming\ Chip\ Select}$** A low level at this pin enables the access to the registers of the programming interface. If $\overline{PCS}$ is active, either $\overline{PRD}$ or $\overline{PWR}$ control whether data is read or written into the registers. $\overline{PCS}$ should always be deactivated between subsequent accesses to the programming interface.
XTAL2	20	–	**XTAL2** Output of the inverting oscillator amplifier.
XTAL1	21	–	**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. During the device programming a clock must be always supplied.
PMS0 PMS1 PMS2 PMS3	35 33 32 10	I	**Programming Mode Select** PMS0-3 are used to put the SAB-C513A-H into the programming mode. In normal mode the programming mode select pins have the meaning as shown in the table below. PMS0-3 must be set to the logic level as described in the table below.
V_{SS}	22	–	**Circuit ground potential**
V_{CC}	44	–	**Power supply terminal** for all operating modes
N.C.	1-9, 11-14, 23-31, 24	–	**No connection** These pins must not be connected.

Normal Mode Pin Names	Progr. Mode Pin Names	Required Logic Level
$\overline{EA}$	PMS0	0
ALE	PMS1	1
$\overline{PSEN}$	PMS2	0
RESET	PMS3	1

*) I = Input
O = Output

9.2 Programming Interface

The programming interface of the SAB-C513A-H allows to erase, write and read the contents of the EEPROM. In the programming mode the device provides access to the programming interface registers via the multiplexed address/data bus, which is provided at the port 0 lines. Four lines are used for controlling of the access to the registers of the programming interface. The programming interface contains registers which hold the address and data of the EEPROM location(s) to be accessed and control registers to control the operating modes and the access.

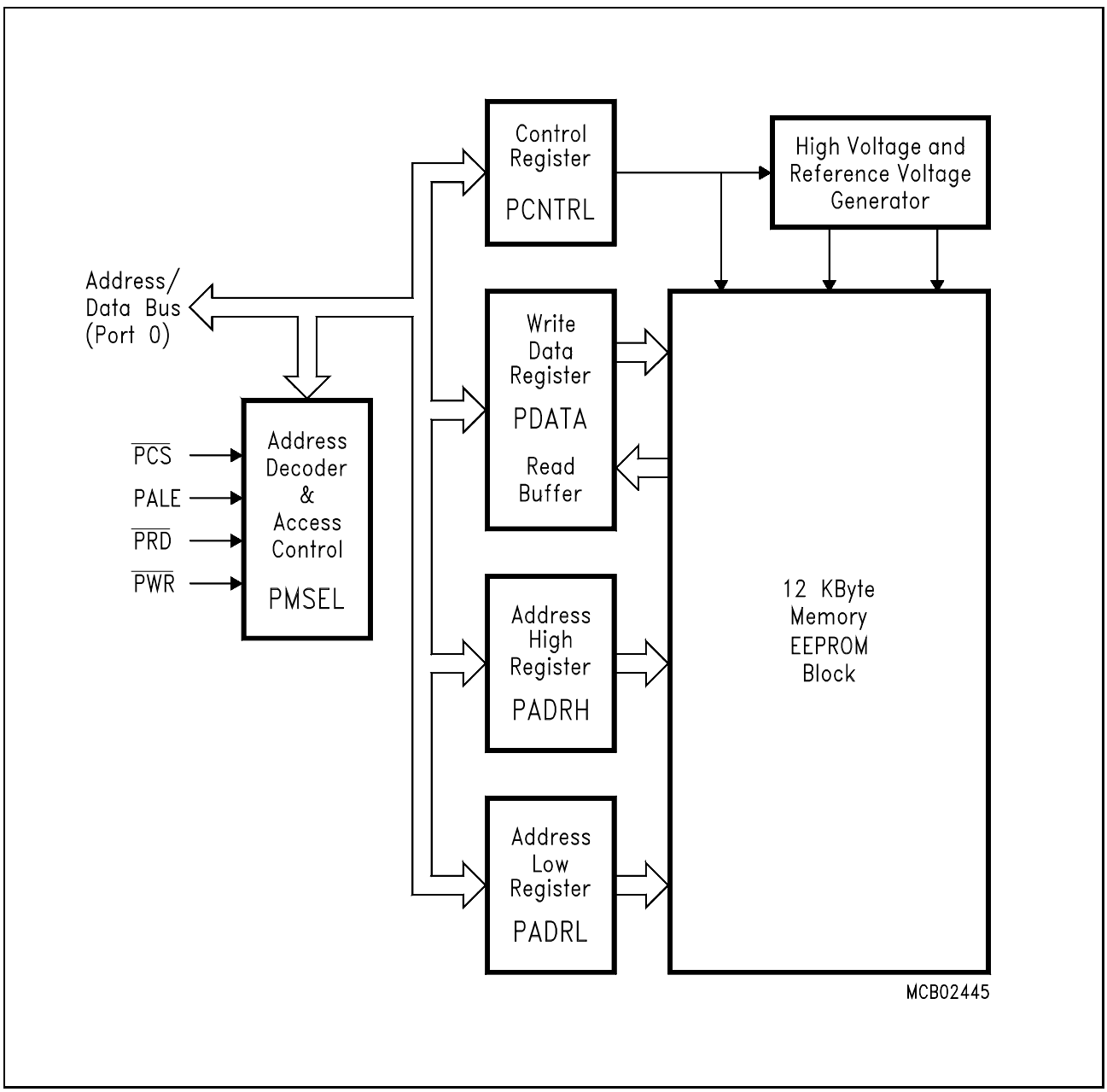


Figure 9-3
Block Diagram of the SAB-C513A-H Programming Interface

9.2.1 Selection of the Programming Mode

The programming mode is selected by applying specific logic levels to the four PMS0-3 pins of the SAB-C513A-H.

Table 9-2
Programming Mode Selection Pins

Mode	PMS3 (RESET)	PMS2 ($\overline{\text{PSEN}}$)	PMS1 (ALE)	PMS0 ($\overline{\text{EA}}$)
Normal Mode	1 or 0	none	none	0 or 1
Programming Mode	1	0	1	0

Note that PMS3 has to be active ('1') during the whole programming mode operation. During normal operation PMS2 and PMS1 are outputs ($\overline{\text{PSEN}}$, ALE) and no external driving circuitry should be connected to these pins.

It should be also noted, that always a clock has to be supplied in programming mode and that the operating voltage is restricted to 5 V ± 10%.

9.2.2 Resetting the Programming Interface

By applying a high level to the PRES pin the contents of the programming interface registers will be set to their initial values. When entering the programming mode, PRES should be held high.

The programming interface must not be reset while programming or erasing the EEPROM (PROG bit =1). The program or erase cycle should be finished properly before resetting the interface.

9.2.3 Interface Bus Operation

The programming interface with its registers can be accessed via a Intel type multiplexed bus interface. **Figures 9-9-4 and 9-9-5** illustrate a read and a write access to the registers. To enable access to the programming interface the enable input $\overline{\text{PCS}}$ must be active during the entire access cycle (from leading edge of ALE until trailing edge of $\overline{\text{RD}}$ or $\overline{\text{WR}}$). $\overline{\text{PCS}}$ should always be deactivated between subsequent accesses to the programming interface. With the falling edge of ALE the address of the register to be accessed is latched. The read signal puts the contents of the selected register onto the bus. During write cycles the write signal latches the data bus content into the selected register. Each read and write cycle must be initiated by a PALE high pulse.

