Infineon technologies

Microcomputer Components

8-bit CMOS Microcontroller

C513AO

User's Manual 05.99

http://www.infineon.com/

Edition 05.99

Published by
Infineon Technologies AG i. Gr.,
St.-Martin-Strasse 53
D-81541 München

© Infineon Technologies AG 1999
All Rights Reserved.

Attention please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Infineon Technologiesis an approved CECC manufacturer.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office in Germany or our Infineon Technologies Representatives worldwide (see address list).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

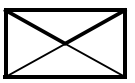
Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

C513AO **Revision History:**	**05.99**
Previous Releases:	-
Page	**Subjects**

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com



Enhanced Hooks Technology™ is a trademark and patent of Metalink Corporation licensed to Siemens.

Contents Page

1 Introduction . . . 1-1
1.1 Pin Configurations . . . 1-4
1.2 Pin Definitions and Functions . . . 1-7

2 Fundamental Structure . . . 2-1
2.1 Central Processing Unit (CPU) . . . 2-2
2.2 CPU Timing . . . 2-4

3 Memory Organization . . . 3-1
3.1 Program Memory, “Code Space” . . . 3-2
3.2 Data Memory, “Data Space” . . . 3-2
3.3 General Purpose Registers . . . 3-2
3.4 XRAM Operation . . . 3-3
3.4.1 Reset Operation of the XRAM . . . 3-3
3.4.2 Accesses to XRAM using the DPTR (16-bit Addressing Mode) . . . 3-3
3.4.3 Accesses to XRAM using the Registers R0/R1 (8-bit Addressing Mode) . . . 3-4
3.5 Special Function Registers . . . 3-5

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 Overlapping External Data and Program Memory Spaces . . . 4-3
4.3.1 Address Latch Enable (ALE) . . . 4-4
4.4 Enhanced Hooks Emulation Concept . . . 4-5
4.5 ROM/OTP Protection . . . 4-6
4.5.1 Unprotected ROM Mode . . . 4-6
4.5.2 Protected ROM/OTP Mode . . . 4-7

5 Reset and System Clock Operation . . . 5-1
5.1 Hardware Reset Operation . . . 5-1
5.2 Fast Internal Reset after Power-On . . . 5-3
5.3 Hardware Reset Timing . . . 5-5
5.4 Oscillator and Clock Circuit . . . 5-6

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 Timing . . . 6-12
6.1.5 Port Loading and Interfacing . . . 6-13
6.1.6 Read-Modify-Write Feature of Ports 2 and 3 . . . 6-14
6.2 Timers/Counters . . . 6-15
6.2.1 Timer/Counter 0 and 1 . . . 6-15
6.2.1.1 Timer/Counter 0 and 1 Registers . . . 6-16
6.2.1.2 Mode 0 . . . 6-19
6.2.1.3 Mode 1 . . . 6-20

Contents **Page**

6.2.1.4 Mode 2 6-21
6.2.1.5 Mode 3 6-22
6.2.2 Timer/Counter 2 6-23
6.2.2.1 Timer/Counter 2 Registers 6-24
6.2.2.2 Auto-Reload (Up or Down Counter) 6-26
6.2.2.3 Capture Mode 6-29
6.3 Serial Interface (USART) 6-30
6.3.1 Multiprocessor Communications 6-31
6.3.2 Serial Port Registers 6-31
6.3.3 Baudrates 6-33
6.3.3.1 Using Timer 1 to Generate Baudrates 6-35
6.3.3.2 Using Timer 2 to Generate Baudrates 6-36
6.3.4 Details about Mode 0 6-37
6.3.5 Details about Mode 1 6-40
6.3.6 Details about Modes 2 and 3 6-43
6.4 SSC Interface 6-46
6.4.1 General Operation of the SSC 6-47
6.4.2 Enable/Disable Control 6-47
6.4.3 Baudrate Generation (Master Mode only) 6-48
6.4.4 Write Collision Detection 6-48
6.4.5 Master/Slave Mode Selection 6-49
6.4.6 Data/Clock Timing Relationships 6-50
6.4.6.1 Master Mode Operation 6-50
6.4.6.2 Slave Mode Operation 6-51
6.4.7 Register Description 6-52

7 Interrupt System 7-1
7.1 Interrupt Structure 7-3
7.2 Interrupt Registers 7-4
7.2.1 Interrupt Enable Registers 7-4
7.2.2 Interrupt Request Flags 7-5
7.2.3 Interrupt Priority Registers 7-10
7.3 Interrupt Handling 7-12
7.4 External Interrupts 7-13
7.5 Interrupt Response Time 7-14

8 Fail Safe Mechanisms 8-1
8.1 Programmable Watchdog Timer 8-1
8.1.1 Refreshing the Watchdog Timer 8-4
8.1.2 Watchdog Reset and Watchdog Status Flag (WDTS) 8-4
8.2 Oscillator Watchdog Unit 8-5
8.2.1 Detailed Description of the Oscillator Watchdog Unit 8-6
8.2.2 Fast Internal Reset after Power-On 8-7

9 Power Saving Modes 9-1
9.1 Idle Mode 9-3
9.2 Slow-down Mode Operation 9-5
9.3 Power-down Mode 9-6
9.3.1 Invoking Power-Down Mode 9-6

Contents **Page**

9.3.2 Exit from Power-down Mode 9-7

10 OTP Memory Operation 10-1
10.1 Programming Configuration 10-1
10.2 Pin Configuration 10-2
10.3 OTP Programming Mode - Pin Definitions 10-5
10.4 OTP Programming Mode Selection 10-7
10.4.1 Basic Programming Mode Selection 10-7
10.4.2 OTP Memory Access Mode Selection 10-8
10.5 Program/Read OTP Memory Bytes 10-9
10.6 Lock Bits Programming / Read 10-11
10.7 Access of Version Bytes 10-13

11 Index 11-1

1 Introduction

The C513AO microcontroller is a member of the Infineon Technologies C500 family of 8-bit microcontrollers. It is an enhanced, upgraded version of the C513A 8-bit microcontroller and is fully compatible with the industry-standard 8051 microcontroller.

The C513AO-2R incorporates 16Kx8 non-volatile read-only program memory, 512x8 volatile read/write data memory, four 8-bit wide ports, three 16-bit timers/counters, an SPI-compatible synchronous interface, versatile fail-safe and power-saving mechanisms. The C513AO-2E is the One-Time Programmable (OTP) version of the C513AO microcontroller with a 16Kx8 OTP memory. The C513AO-L is the version without program memory. The term C513AO refers to all versions within the documentation unless otherwise noted.

Figure 1-1 shows the basic functional units of the C513AO. **Figure 1-2** shows the simplified logic symbol of the C513AO device.

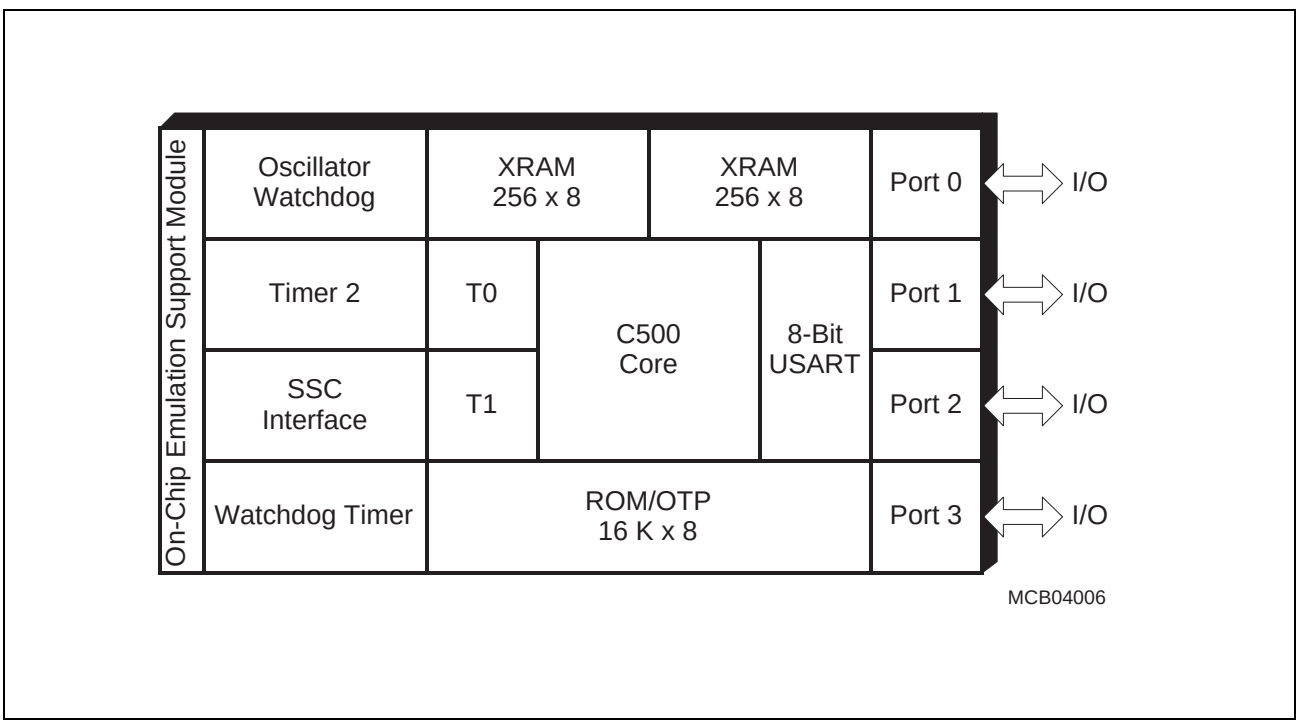


Figure 1-1
C513AO Functional Units

The following list summarizes the main features of the C513AO microcontroller:

- Fully compatible with the standard 8051 microcontroller
- Up to 16 MHz external operating frequency
  - 750 ns instruction cycle time at 16 MHz
- On-chip program memory
  - C513AO-2R: 16 Kbytes ROM (with optional ROM protection)
  - C513AO-2E: 16 Kbytes OTP
  - C513AO-L: version without on-chip program memory (ROMless)
- 256x8 RAM
- 256x8 XRAM
- Four 8-bit digital I/O ports
- Three 16-bit timers/counters (Timer 2 with Up/Down and 16-bit auto-reload features)
- Full duplex serial interface (USART)
- Synchronous Serial Channel (SSC)
- Seven interrupt sources with two priority levels
- On-chip emulation support logic (Enhanced Hooks Emulation Technology™ 1))
- Programmable 15-bit Watchdog Timer
- Oscillator Watchdog
- Fast Power On Reset
- Power Saving Modes
  - Slow-down mode
  - Idle mode
  - Software power-down mode with optional wake up capability through pin P3.2/$\overline{\text{INT0}}$
- Available in P-DIP40, P-LCC-44 and P-MQFP-44-2 packages
- Fully pin-compatible with C501, C504, C505C, C505CA and C511/C513-devices.
- Temperature ranges: SAB-C513AO T_A: 0 to 70 °C
  SAF-C513AO T_A: – 40 to 85 °C

1)“Enhanced Hooks Technology” is a trademark and patent of MetaLink Corporation licensed to Infineon Technologies.

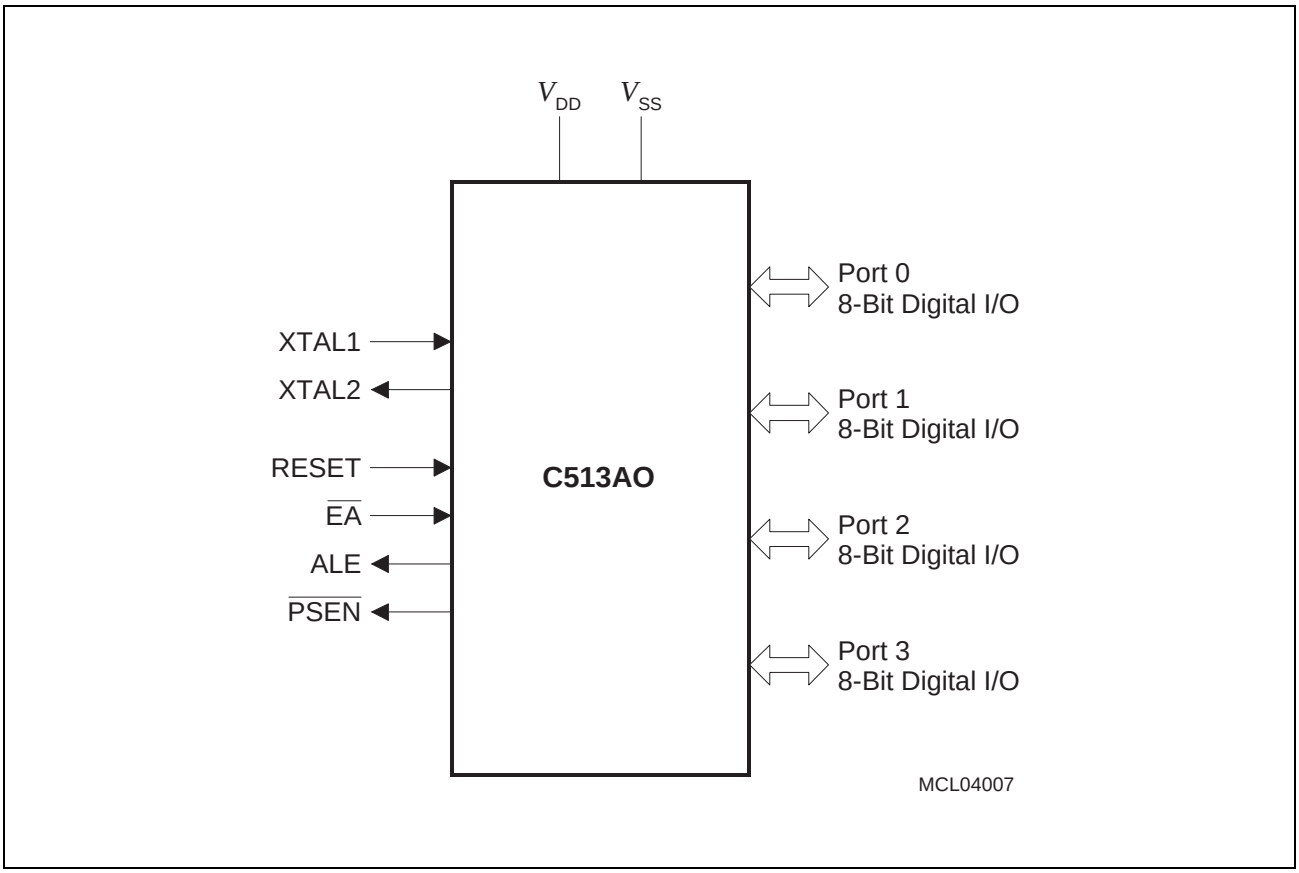


Figure 1-2
Logic Symbol

1.1 Pin Configurations

This section describes the pin configurations of the C513AO device in various packages.

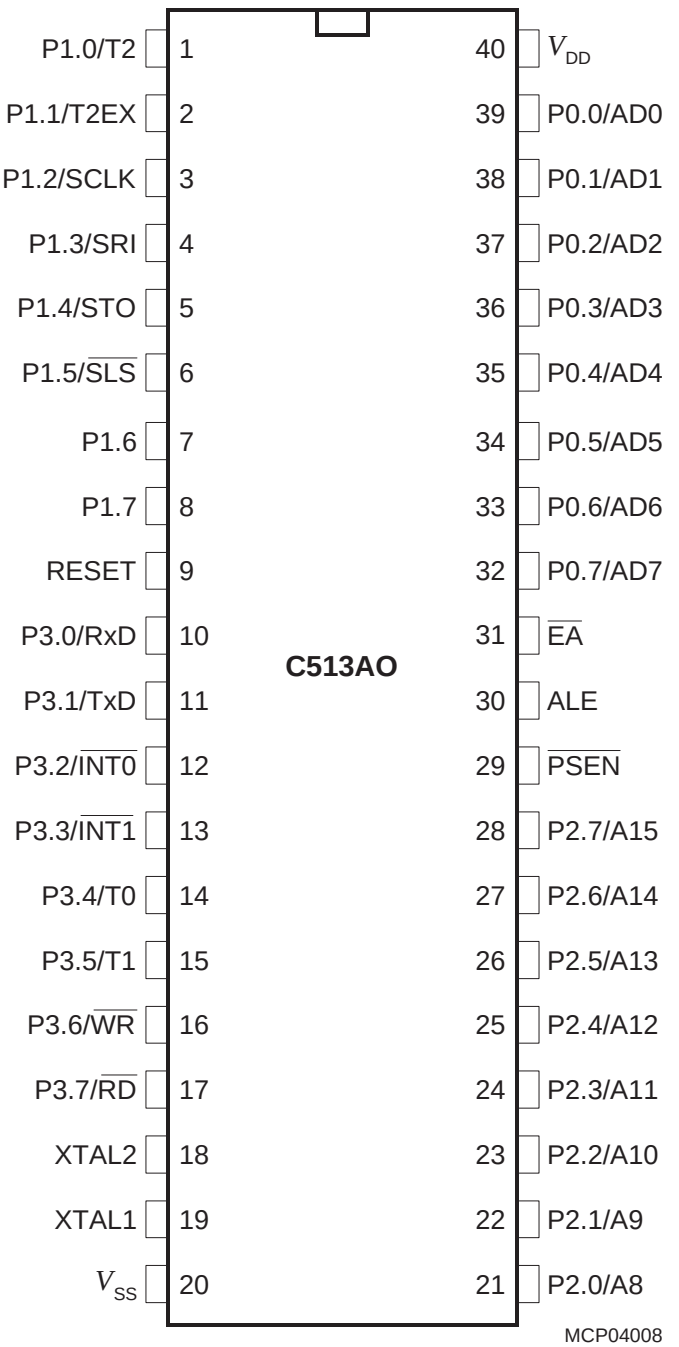


Figure 1-3
P-DIP-40 Package Pin Configuration (top view)

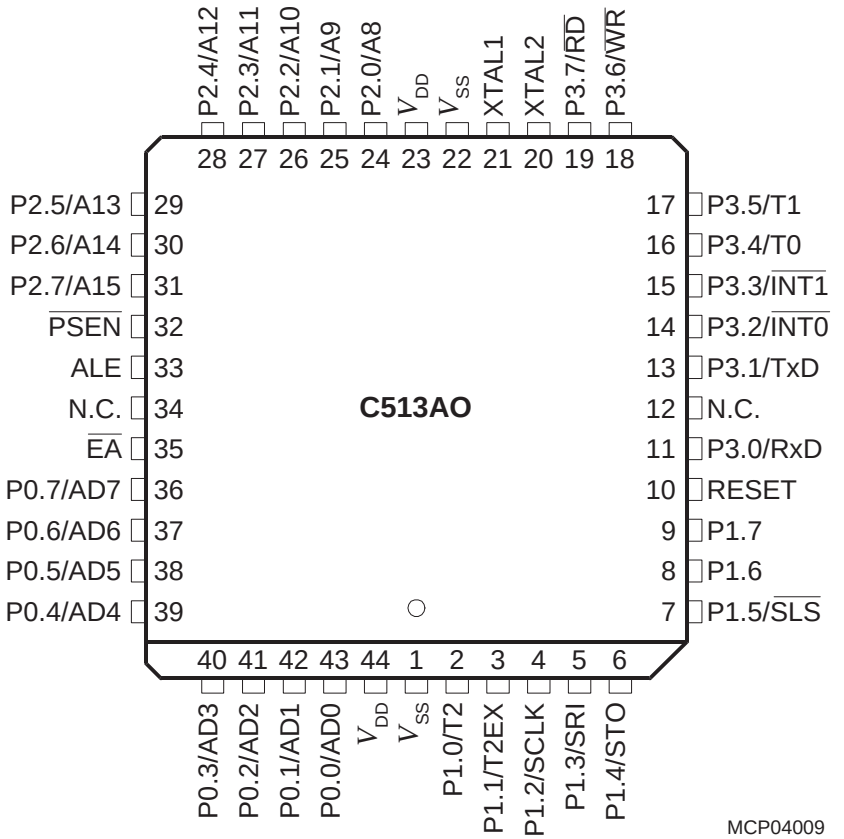


Figure 1-4
P-LCC-44 Package Pin Configuration (top view)