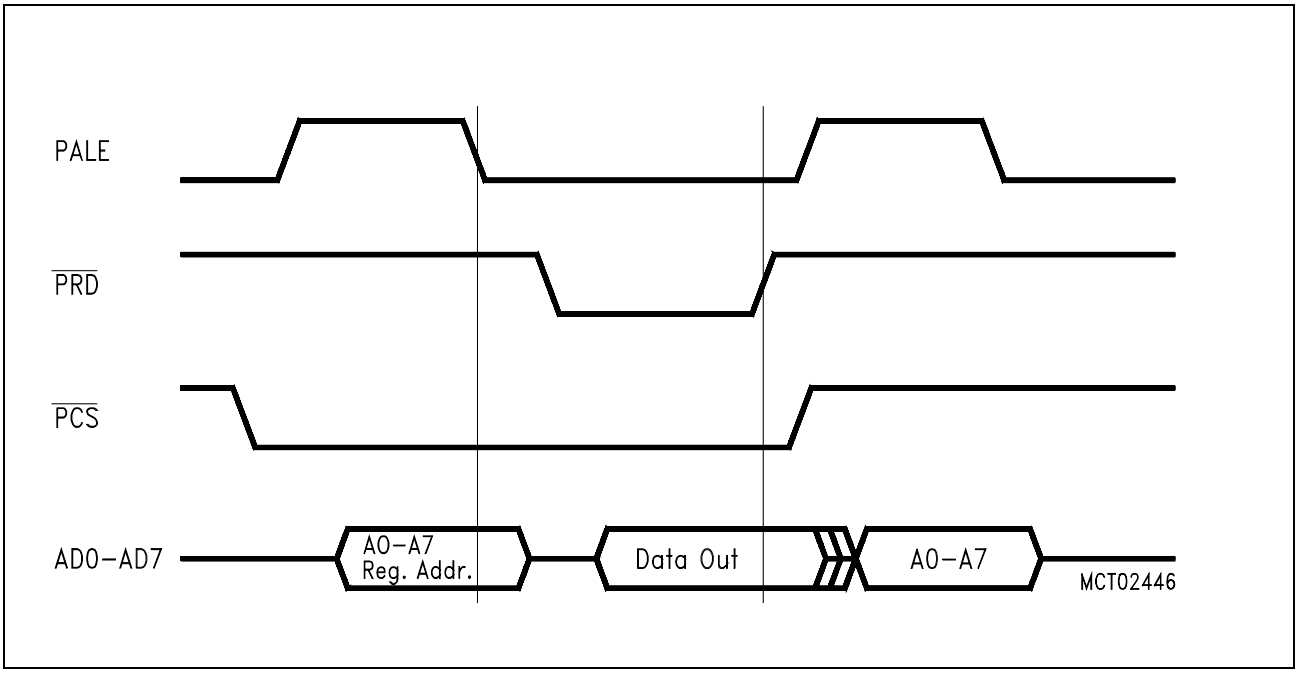


Figure 9-4
Program Interface Read Cycle

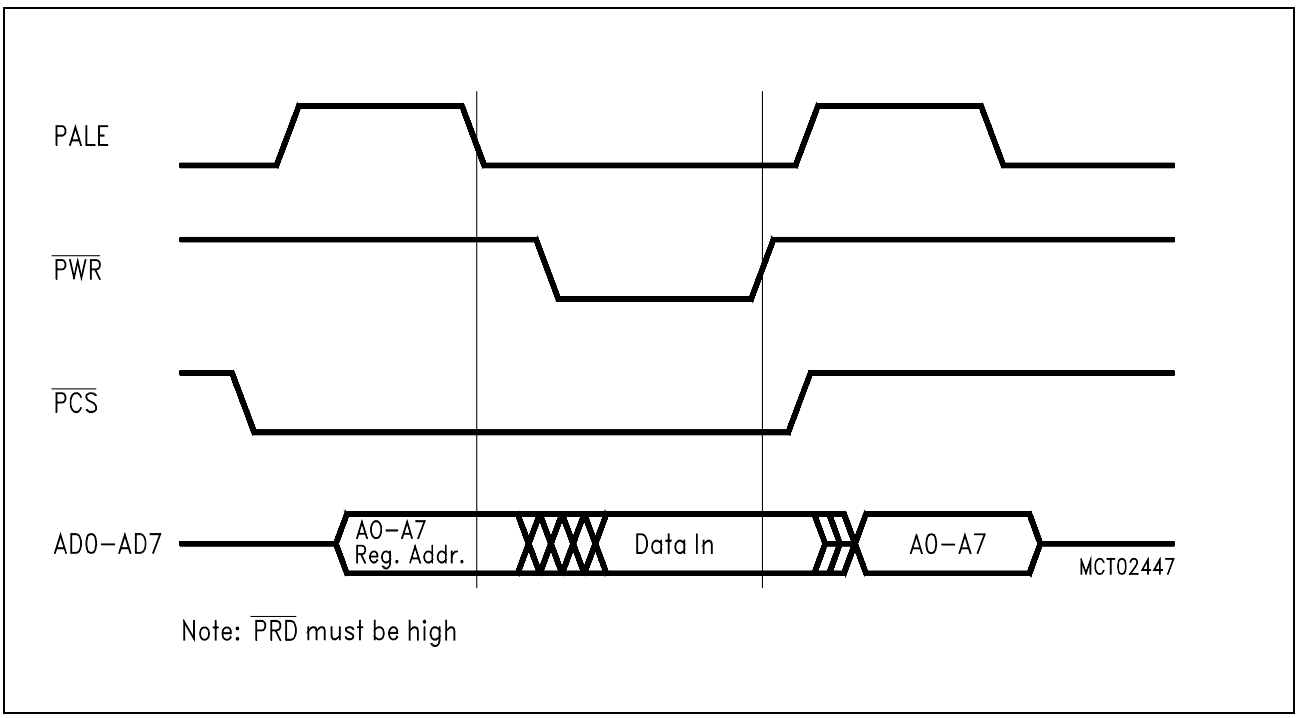


Figure 9-5
Program Interface Write Cycle

9.2.4 Programming Interface Registers

The programming interface contains five 8-bit registers, two for control and memory select modes, two for holding the address of the EEPROM cell or EEPROM block to be accessed and one for the data to be transferred.

Table 9-3
Programming Registers

Register	Address	Description
PCNTRL	00_H	Programming control register
PMSEL	01_H	Memory select register
PDATA	02_H	Read/write data register
PADDRL	03_H	LSB of memory address
PADDRH	04_H	MSB of memory address

9.2.4.1 Programming Control Register PCNTRL

This register holds the bits to control the access to the EEPROM memory and the programming process.

Programming Interface Register PCNTRL (Address 00_H) **Reset Value : $0XXX0000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
00_H	ADRI	–	–	–	EOM2	EOM1	EOM0	PROG	PCNTRL

Bit	Function
ADRI	Address Increment Enable If ADRI=1, the address in the address registers PADRH/PADRL is incremented when a read operation of PDATA is performed. If ADRI=0, the address is not modified.
–	Reserved Bits These bits must be written with ’0’. During reads these bits will be undefined.
EOM 2 - EOM 0	EEPROM Operation Mode These bits define the operation to be performed on the selected memory byte or memory block.

EOM2	EOM1	EOM0	Operation
0	0	0	NOP
0	0	1	Erase (clear) selected memory bytes/blocks
0	1	0	Write selected byte(s) with the content of PDATA
1	0	0	Read content of the addressed memory byte (address in PADRH/PADRL)

All other combinations of EOM2-0 are not allowed. Their use might lead to unpredictable results.

PROG	Program Voltage Enable If PROG=1, the programming voltage is turned on if EOM2-0=001_B or 010_B. If EOM2-0 is set to any other value, the programming voltage is inhibited. If PROG=0, the programming voltage is turned off.

9.2.4.2 Memory Select Register PMSEL

This register specifies which EEPROM memory cells or memory blocks will be accessed during a write or delete operation. The smallest amount of memory that can be accessed at a time is one byte, the largest is the whole memory. For read operations the settings of this register have no effect.

Programming Interface Register PMSEL (Address 01_H) **Reset Value : $XXX00000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
01_H	–	–	–	SEL4	SEL3	SEL2	SEL1	SEL0	PMSEL

Bit	Function
–	Reserved Bits These bits must be written with ’0’. During reads these bits will be undefined.
SEL4 - 0	Select Mode Bits These bits allow to define various granularities for writing or deleting specific memory areas. Single byte, 64 byte or 1KB byte blocks as well as even/odd addresses in 64 byte/1K byte blocks can be selected according the table below.

SEL4 - 0	Mode
0 0 0 0 0	The address in PADRH/PADRL specifies the byte to be accessed.
0 0 0 0 1	The complete 1KB block which contains the byte addressed by PADRH/PADRL is selected.
0 0 0 1 0	The complete 64 byte block which contains the byte addressed by PADRH/PADRL is selected.
0 0 1 0 1	If A0=0 all even addressed bytes, if A0=1 all odd addressed bytes of the1KB block which contains the byte addressed by PADRH/PADRL are selected.
0 0 1 1 0	If A0=0 all even addressed bytes, if A0=1 all odd addressed bytes of the 64 byte block which contains the byte addressed by PADRH/PADRL are selected.
1 1 0 0 1	With this combination of SEL4-0 the total 12K memory is selected.This combination is useful for fast erase/fill operations of the EEPROM memory.
1 1 1 0 1	If A0=0 all even addressed bytes, if A0=1 all odd addressed bytes of the total 12K memory are selected.

All other combination of SEL4-0 are not defined and might lead to unpredictable results.

9.2.4.3 Data Register PDATA

This register contains the data to be written into the selected memory bytes during write operations. When reading from this address, the contents of the memory byte selected by PADRH/PADRL will be put onto the data bus. The written content of PDATA will not be disturbed by reading PDATA.