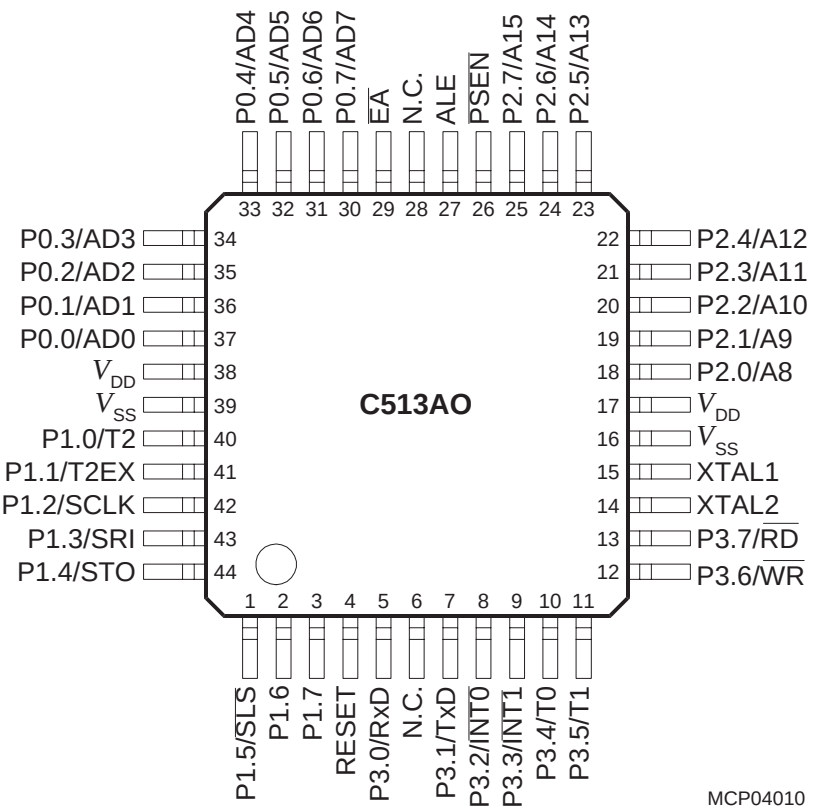


Figure 1-5
P-MQFP-44 Package Pin Configuration (top view)

1.2 Pin Definitions and Functions

This section describes all external signals to the C513AO and their functions.

Table 1-1
Pin Definitions and Functions

Symbol	Pin Number			I/O *)	Function	
	P-DIP-40	P-LCC-44	P-MQFP-44			
P1.7-P1.0	8-1	9-2	3-1, 44-40	I/O	**Port 1** Port 1 is an 8-bit quasi-bidirectional port with internal pull-up arrangement. Port 1 pins that have “1”s written to them are pulled high by the internal pull-up transistors 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 pull-up transistors. The output latch corresponding to a secondary function must be programmed to a “1” for that function to operate. For the outputs of the Synchronous Serial Channel (SSC), SCLK and 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 transistors will be switched off and the inputs float (high ohm inputs). The secondary functions are assigned to the pins of Port 1 as follows:	
	1	2	40		P1.0 / T2	Input to Counter 2
	2	3	41		P1.1 / T2EX	Capture/reload trigger of Timer 2 Up-Down count
	3	4	42		P1.2 / SCLK	SSC Master Clock Output SSC Slave Clock Input
	4	5	43		P1.3 / SRI	SSC Receive Input
	5	6	44		P1.4 / $\overline{\text{STO}}$	SSC Transmit Output
	6	7	1		P1.5 / $\overline{\text{SLS}}$	Slave Select Input

*) I = Input
O = Output

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

Symbol	Pin Number			I/O *)	Function	
	P-DIP-40	P-LCC-44	P-MQFP-44			
P3.0-P3.7	10-17	11, 13-19	5, 7-13	I/O	**Port 3** Port 3 is an 8-bit quasi-bidirectional port with internal pull-up arrangement. Port 3 pins that have "1"s written to them are pulled high by the internal pull-up transistors 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 pull-up transistors. The output latch corresponding to a secondary function must be programmed to a "1" for that function to operate (except for TxD and $\overline{WR}$). The secondary functions are assigned to the pins of Port 3 as follows:	
	10	11	5		P3.0 / RxD	Receiver data input (asynch.) or data input/output (synch.) of serial interface
	11	13	7		P3.1 / TxD	Transmitter data output (asynch.) or clock output (synch.) of serial interface
	12	14	8		P3.2 / $\overline{INT0}$	External Interrupt 0 input / Timer 0 gate control input
	13	15	9		P3.3 / $\overline{INT1}$	External Interrupt 1 input / Timer 1 gate control input
	14	16	10		P3.4 / T0	Timer 0 counter input
	15	17	11		P3.5 / T1	Timer 1 counter input
	16	18	12		P3.6 / $\overline{WR}$	$\overline{WR}$ control output; latches the data byte from Port 0 into the external data memory
	17	19	13		P3.7 / $\overline{RD}$	$\overline{RD}$ control output; enables the external data memory to Port 0

*) I = Input
O = Output

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

Symbol	Pin Number			I/O *)	Function
	P-DIP-40	P-LCC-44	P-MQFP-44		
RESET	9	10	4	I	**RESET** A high level on this pin for the duration of two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{DD}.
XTAL2	18	20	14	O	**XTAL2** Output of the inverting oscillator amplifier.
XTAL1	19	21	15	I	**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	21-28	24-31	18-25	I/O	**Port 2** Port 2 is a an 8-bit quasi-bidirectional I/O port with internal pull-up arrangement. Port 2 pins that have "1"s written to them are pulled high by the internal pull-up transistors, 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 transistors. 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 transistors when issuing "1"s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), Port 2 issues the contents of the P2 Special Function Register and uses only the internal pull-up transistors.

*) I = Input
O = Output

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

Symbol	Pin Number			I/O *)	Function
	P-DIP-40	P-LCC-44	P-MQFP-44		
$\overline{\text{PSEN}}$	29	32	26	O	**Program Store Enable** This is a control signal that enables output of the external program memory to the bus during external fetch operations. It is activated every six oscillator periods except during external data memory accesses. It remains high during internal program execution. This pin should not be driven during reset operation.
ALE	30	33	27	O	**Address Latch Enable** This output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access. When instructions are executed from internal program memory ($\overline{\text{EA}}$ = 1) the ALE generation can be disabled by bit EALE in SFR SYSCON. This pin should not be driven during reset operation.
$\overline{\text{EA}}$	31	35	29	I	**External Access Enable** When held at high level, instructions are fetched from the internal program memory when the PC is less than 4000_H. When held at low level, the C513AO fetches all instructions from external program memory. This pin should not be driven during reset operation. **Note**: For the C513AO-L this pin must be tied low.
P0.0-P0.7	32-39	43-36	37-30	I/O	**Port 0** Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have "1"s 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 pull-up transistors when issuing "1"s. External pull-up resistors are required during program verification.

*) I = Input
O = Output

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

Symbol	Pin Number			I/O *)	Function
	P-DIP-40	P-LCC-44	P-MQFP-44		
V_{SS}	20	22	16	–	**Ground** (0 V)
	–	1	39	–	**Ground** (0 V), **Optional** This pin may be left unconnected. It is, however, recommended to connect this pin to V_{SS} for optimized EMC performance
V_{DD}	40	44	38	–	**Power Supply** (+ 5 V)
	–	23	17	–	**Ground** (0 V), **Optional** This pin may be left unconnected. It is, however, recommended to connect this pin to V_{DD} for optimized EMC performance
N.C.	–	12, 34	6, 28	–	**No Connection**. These pins should not be connected.

*) I = Input
O = Output

2 Fundamental Structure

The C513AO family of microcontrollers is based on the C501 architecture. Therefore, they are also fully compatible with the industry-standard 8051 microcontrollers. The synchronous serial channel and the external memory (XRAM) are important features of the C513AO not found in the C501.

Figure 2-1 illustrates the major blocks of the C513AO device.

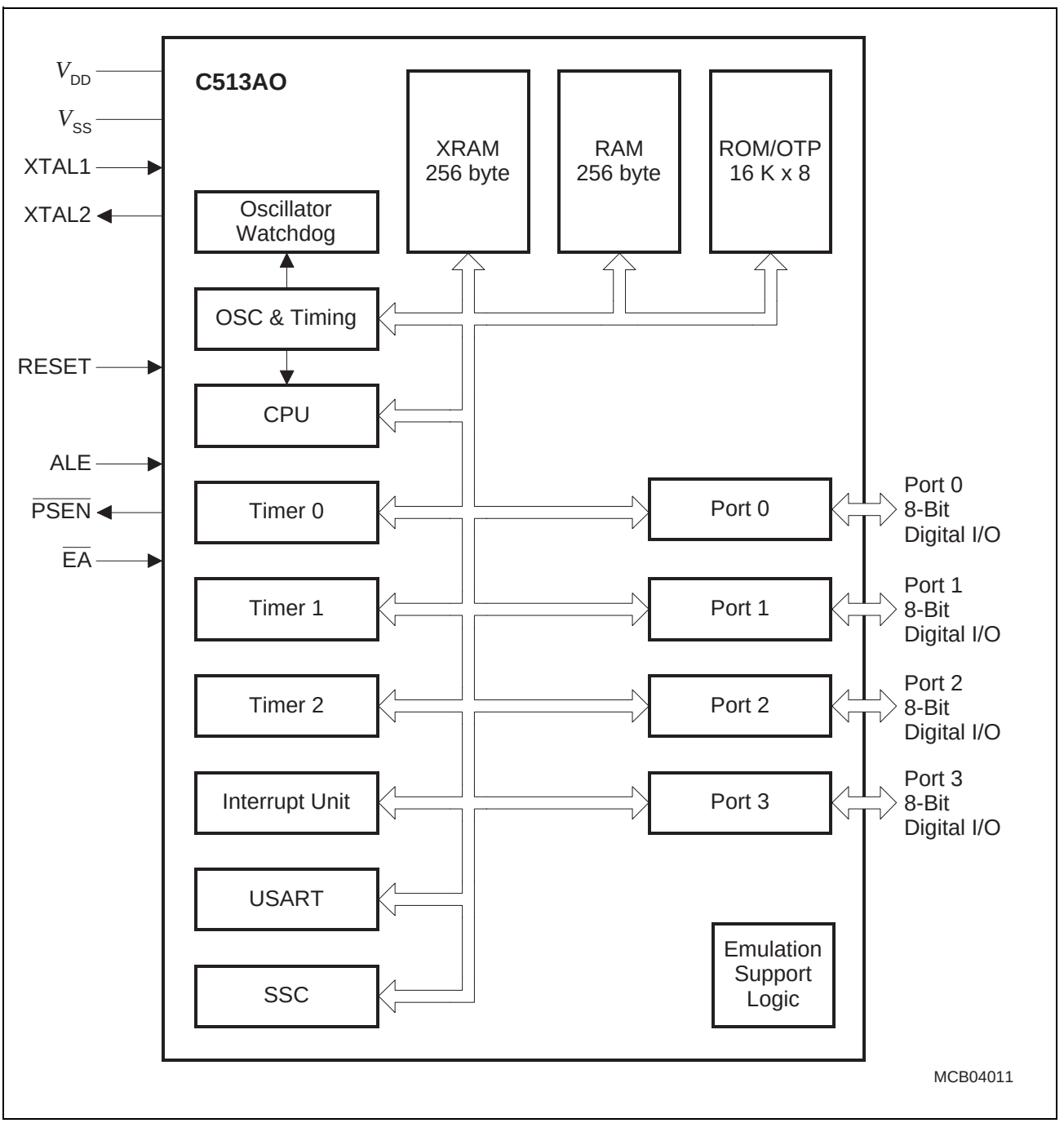


Figure 2-1
Block Diagram of the C513AO

2.1 Central Processing Unit (CPU)

The C513AO is efficient both as a controller and as an arithmetic processor. It has 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 a 16-MHz crystal, 58% of the instructions execute in 750 ns.

The CPU of the C513AO consists of the instruction decoder, the arithmetic section, and the program control section. Each program instruction is decoded by the instruction decoder. This section generates the internal signals which control the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers, and control the Arithmetic/Logic Unit (ALU) processing.

The arithmetic section of the processor performs extensive data manipulation and is comprised of the ALU and registers A, B, and PSW. 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. It performs the logic operations: AND, OR, Exclusive OR, complement and rotate (right, left or swap nibble (left four). Also included is a Boolean processor which performs the bit operations: set, clear, complement, 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 which reflect the current state of the CPU.

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

Bit No.	MSB							LSB	
	$D7_H$	$D6_H$	$D5_H$	$D4_H$	$D3_H$	$D2_H$	$D1_H$	$D0_H$	
$D0_H$	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

Bit	Function
CY	**Carry Flag** Used by arithmetic instruction.
AC	**Auxiliary Carry Flag** Used by instructions which execute BCD operations.
F0	**General Purpose Flag 0**
RS1 RS0	**Register bank Select control bits** These bits are used to select one of the four register banks.

RS1	RS0	Function
0	0	Bank 0 selected, data address 00_H-07_H
0	1	Bank 1 selected, data address 08_H-$0F_H$
1	0	Bank 2 selected, data address 10_H-17_H
1	1	Bank 3 selected, data address 18_H-$1F_H$

OV	**Overflow Flag** Used by arithmetic instruction.
F1	**General Purpose Flag 1**
P	**Parity Flag** Set or cleared by hardware after each instruction to indicate an odd or even number of 1 bits in the Accumulator. A “1” indicates odd parity while a “0” indicates 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. That is, 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 at location = 08_H above Register Bank 0. The SP can be read or written by software.

2.2 CPU Timing

A machine cycle consists of six states (twelve oscillator periods). Each state is divided into two phases. In Phase 1, the Phase 1 clock is active; in Phase 2, the Phase 2 clock is active. Thus, a machine cycle consists of twelve 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** show the fetch/execute timing related to the internal states and phases. Since these internal clock signals are not user-accessible, the XTAL2 oscillator signal and the Address Latch Enable (ALE) 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. Execution 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.
Figure 2-2 (a) and **(b)** show the timing for a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction, respectively.

Most C513AO instructions are executed in one cycle. Multiply (MUL) and divide (DIV) 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 execution of a MOVX instruction. 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 (c)** and **(d)** show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction, respectively.

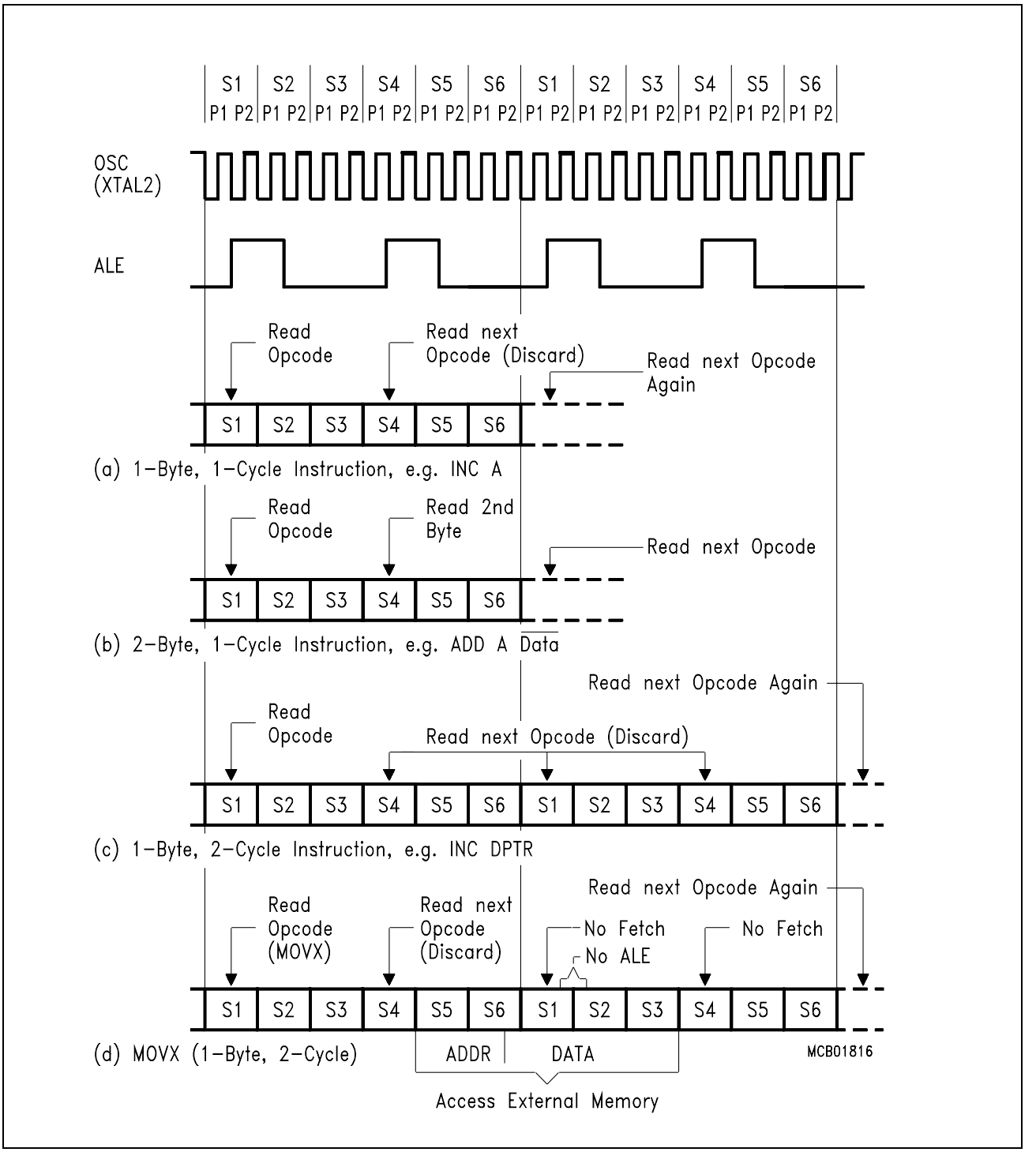


Figure 2-2
Fetch Execute Sequence

3 Memory Organization

The C513AO CPU manipulates operands in the following four address spaces:

- Up to 64 Kbytes of program memory (up to 16 KB on-chip program memory for the C513AO-2E/2R)
- Up to 64 Kbytes of external data memory
- 256 bytes of internal data memory
- 256 bytes of internal XRAM data memory
- One 128-byte special function register area

Figure 3-1 illustrates the memory address spaces of the C513AO.