Programming Interface Register PDATA (Address 02_H) **Reset Value : XX_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
02_H	D7	D6	D5	D4	D3	D2	D1	D0	PDATA

9.2.4.4 Address Low Register PADRL and PADRH

The PADRH/PADRL registers contain the 14-bit address of the EEPROM byte to be accessed during read or write operations. The LSB part of the address is stored in PADRL and the MSB part of the address is stored in PADRH. PADRL is combined with PADRH to form the complete 14-bit address.

Depending on the settings in the register PMSEL (SEL4-0 bits) the combined address also may specify memory blocks for write/erase operations. If automatic address increment is selected the combined address will be incremented by 1 after every read access.

Reading the PARDH/PADRL registers will put its actual value onto the data bus. Bits 6 and 7 of PADRH must be written with ’0’. During reads these bits will be undefined.

Programming Interface Registers PADRL/PADRH (Address 03_H/04_H) **Reset Values : XX_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
03_H	A7	A6	A5	A4	A3	A2	A1	A0	PADRL
04_H	–	–	A13	A12	A11	A10	A9	A8	PADRH

9.2.5 EEPROM Access Operations

Three different operations are provided for programming and verifying the EEPROM memory contents:

- Erase (delete, clear) memory content(s)
- Write (program) memory content(s)
- Read memory content.

All operations are byte oriented, however, for deleting and writing the memory more than one byte can be selected (see PMSEL register description). This allows quick erasure of memory blocks or the complete memory as well as efficient testing of the memory. Read operations only will operate on single bytes.

9.2.5.1 Erase Operation

The erase operation clears the total EEPROM memory or a selected area of it to an initial state. In this initial state all memory locations have a FF_H as content. As in an EPROM, a program operation can only write a “0” to a bit of an EEPROM cell, not vice versa. Therefore, before programming of the EEPROM, the bytes to be programmed should be erased.

At the erase operation the following steps must be executed:

1. Selection of the memory area (block or single byte) to be erased by programming PMSEL.
2. Writing the address of the byte/block to be cleared into PADRH and PADRL.
3. Set EEPROM operation mode bits in PCNTRL to EOM2-0=001_B and the PROG bit to 1.
4. Wait at least 10 ms.
5. Reset the PROG bit to 0 with still EOM2-0=001_B.
6. Wait 20 µs (guard time for discharging high voltage generator).
7. Set EOM2-0=000_B and PROG=0.

The steps of the erase operation are illustrated in **figure 9-9-6**.

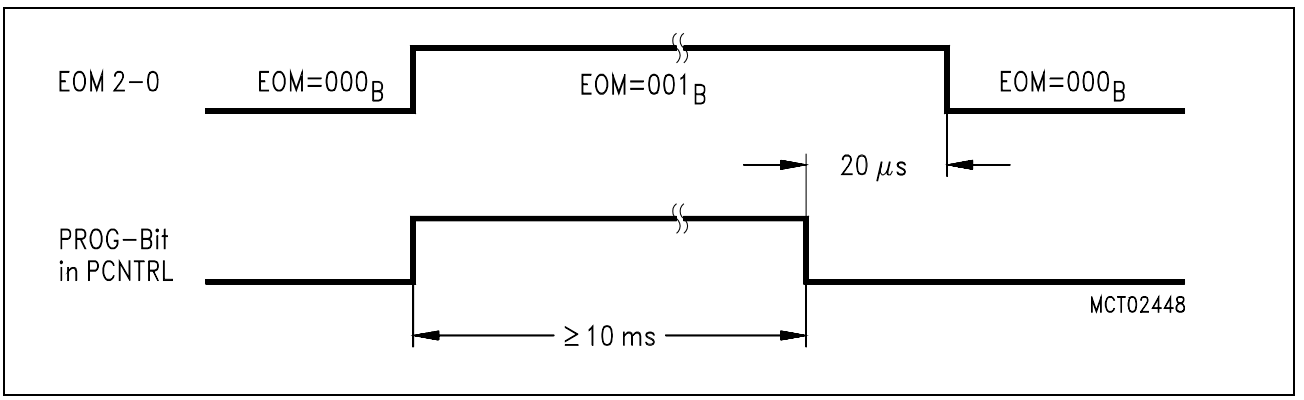


Figure 9-6
Erase Sequence

After erasing a memory byte contains FF_H.

9.2.5.2 Write Operation

The following sequence of operations is necessary for writing (programming) bytes or memory blocks:

1. Writing the data byte to be programmed into the register PDATA.
2. Selection of the memory area (block or single byte) to be written by programming PMSEL.
3. Writing the address of the byte/block to be programmed into PADRH and PADRL.
4. Set EEPROM operation mode bits in PCNTRL to EOM2-0=010_B and the PROG bit to 1.
5. Wait at least 10 ms.
6. Reset the PROG bit to 0 with still EOM2-0=010_B.
7. Wait 20 µs (guard time for discharging high voltage generator).
8. Set EOM2-0=000_B and PROG=0.

The steps of the program operation are illustrated in **figure 9-9-7**.

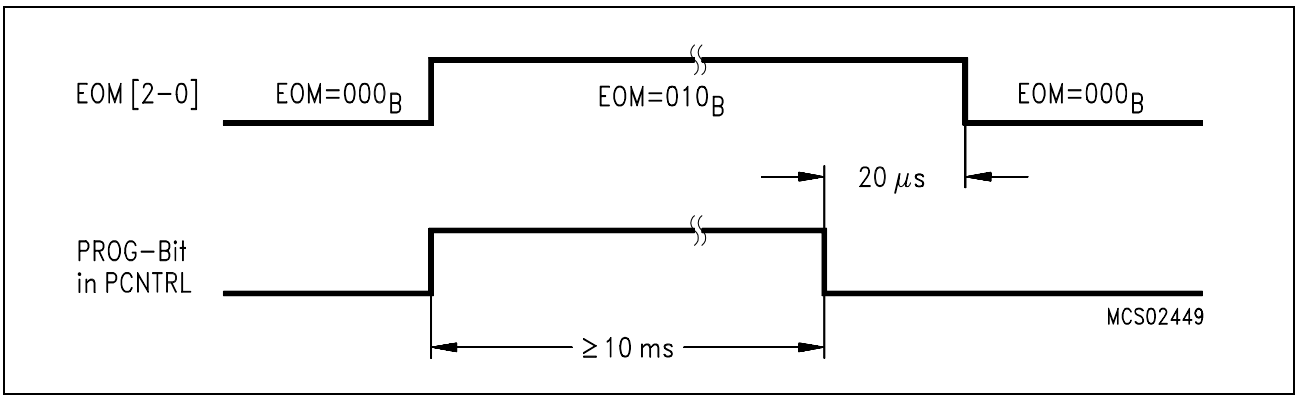


Figure 9-7
Write (Programming) Sequence

Before writing (programming) a memory location it should be erased (=FF_H), otherwise the old data would be “and-ed’ with the new data and the result being programmed.

9.2.5.3 Read Operation

The following sequence of operations is necessary for reading of a byte:

1. Writing the address of the byte to be read into PADRH and PADRL.
2. Set EEPROM operation mode bits in PCNTRL to EOM2-0=100_B and select whether automatic address increment should be performed (ADRI=1) or not (ADRI=0)
3. Perform a read from register PDATA. The content of the selected memory location will be transferred during this read.

Read operations can be done on the byte selected by PADRH/PADRL only. A requested block operation (SEL bits in PMSEL) will be ignored.

The automatic address increment mode reduces the overhead when reading out the memory but also when programming subsequent locations: when doing the read for verifying correct programming, the address is automatically incremented and points to the next location to be programmed.