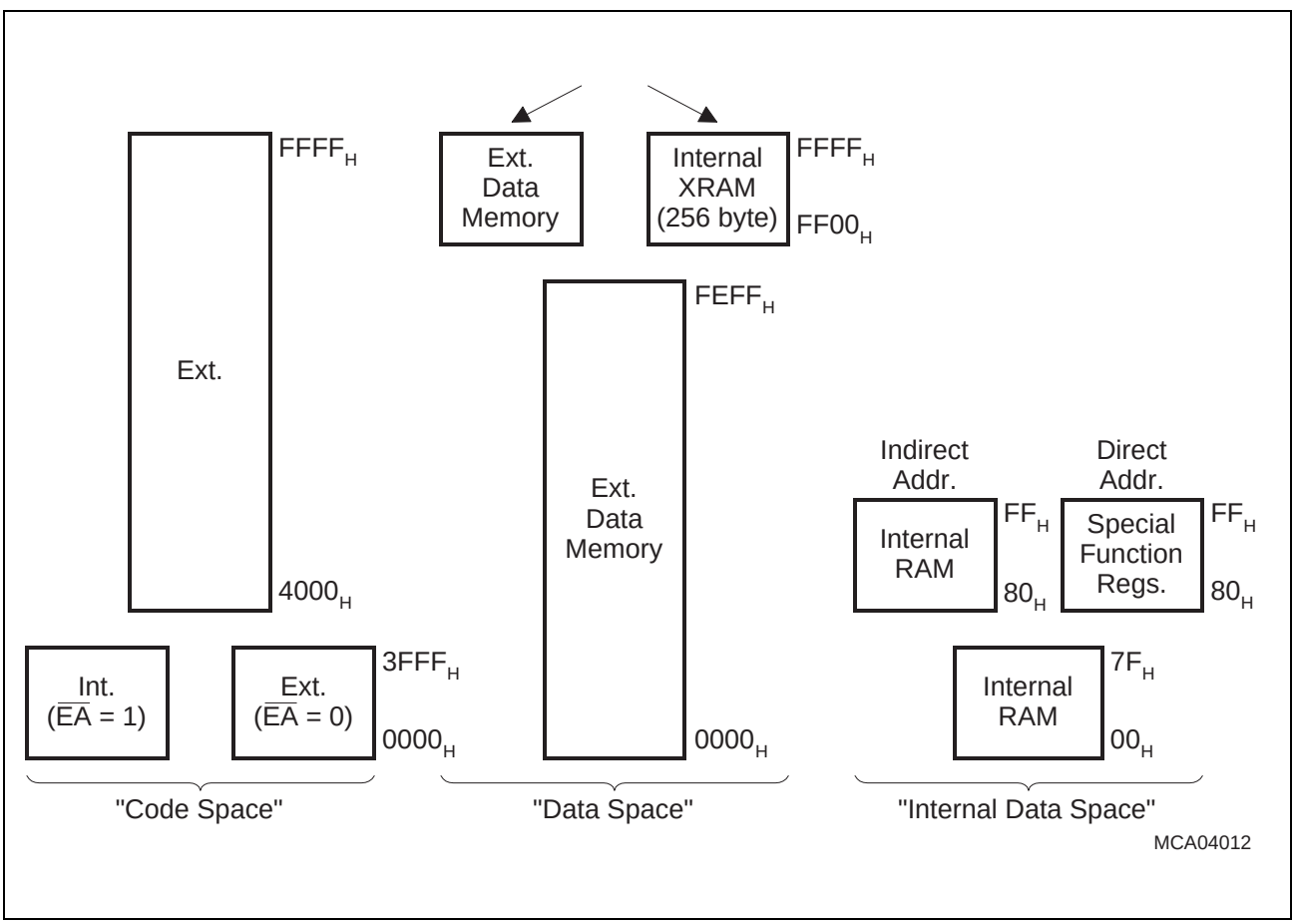


Figure 3-1
C513AO Memory Map

3.1 Program Memory, “Code Space”

The C513AO-2E/2R device has 16 Kbytes of program memory and can be externally expanded up to 64 Kbytes. If the $\overline{EA}$ pin is held high, the C513AO-2E/2R executes program code from the on-chip program memory unless the program counter address exceeds $3FFF_H$. Address locations 4000_H through $FFFF_H$ are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the C513AO (all versions) fetches all instructions from the external program memory. For C513AO-2R with ROM protection, the $\overline{EA}$ pin is sampled and latched on reset. In this case, the instruction fetches for PC < 4000_H are always made from the on-chip program memory. This is also true for C513AO-2E with programmed protection Level 1 or Level 2. If the on-chip program memory has been programmed with protection Level 3, the C513AO-2E/2R always starts program execution from the on-chip program memory unless an external code fetch is initiated by a program counter value exceeding $3FFF_H$.

3.2 Data Memory, “Data Space”

The data memory address space consists of both an internal and an external memory space. The internal data memory is divided into three physically separate and distinct blocks: the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128-byte Special Function Register (SFR) area. While the upper 128 bytes of data memory and the SFR area share the same address locations, they are accessed through different addressing modes. The lower 128 bytes of data memory can be accessed through direct or register-indirect addressing. The upper 128 bytes of RAM can be accessed through register indirect addressing. The special function registers are accessible through direct addressing. Four 8-register banks, each consisting of eight 8-bit general-purpose registers, occupy locations 0 through $1F_H$ in the lower RAM area. The next 16 bytes, locations 20_H through $2F_H$, contain 128 directly-addressable bit locations. The stack can be located anywhere in the internal RAM area and the stack depth can be expanded up to 256 bytes.

The external data memory can be expanded up to 64 Kbytes and can be accessed by instructions which use a 16-bit or an 8-bit address. The internal 256 Kbytes of XRAM are located in the external memory address area at addresses $FF00_H$ to $FFFF_H$.

3.3 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks of 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) select the active register bank. These bits are RS0 (PSW.3) and RS1 (PSW.4), (see description of the PSW in **Chapter 2**). This allows fast context switching, which is useful when entering subroutines or interrupt service routines.

The eight General Purpose Registers of the selected register bank may be accessed by register addressing. With register addressing, the instruction op-code indicates which register is to be used. For indirect addressing, R0 and R1 are used as pointer or index registers to address internal or external memory (such as: 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.

3.4 XRAM Operation

The XRAM in the C513AO is a memory area that is logically located at the upper end of the external data 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 C513AO maps 256 bytes of the external data space into the on-chip XRAM. This could prevent access to the external memory extension and might induce problems when porting software, especially when using the 8-bit addressing modes. Therefore, it is possible to enable and disable the on-chip XRAM using the bit XMAP in SFR SYSCON. When the XRAM is disabled (default after reset), all external data memory accesses will go to the external data memory area.

3.4.1 Reset Operation of the XRAM

The content of the XRAM is not affected by a reset. After power-up, the content is undefined; whereas, it remains unchanged during and after a reset if the power supply is not turned off. However, if a reset occurs during a write operation to XRAM, the effect on the content of an XRAM memory location depends on the cycle in which the active reset signal is detected (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, access to the XRAM is disabled (bit XMAP of SYSCON = 0).

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

- The XRAM can be accessed by two read/write instructions, which use the 16-bit DPTR for indirect addressing. These instructions are:
- MOVX A, @DPTR (Read)
- MOVX @DPTR, A (Write)

Using these instructions with the XRAM disabled implies that Port 0 is used as the address low/data bus, Port 2 is used for high address output, and two lines of Port 3 (P3.6/$\overline{WR}$/$\overline{INT2}$, P3.7/$\overline{RD}$) are used for control to access up to 64 KB of external memory. If the XRAM is enabled and if the effective address stored in DPTR is in the range of 0000_H to $FEFF_H$, these instructions will access external memory.

If XRAM is enabled and if the address is within the range $FF00_H$ to $FFFF_H$, the physically internal XRAM of the C513AO will be accessed. External memory, which is located in this address range, cannot be accessed in this case because no external bus cycles will be generated. Therefore, Ports 0, 2, and 3 can be used as general purpose I/O if only the XRAM memory space is addressed by the user program.

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

The C513AO architecture provides instructions for accesses to external data memory and XRAM which use an 8-bit address (indirect addressing with Registers R0 or R1). These instructions are:

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

Using these instructions with the XRAM disabled implies that Port 0 is used as the address/data bus, Port 2 is used for high address output, and two lines of Port 3 (P3.6/$\overline{WR}$, P3.7/$\overline{RD}$) are used for control. Normally, these instructions are used to access 256-byte pages of external memory.

If the XRAM is enabled, these instructions will only access the internal XRAM. External memory cannot be accessed in this case because no external bus cycle will be generated. Therefore Ports 0, 2, and 3 can be used as standard I/O, if only the internal XRAM is used.

3.5 Special Function Registers

The registers reside in the special function register area, with the exception of the Program Counter and the four General Purpose Register banks. The special function register area consists of two portions: the *standard* special function register area and the *mapped* special function register area. Four special function registers of the C513AO (PCON1, VR0, VR1 & VR2) are located in the mapped special function register area. For accessing the mapped special function register area, bit RMAP in special function register SYSCON must be set. All other special function registers of the C513AO are located in the standard special function register area.