10 Device Specifications

10.1 Absolute Maximum Ratings

Ambient temperature under bias (T_A) ..... 0 ˚C to + 70 ˚C
Storage temperature (T_{ST}).....– 65 ˚C to + 150 ˚C

Voltage on V_{CC} pins with respect to ground (V_{SS}) .....– 0.5 V to 6.5 V
Voltage on any pin with respect to ground (V_{SS}).....– 0.5 V to V_{CC} + 0.5 V

Input current on any pin during overload condition.....– 10 mA to + 10 mA
Absolute sum of all input currents during overload condition .....| 100 mA |

Power dissipation.....TBD

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of 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 for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.*

10.2 DC Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V; T_A = 0 to + 70 ˚C

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{EA}$, RESET)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage (RESET)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except $\overline{EA}$, RESET, XTAL1)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL1	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	
Input high voltage to $\overline{EA}$, RESET	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage Ports 1, 2, 3 (except P1.2, P1.4) Port 0, ALE, $\overline{PSEN}$ P1.2 / P1.4 pull-down transistor resistance	 V_{OL} V_{OL1} R_{DSon}	 – – –	 0.45 0.45 120	 V V Ω	 I_{OL} = 1.6 mA [1] I_{OL} = 3.2 mA [1] V_{OL} = 0.45 V
Output high voltage Ports 1, 2, 3 Port 0 in ext. bus mode, ALE, $\overline{PSEN}$ P1.2 / P1.4 pull-up transistor resistance	 V_{OH} V_{OH1} R_{DSon}	 2.4 0.9 V_{CC} 2.4 0.9 V_{CC} –	 – – – – 120	 V V V V Ω	 I_{OH} = – 80 µA I_{OH} = – 10 µA I_{OH} = – 800 µA I_{OH} = – 80 µA V_{OH} = 0.9 V_{CC}
Logic 0 input current (Ports 1, 2, 3)	I_{IL}	– 10	– 50	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (Ports 1, 2, 3)	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Maximum output low current per pin (Ports 0, 1, 2, 3)	I_{OLM}	–	5	mA	$V_{OL} \leq$ 1 V
Maximum output low current per port	I_{PL}	–	30	mA	–
Input leakage current Port 0 (if $\overline{EA}$=0), $\overline{EA}$, P1.2, P1.3, P1.5 as SSC inputs	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance [7]	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 ˚C

DC Characteristics (cont'd)

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V; T_A = 0 to + 70 ˚C

Parameter	Symbol	Limit Values		Unit	Test Condition
		typ. [8]	max.		
Power supply current: C511/C511A/C513/C513A					
Active mode, 12 MHz [6]	I_{CC}	7	9.5	mA	V_{CC} = 5 V,[4]
Idle mode, 12 MHz [6]	I_{CC}	3.5	4.5	mA	V_{CC} = 5 V,[5]
Power Down Mode	I_{PD}	TBD	50	µA	V_{CC} = 2 … 5.5 V,[3]
C513A-H					
Active mode, 12 MHz [6]	I_{CC}	16	TBD	mA	V_{CC} = 5 V,[4]
Idle mode, 12 MHz [6]	I_{CC}	6	TBD	mA	V_{CC} = 5 V,[5]
Power Down Mode	I_{PD}	TBD	50	µA	V_{CC} = 2 … 5.5 V[3]

Notes:

1) Capacitive loading on ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL} of ALE and port 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operation. In the worst case (capacitive loading > 100 pF), the noise pulse on ALE line may exceed 0.8 V. In such cases it may be desirable to qualify ALE with a schmitt-trigger, or use an address latch with a schmitt-trigger strobe input.

2) Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall bellow the 0.9 V_{CC} specification when the address lines are stabilizing.

3) I_{PD} (Power Down Mode) is measured under following conditions:
$\overline{EA}$ = Port0 = V_{CC}; RESET = V_{SS}; XTAL2 = N.C.; XTAL1 = V_{CC}; all other pins are disconnected.

4) I_{CC} (active mode) is measured with:
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
$\overline{EA}$ = Port0 = RESET = V_{CC}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

5) I_{CC} (Idle mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL2 = N.C.;
RESET = $\overline{EA}$ = V_{SS}; Port0 = V_{CC}; all other pins are disconnected;

6) $I_{CC\ Max}$ at other frequencies is given by:

C511/C511A/C513/C513A :	Active mode:	TBD
	Idle mode:	TBD
C513A-H :	Active mode:	TBD
	Idle mode:	TBD

where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V.

7) This parameter is periodically sampled and not 100% tested.

8) The typical I_{CC} values are periodically measured at T_A = +25 ˚C but not 100% tested.

10.3 AC Characteristics (applies to all C511/513 Family Microcontrollers)

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V T_A = 0 ˚C to + 70 ˚C

(C_L for port 0, ALE and $\overline{\text{PSEN}}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

10.3.1 Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	127	–	$2t_{CLCL}$ – 40	–	ns
Address setup to ALE	t_{AVLL}	43	–	t_{CLCL} – 40	–	ns
Address hold after ALE	t_{LLAX}	60	–	t_{CLCL} – 23	–	ns
ALE low to valid instr in	t_{LLIV}	–	233	–	$4t_{CLCL}$ – 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	58	–	t_{CLCL} – 25	–	ns
$\overline{\text{PSEN}}$ pulse width	t_{PLPH}	215	–	$3t_{CLCL}$ – 35	–	ns
$\overline{\text{PSEN}}$ to valid instr in	t_{PLIV}	–	150	–	$3t_{CLCL}$ – 100	ns
Input instruction hold after $\overline{\text{PSEN}}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{\text{PSEN}}$	t_{PXIZ}*)	–	63	–	t_{CLCL} – 20	ns
Address valid after $\overline{\text{PSEN}}$	t_{PXAV}*)	75	–	t_{CLCL} – 8	–	ns
Address to valid instr in	t_{AVIV}	–	302	–	$5t_{CLCL}$ – 115	ns
Address float to $\overline{\text{PSEN}}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the C511/513 microcontrollers to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

10.3.2 External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	400	–	$6t_{CLCL} - 100$	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	400	–	$6t_{CLCL} - 100$	–	ns
Address hold after ALE	t_{LLAX2}	132	–	$2t_{CLCL} - 35$	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	252	–	$5t_{CLCL} - 165$	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	97	–	$2t_{CLCL} - 70$	ns
ALE to valid data in	t_{LLDV}	–	517	–	$8t_{CLCL} - 150$	ns
Address to valid data in	t_{AVDV}	–	585	–	$9t_{CLCL} - 165$	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	203	–	$4t_{CLCL} - 130$	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	33	–	$t_{CLCL} - 50$	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	433	–	$7t_{CLCL} - 150$	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	33	–	$t_{CLCL} - 50$	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

10.3.3 SSC Interface Characteristics

Parameter	Symbol	Limit Values		Unit
		12 MHz Clock		
		min.	max.	
Clock Cycle Time : Master Mode Slave Mode	t_{SCLK} t_{SCLK}	666 600	– –	ns ns
Clock high time	t_{SCH}	250	–	ns
Clock low time	t_{SCL}	250	–	ns
Data output delay	t_{D}	–	100	ns
Data output hold	t_{HO}	0	–	ns
Data input setup	t_{S}	100	–	ns
Data input hold	t_{HI}	100	–	ns
TC bit set delay	t_{DTC}	–	16 t_{CLCL}	ns

10.3.4 External Clock Characteristics

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 12 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	83.3	285	ns
High time	t_{CHCX}	20	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	20	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

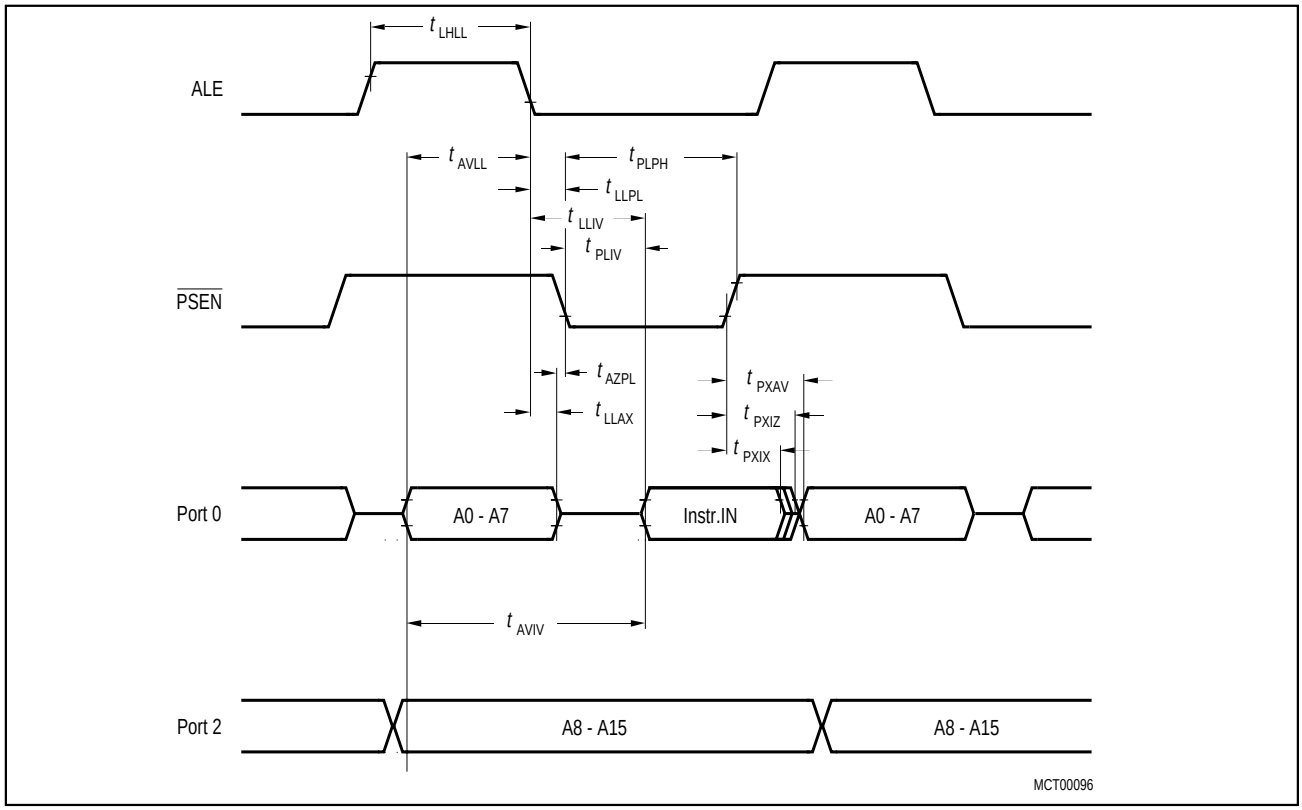


Figure 10-1
Program Memory Read Cycle

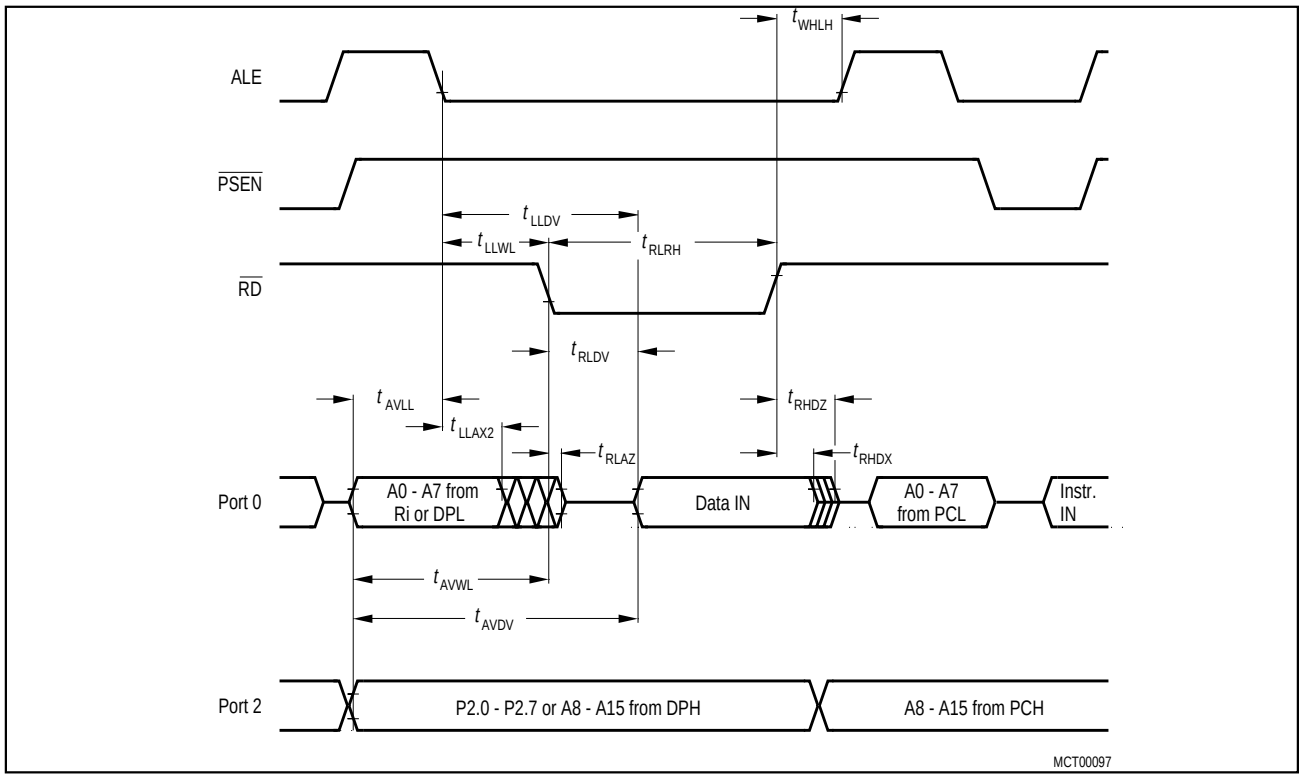


Figure 10-2
Data Memory Read Cycle

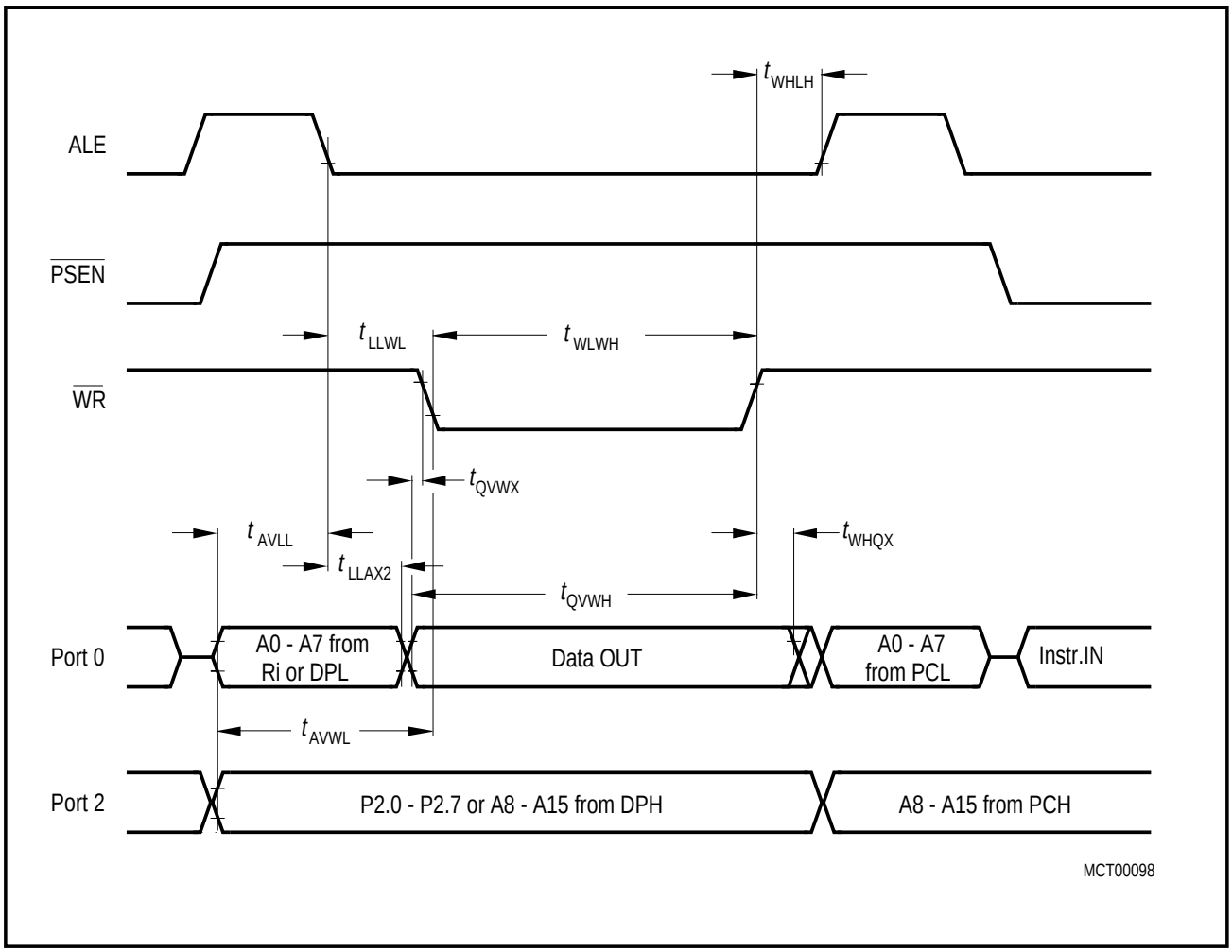
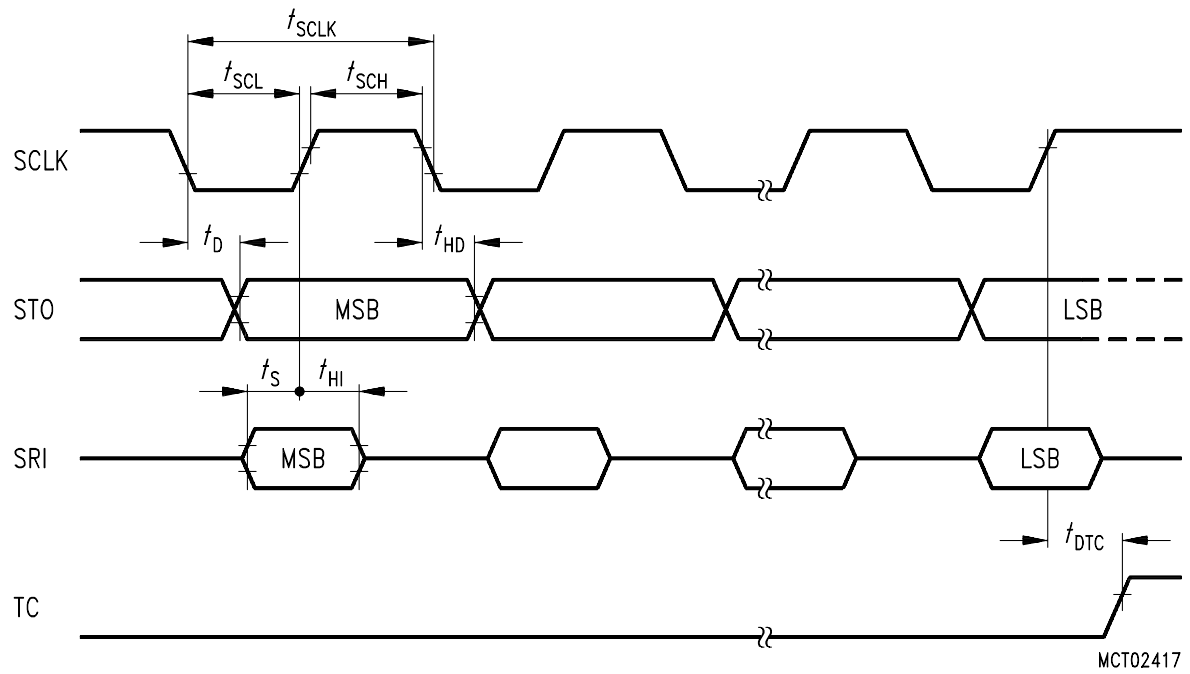