Special Function Register SYSCON (Address $B1_H$) **Reset Value: $XX10XXX0_B$**

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

Bit	Function
EALE	**Enable ALE Output** EALE = 0: ALE generation is disabled during internal code memory accesses ($\overline{EA}$ = 1); ALE is generated during MOVX instructions. EALE = 1: ALE generation is enabled. If $\overline{EA}$ = 0, the ALE generation is always enabled and the bit EALE has no effect on ALE generation.
RMAP	**Special function Register MAP bit** RMAP = 0: The access to the non-mapped (standard) special function register area is enabled. RMAP = 1: The access to the mapped special function register area is enabled.
XMAP	**Global XRAM MAP access enable/disable control** XMAP = 0: On-chip XRAM disabled (default after reset). XMAP = 1: On-chip XRAM enabled.
–	Reserved for future use. Read by CPU; returns undefined values.

If bit RMAP is set, mapped special function registers can be accessed. This bit is not cleared by hardware automatically.

The forty Special Function Registers (SFRs) in the standard and mapped SFR area include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. The SFRs of the C513AO 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 C513AO. **Table 3-2** illustrates the contents of the SFRs in numeric order of their addresses.

Table 3-1
Special Function Registers - 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 Register	**$D0_H$** [1]	00_H
	SP	Stack Pointer	81_H	07_H
	SYSCON	System Control Register	$B1_H$	$XX10XXX0_B$ [3]
	VR0 [4] [5]	Version Register 0	FC_H	$C5_H$
	VR1 [4] [5]	Version Register 1	FD_H	[6]
	VR2 [4] [5]	Version Register 2	FE_H	[7]
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
Serial Channel (USART)	PCON [2]	Power Control Register	87_H	$00XX0000_B$ [3]
	SBUF	Serial Channel Buffer Register	99_H	XX_H [3]
	SCON	Serial Channel Control Register	**98_H** [1]	00_H
SSC Interface	SSCCON	SSC Control Register	**$E8_H$** [1]	07_H
	STB	SSC Transmit Register	$E9_H$	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 [8]	SSC Mode Test Register	EB_H	00_H
Timer 0/ Timer 1	TCON	Timer 0/1 Control Register	**88_H** [1]	00_H
	TH0	Timer 0, High Byte	$8C_H$	00_H
	TH1	Timer 1, High Byte	$8D_H$	00_H
	TL0	Timer 0, Low Byte	$8A_H$	00_H
	TL1	Timer 1, Low Byte	$8B_H$	00_H
	TMOD	Timer Mode Register	89_H	00_H

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 undefined and the location is reserved
4) This SFR is a mapped SFR. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
5) This SFR is read-only.
6) C513AO-L/2R: 13H
C513AO-2E: 83H
7) This SFR varies with the step of the microcontroller: for example, 01_H for the first step
8) This register is only used for test purposes and must not be written during normal operation. Unpredictable results may occur upon a write operation.

Table 3-1
Special Function Registers - Functional Blocks (cont'd)

Block	Symbol	Name	Address	Contents after Reset
Timer 2	T2CON	Timer 2 Control Register	**$C8_H$** [1]	00_H
	T2MOD	Timer 2 Mode Register	$C9_H$	$XXXXXXX0_B$ [3]
	RC2H	Timer 2 Reload/Capture Register, High Byte	CB_H	00_H
	RC2L	Timer 2 Reload/Capture Register, Low Byte	CA_H	00_H
	TH2	Timer 2 High Byte	CD_H	00_H
	TL2	Timer 2 Low Byte	CC_H	00_H
Watchdog	WDCON	Watchdog Timer Control Register	**$C0_H$** [1]	$XXXX0000_B$ [3]
	WDTREL	Watchdog Timer Reload Register	86_H	00_H
Power Save Mode	PCON [2]	Power Control Register	87_H	$00XX0000_B$ [3]
	PCON1 [4]	Power Control Register 1	88_H	$0XXXXXX_B$ [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 undefined and the location is reserved
4) This SFR is a mapped SFR. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
5) This SFR is read-only.
6) C513AO-L/2R: 13H
C513AO-2E: 83H
7) This SFR varies with the step of the microcontroller: for example, 01_H for the first step
8) This register is only used for test purposes and must not be written during normal operation. Unpredictable results may occur upon a write operation.

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 [2)]	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
86_H	WDTREL	00_H	WDT PSEL	.6	.5	.4	.3	.2	.1	.0
87_H	PCON	$0XX0\text{-}0000_B$	SMOD	–	–	SD	GF1	GF0	PDE	IDLE
88_H [2)]	TCON	00_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
88_H [3)]	PCON1	$0XXX\text{-}XXXX_B$	EWPD	–	–	–	–	–	–	–
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 [2)]	P1	FF_H	–	–	$\overline{SLS}$	STO	SRI	SCLK	T2EX	T2
98_H [2)]	SCON	00_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99_H	SBUF	XX_H	.7	.6	.5	.4	.3	.2	.1	.0
$A0_H$ [2)]	P2	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
$A8_H$ [2)]	IE	00_H	EA	ESSC	ET2	ES	ET1	EX1	ET0	EX0
$B0_H$ [2)]	P3	FF_H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD
$B1_H$	SYSCON	$XX10\text{-}XXX0_B$	–	–	EALE	RMAP	–	–	–	XMAP
$B8_H$ [2)]	IP	$X000\text{-}0000_B$	–	PSSC	PT2	PS	PT1	PX1	PT0	PX0
$C0_H$ [2)]	WDCON	$XXXX\text{-}0000_B$	–	–	–	–	OWDS	WDTS	WDT	SWDT

1) “X” means that the value is undefined and the location is reserved.
2) Bit-addressable special function registers.
3) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
4) These are read-only registers.
5) The content of this SFR varies with the actual step of the C513A0: for example, 01_H for the first step).
6) This register is only used for test purposes and must not be written during normal operation. Unpredictable results may occur upon a write operation.

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
$C8_H$ [2]	T2CON	00_H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
$C9_H$	T2MOD	XXXX-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
$D0_H$ [2]	PSW	00_H	CY	AC	F0	RS1	RS0	OV	F1	P
$E0_H$ [2]	ACC	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E8_H$ [2]	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 [6]	SSCMOD	00_H	LOOPB	TRIO	0	0	0	0	0	LSBSM
$F0_H$ [2]	B	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F8_H$ [2]	SCF	XXXX-XX00$_B$	–	–	–	–	–	–	WCOL	TC
$F9_H$	SCIEN	XXXX-XX00$_B$	–	–	–	–	–	–	WCEN	TCEN
FC_H [3] [4]	VR0	$C5_H$	.7	.6	.5	.4	.3	.2	.1	.0
FD_H [3] [4]	VR1	– [7]	.7	.6	.5	.4	.3	.2	.1	.0
FE_H [3] [4]	VR2	– [5]	.7	.6	.5	.4	.3	.2	.1	.0

1) X means that the value is undefined and the location is reserved
2) Bit-addressable special function registers
3) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
4) These SFRs are read-only registers.
5) The content of this SFR varies with the actual step of the C513A0: for example, 01_H for the first step)
6) This register is only used for test purposes and must not be written during normal operation. Unpredictable results may occur upon a write operation.
7) C513AO-L/2R: 13_H
C513AO-2E: 83_H

4 External Bus Interface

The C513AO allows for 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 C513AO is used in systems with no external memory, the generation of the ALE signal can be suppressed. By resetting bit EALE in the SFR SYSCON register, the ALE signal will not be generated externally. This feature reduces RFI emissions of the system.

4.1 Accessing External Memory

It is possible to differentiate between accesses to external program memory and external data memory or other peripheral components. This differentiation 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 Port 0 and Port 2 functions relevant to external memory accesses are described.

Fetches from external program memory always use 16-bit addresses. Accesses to external data memory can use either 16-bit addresses (MOVX @DPTR) or 8-bit addresses (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, 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 need to contain “1”s, and the contents of the Port 2 SFR are not modified. If the XRAM is enabled, at 16-bit address accesses with address values 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 pull-ups will be used for the entire read/write cycle and not only for two oscillator periods. Regardless of the address, if the XRAM is enabled, no external bus cycle will be seen.