Figure 10-3
Data Memory Write Cycle



Notes: Shown is the data/clock relationship for CPOL = CPHA = 1. The timing diagram is valid for the other cases accordingly.

In the case of slave mode and CPHA = 0, the output delay for the MSB applies to the falling edge of $\overline{SLS}$ (if transmitter is enabled).

In the case of master mode and CPHA = 0, the MSB becomes valid after the data has been written into the shift register, i.e. at least one half SCLK clock cycle before the first clock transition.

Figure 10-4
SSC Timing

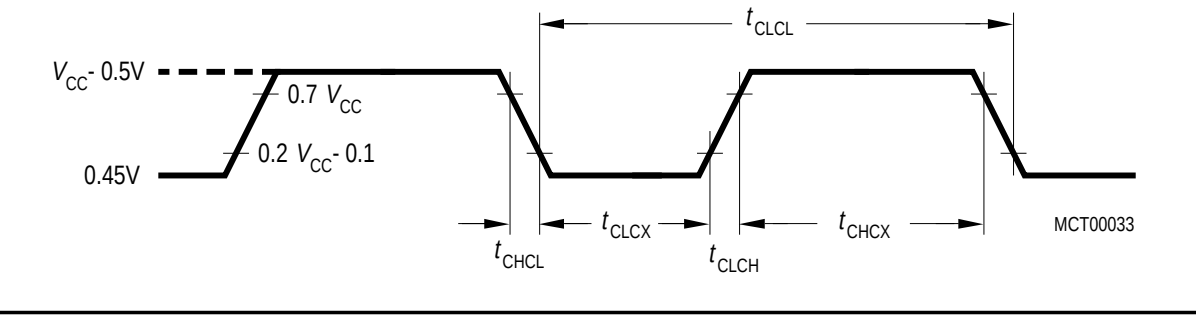
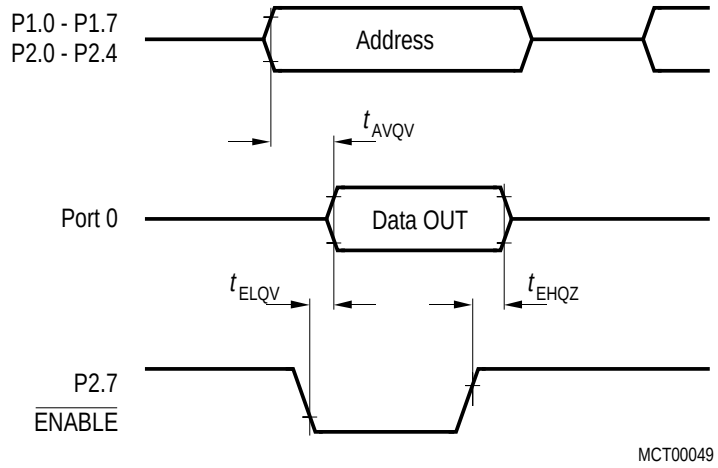


Figure 10-5
External Clock Drive at XTAL1

10.3.5 ROM Verification Characteristics (only C511/C511A/C513/C513A)

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	–	$48t_{CLCL}$	ns
ENABLE to valid data	t_{ELQV}	–	$48t_{CLCL}$	ns
Data float after ENABLE	t_{EHQZ}	0	$48t_{CLCL}$	ns
Oscillator frequency	$1/t_{CLCL}$	4	6	MHz



Address: P1.0 - P1.7 = A0 - A7
P2.0 - P2.4 = A8 - A12
Data: P0.0 - P0.7 = D0 - D7

Inputs: P2.5 - P2.6, $\overline{PSEN}$ = V_{SS}
ALE, $\overline{EA}$ = V_{IH}
$\overline{RESET}$ = V_{SS}

Device Type	ROM Size	Active Address Lines at Port 2	Inactive Address Lines at Port 2
C511	2.5 KB	P2.0 - P2.3 = A8 - A11	P2.4 - P2.6 = V_{SS}
C511A	4 KB	P2.0 - P2.3 = A8 - A11	P2.4 - P2.6 = V_{SS}
C513	8 KB	P2.0 - P2.4 = A8 - A12	P2.5 - P2.6 = V_{SS}
C513A	12/16 KB	P2.0 - P2.5 = A8 - A13	P2.6 = V_{SS}

Figure 10-6
ROM Verification Timing

10.4 AC Characteristics of C513A-H Programming Interface

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V; T_A = +25 °C 10 °C; 1/tCLCL = 8 MHz

Parameter	Symbol	Limit Values		Unit
		min.	max.	
ALE pulse width	t_{PLL}	60	–	ns
Address setup to ALE	t_{PAL}	20	–	ns
Address hold after ALE	t_{PLA}	20	–	ns
Address to valid data out	t_{PAD}	–	230	ns
$\overline{PRD}$/$\overline{PWR}$ pulse width	t_{PCC}	250	–	ns
$\overline{PRD}$ to valid data out	t_{PRDV}	–	200	ns
Data hold after $\overline{PWR}$	t_{PWDH}	0	–	ns
Data float after $\overline{PRD}$	t_{PDZ}	–	40	ns
Chip select setup to ALE active	t_{PCS}	0	–	ns
Chip select hold after $\overline{PRD}$/$\overline{PWR}$ inactive	t_{PCH}	0	–	ns
ALE to $\overline{PWR}$ or $\overline{PRD}$	t_{PLC}	90	–	ns
$\overline{PWR}$ or $\overline{PRD}$ high to ALE high	t_{PCL}	20	–	ns
Data setup before $\overline{PWR}$ rising edge	t_{PWDS}	50	–	ns
Data hold after $\overline{PWR}$ rising edge	t_{PWDH}	0	–	ns
Data float after $\overline{PCS}$	t_{PDF}	–	40	ns

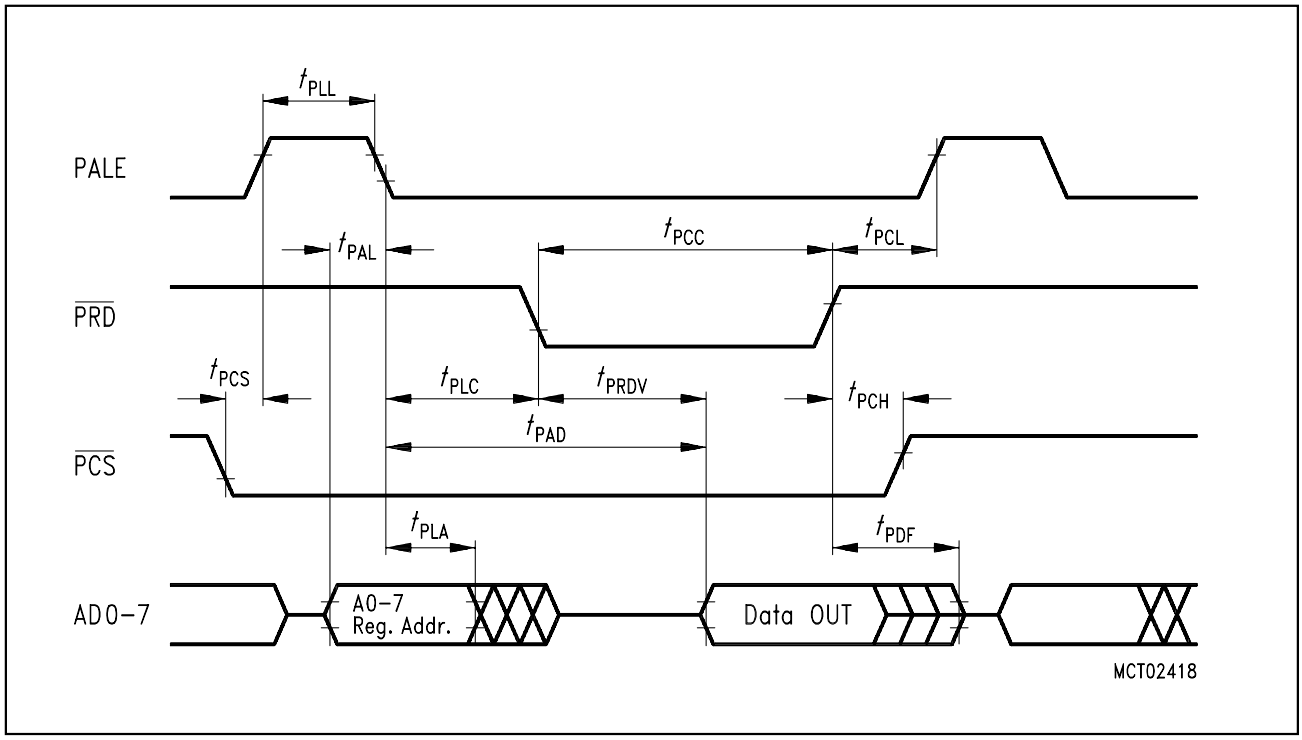


Figure 10-7
C513A-H Programming Interface Read Cycle

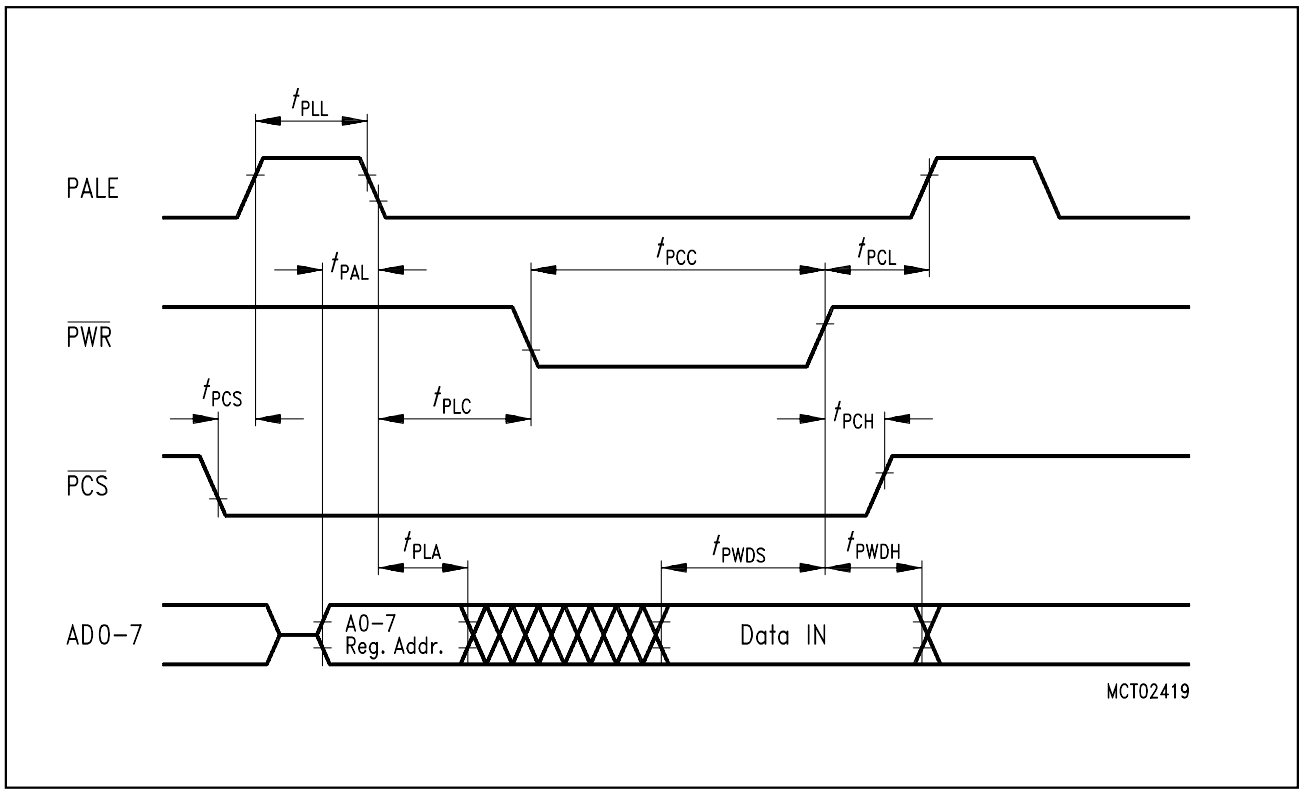


Figure 10-8
C513A-H Programming Interface Write Cycle

10.4.1 Reset Characteristics (C513A-H only)

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
RESET pulse width	t_{RLRH}	10	–	10	–	ms

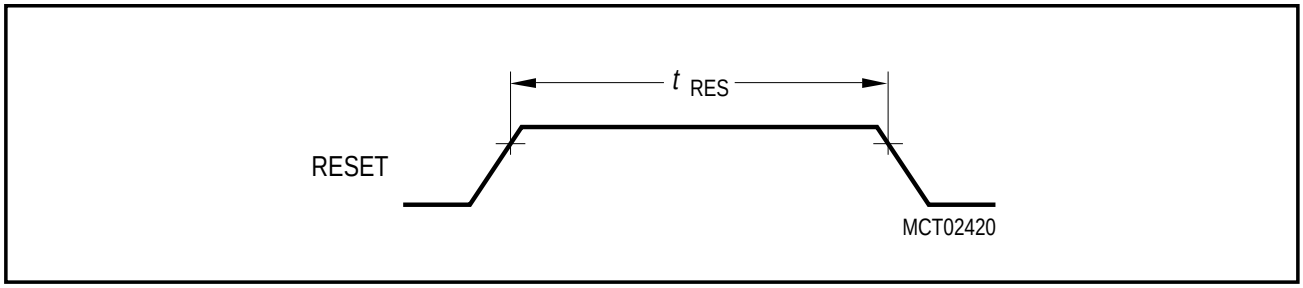
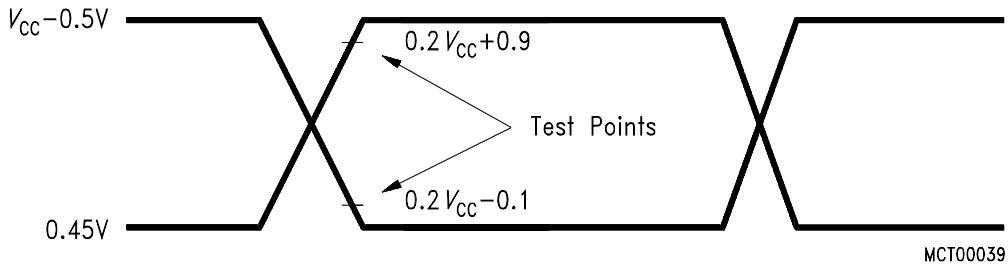
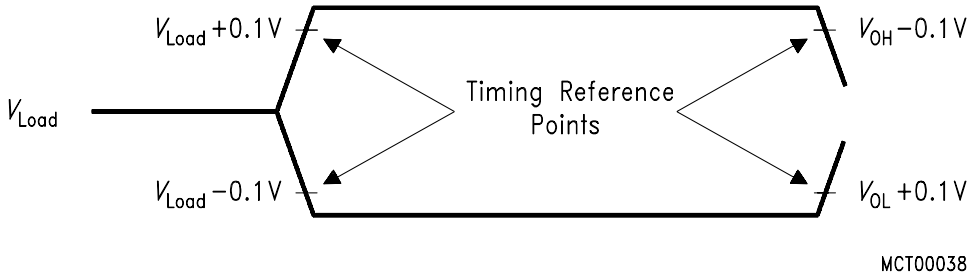


Figure 10-9
C513A-H Reset Pulse



AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at V_{IHmin} for a logic '1' and V_{ILmax} for a logic '0'.