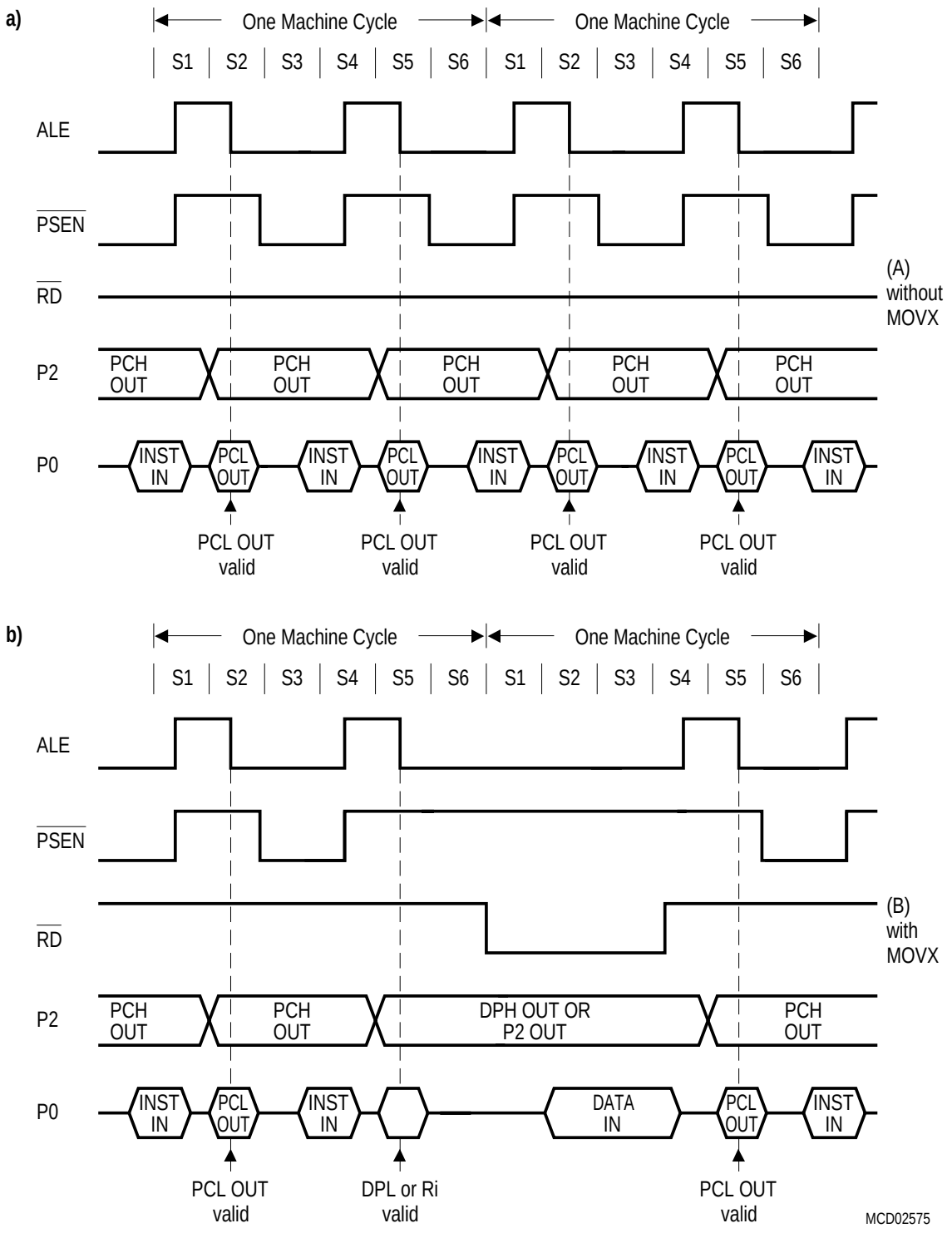


Figure 4-1
External Program Memory Execution

4.1.2 Timing

Figure 4-1 (a) and **(b)** illustrate timing of the external bus interface with particular emphasis on the relationship between the control signals ALE, PSEN, RD, WR and information on Port 0 and Port 2.

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 WR is deactivated. In a read cycle, the incoming byte is accepted at Port 0 before the read strobe is deactivated.

Program memory: Signal PSEN functions as a read strobe.

4.1.3 External Program Memory Access

The external program memory is accessed under either of the following conditions:

- Whenever signal EA of an unprotected device or latched signal EA of a protected device is active
- Whenever the Program Counter (PC) contains a number that is larger than $3FFF_H$.

*Note: For information on unprotected/protected devices, see **Section 4.5** ROM/OTP Protection.*

For the ROMless version of the device, C513AO-L, to have EA wired low allows program bytes to be fetched from external memory.

When the CPU is executing out of external program memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general-purpose I/O. The contents of the Port 2 SFR, however, are not affected. During external program memory fetches, Port 2 lines output the high byte of the Program Counter. 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).

Since the C513AO-L has no internal program memory, accesses to program memory are always external, and Port 2 is dedicated at all times to output of the high-order address byte. This means that Port 0 and Port 2 of the C513AO-L can never be used as general-purpose I/O. This also applies to the C513AO-2R or C513AO-2E if they are operated with only an external program memory.

4.2 PSEN, Program Store Enable

The read strobe for external fetches is PSEN. PSEN is not activated for internal fetches. When the CPU is accessing external program memory, PSEN is activated twice every cycle (except during a MOVX instruction) whether or not the byte fetched is actually needed for the current instruction. When PSEN is activated, its timing is not the same as for RD. A complete RD cycle, including activation and deactivation of ALE and RD, takes twelve oscillator periods. A complete PSEN cycle, including activation and deactivation of ALE and PSEN, takes six oscillator periods. The execution sequence for these two types of read cycles is shown in **Figure 4-1 (a)** and **(b)**.

4.3 Overlapping External Data and Program Memory Spaces

In some applications, it is desirable to execute a program from the same physical memory as that used for storing data. In the C513AO, the external program and data memory spaces can be combined by AND-ing PSEN and 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 PSEN cycle is faster than the RD cycle, the external memory needs to be fast enough to adapt to the PSEN cycle.

4.3.1 Address Latch Enable (ALE)

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from Port 0 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 4-1 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 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 only rarely.

The C513AO allows the ALE output signal to be switched off. If the internal program memory is used by setting $\overline{EA}$ = 1 and ALE is switched off by setting EALE = 0, ALE will only go active during external data memory accesses (MOVX instructions) and code memory accesses with an address greater than $3FFF_H$ (external code memory fetches). If $\overline{EA}$ = 0, the ALE generation is always enabled and the bit EALE has no effect.

After a hardware reset, the ALE generation is enabled.

Special Function Register SYSCON (Address $B1_H$) **Reset Value: $XX10XXX0_B$**

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

Bit	Function
–	Not implemented. Reserved for future use.
EALE	**Enable ALE output** EALE = 0: ALE generation is disabled; disables ALE signal generation during internal code memory accesses ($\overline{EA}$ = 1). With $\overline{EA}$ = 1, ALE is automatically generated at MOVX instructions and code memory accesses with an address greater $3FFF_H$. EALE = 1: ALE generation is enabled If $\overline{EA}$ = 0, the ALE generation is always enabled and the bit EALE has no effect on the ALE generation.

4.4 Enhanced Hooks Emulation Concept

The Enhanced Hooks Emulation Concept of the C500 microcontroller family enables innovative control of C500 MCU execution and provides extensive information on the internal operation of the controllers. Emulation of on-chip ROM based programs is possible, too. Each device in the C500 family has built-in logic to support the Enhanced Hooks Emulation Concept. Therefore, no costly bond-out chips are necessary for emulation. This also ensures that emulation and production chips are identical.

The Enhanced Hooks Technology™ 1), which requires embedded logic in the C500 allows the C500 together with an EH-IC to function in a manner similar to a bond-out chip. This simplifies design and reduces costs of an ICE-system. ICE-systems using an EH-IC and a compatible C500 are able to emulate all operating modes of the different versions of the C500. This includes emulation of ROM, ROM with code rollover and ROMless modes of operation. It is also able to operate in Single Step Mode and to read the SFRs after a break.

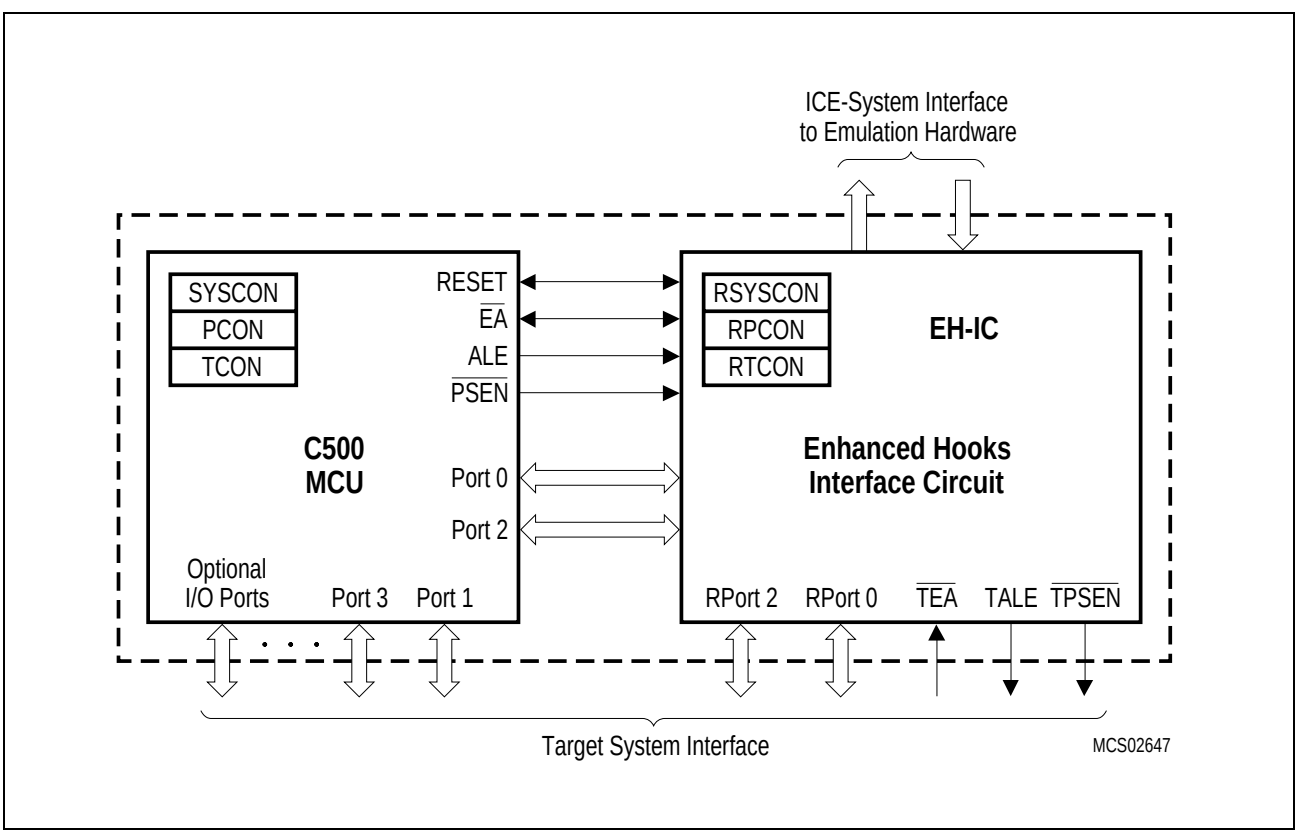


Figure 4-2
Basic C500 MCU Enhanced Hooks Concept Configuration

Port 0, Port 2 and some of the control lines of the C500 based MCU are used by the Enhanced Hooks Emulation Concept to control the operation of the device during emulation and to transfer information about the program execution and data transfer between the external emulation hardware (ICE-system) and the C500 MCU.