Figure 10-10
AC Testing: Input, Output Waveforms



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA

Figure 10-11
AC Testing: Float Waveforms

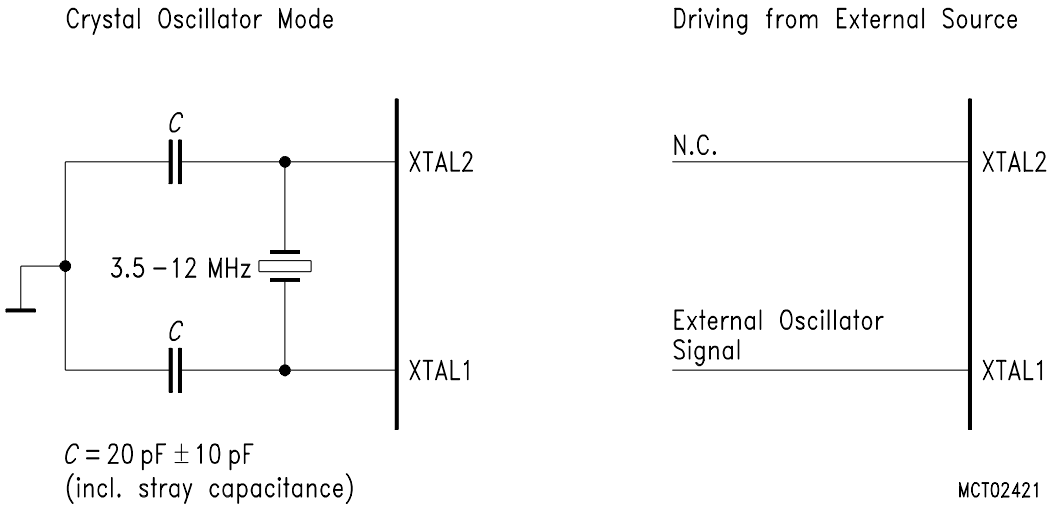
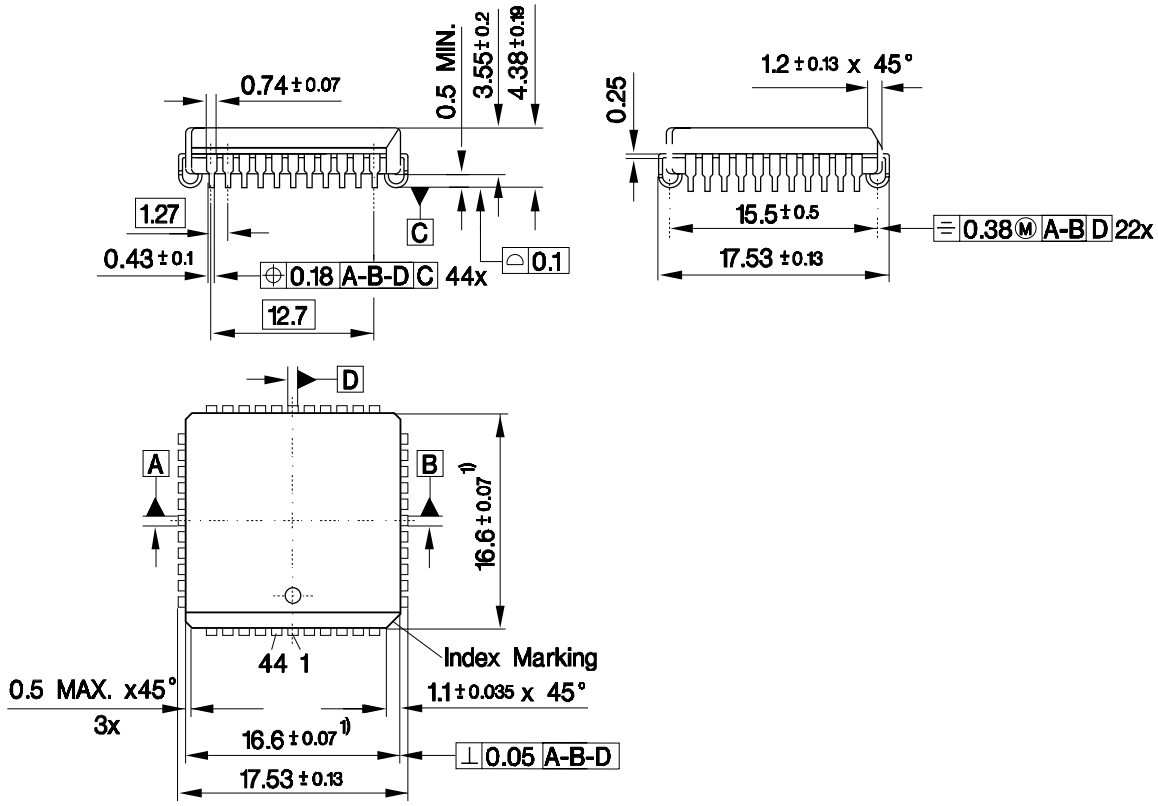


Figure 10-12
Recommended Oscillator Circuits for Crystal Oscillator

10.5 Package Outlines

Plastic Package, P-LCC-44-2 – SMD
(Plastic Leaded Chip-Carrier)



1) Does not include plastic or metal protrusion of 0.15 max. per side

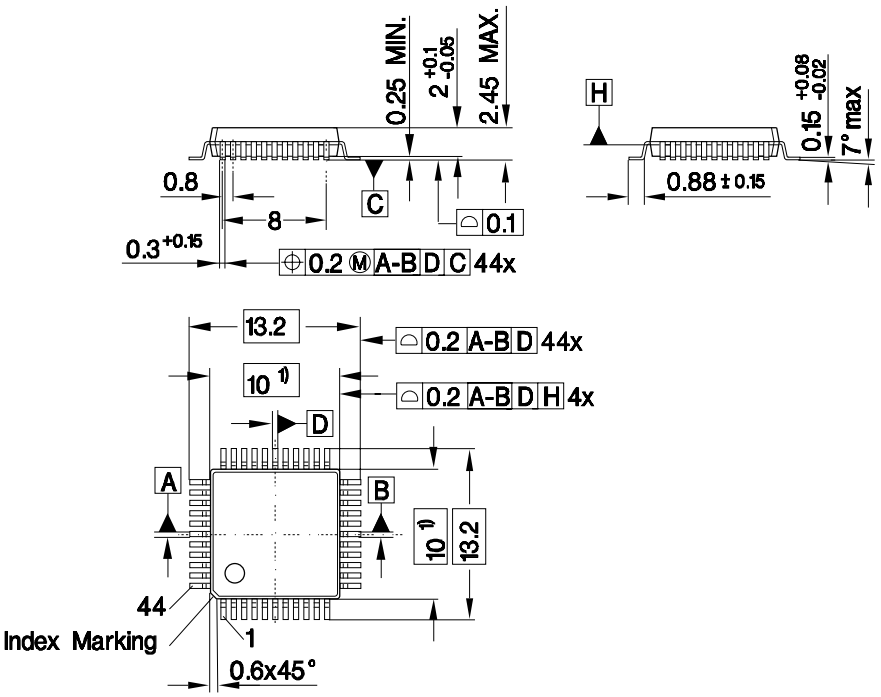
Dimensions in mm

Sorts of Packing
Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”

SMD = Surface Mounted Device

Dimensions in mm

Plastic Package, P-MQFP-44 – SMD
(Plastic Metric Quad Flat Package)



1) Does not include plastic or metal protrusion of 0.25 max. per side

Dimensions in mm

Sorts of Packing
Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”

SMD = Surface Mounted Device

Dimensions in mm