1)"Enhanced Hooks Technology" is a trademark and patent of MetaLink Corporation licensed to Infineon Technologies.

4.5 ROM/OTP Protection

The C513AO-2R ROM version allows protection of the content of the internal ROM against read out by unauthorized people. The type of ROM protection (protected or unprotected) is fixed with the ROM mask. Therefore, users of the C513AO-2R ROM version device must predefine whether ROM protection is to be selected or not.

The C513AO-2E OTP version also allows program memory protection in several levels (see Chapter 10.6). The program memory protection for the C513AO-2E can be activated after programming of the device.

The C513AO-2R devices, which operate from internal ROM, are always checked for correct ROM content during production test. Therefore, both unprotected and protected ROMs must provide a procedure to verify the ROM content. In ROM Verification Mode 1, which is used to verify unprotected ROMs, a ROM address is applied externally to the C513AO-2R and the ROM data byte is output at Port 0. ROM Verification Mode 2, which is used to verify ROM protected devices, operates differently: ROM addresses are generated internally and the expected data bytes must be applied externally to the device (by the manufacturer or by the customer) and are compared internally with the data bytes from the ROM. After 16 byte-verify operations, the state of the P3.5 pin shows whether the last 16 bytes have been verified correctly.

This mechanism provides very secure ROM protection. Only the owner of the ROM code and the manufacturer who knows the content of the ROM can verify it.

4.5.1 Unprotected ROM Mode

If the ROM is unprotected, ROM Verification Mode 1, shown in **Figure 4-3,** is used to read out the content of the ROM. (See also the AC Specifications in the Data Sheet; not valid for C513AO-2E).

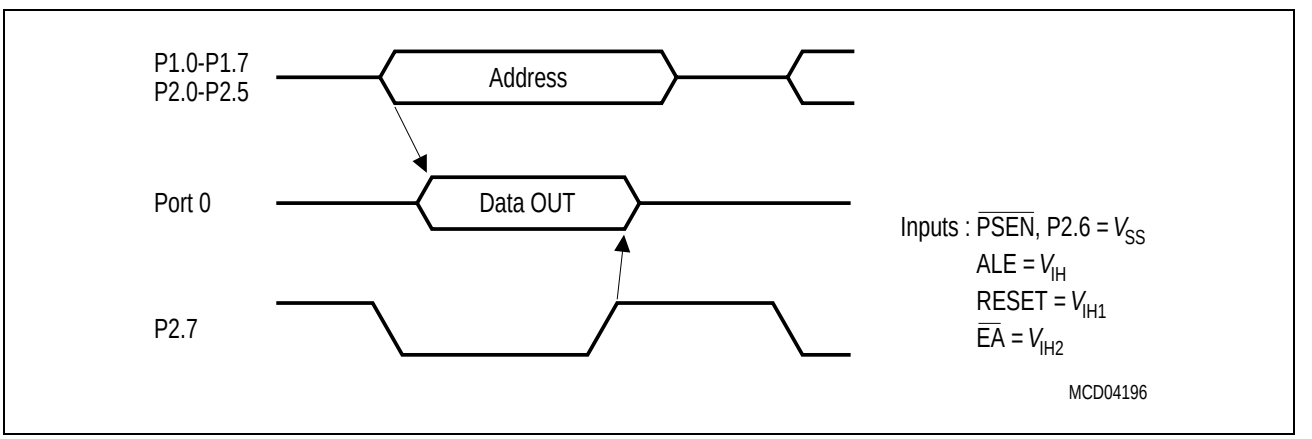


Figure 4-3
ROM Verification Mode 1

ROM Verification Mode 1 is selected if the inputs $\overline{\text{PSEN}}$, ALE, $\overline{\text{EA}}$, and RESET are put to the specified logic level. P2.6 and P2.7 must be held at low level. Whenever the 14-bit address of the internal ROM byte to be read is applied to Port 1 and Port 2, Port 0 outputs the content of the addressed internal program memory cell after a delay time. In ROM Verification Mode 1, the C513AO-2R must be provided with a system clock at the XTAL pins and pull-up resistors on the Port 0 lines.

4.5.2 Protected ROM/OTP Mode

For the C513AO-2R ROM protected by mask, and for the C513AO-2E OTP in Protection Level 1, ROM/OTP Verification Mode 2, shown in **Figure 4-4**, is used to verify the content of the ROM/OTP. The detailed timing characteristics of the ROM/OTP verification mode are shown in the data sheet.

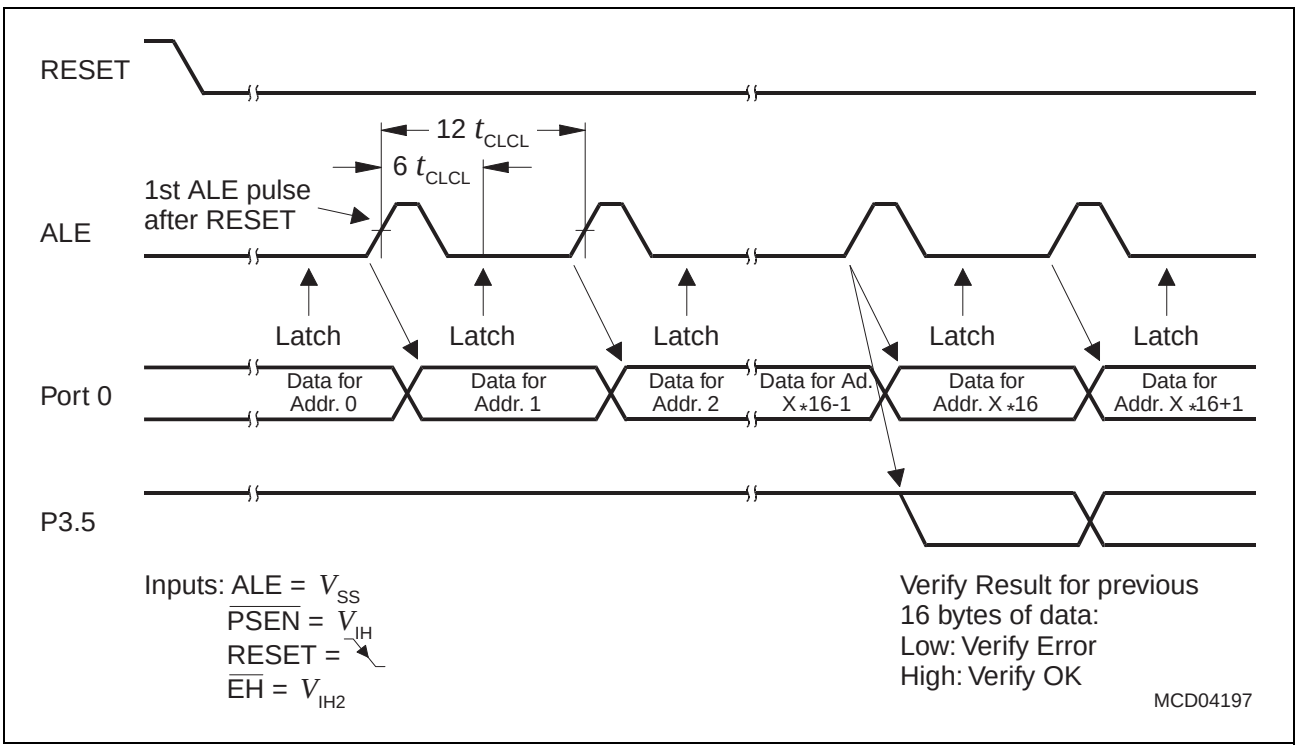


Figure 4-4
ROM Verification Mode 2

ROM/OTP Verification Mode 2 is selected if the inputs $\overline{PSEN}$, $\overline{EA}$, and ALE are set to the specified logic levels. With RESET going inactive, the ROM/OTP Verification Mode 2 sequence is started. The C513AO outputs an ALE signal with a period of 12 t_{CLCL} and expects data bytes at Port 0. The data bytes at Port 0 are assigned to the ROM addresses in the following way:

1. Data Byte = content of internal ROM/OTP address 0000_H
2. Data Byte = content of internal ROM/OTP address 0001_H
3. Data Byte = content of internal ROM/OTP address 0002_H
:
16. Data Byte = content of internal ROM/OTP address 000FH
:

The C513AO does not output any address information during ROM/OTP Verification Mode 2. The first data byte to be verified is always the byte which is assigned to the internal ROM address 0000_H and must be put onto the data bus with the falling edge of RESET. With each following ALE pulse, the ROM/OTP address pointer is internally incremented and the expected data byte for the next ROM address must be delivered externally.

The data at Port 0 is latched between two ALE pulses, (at 6 t_{CLCL} after ALE rising edge) and is compared internally with the ROM/OTP content of the actual address. If a verify error is detected, the error condition is stored internally. After each 16th data byte the cumulated verify result (pass

or fail) of the last 16 verify operations is output at P3.5. P3.5 is always set or cleared after each 16 byte block of the verify sequence. In ROM/OTP Verification Mode 2, the C513AO must be provided with a system clock at the XTAL pins.

Figure 4-5 shows an application example of external circuitry used to verify a protected ROM/OTP inside the C513AO in ROM/OTP Verification Mode 2. With RESET going inactive, the C513AO starts the ROM/OTP verify sequence. Its ALE is clocking a 14-bit address counter. This counter generates the addresses for an external EPROM which is programmed with the content of the internal (protected) ROM/OTP. The verify detect logic typically displays the state of the verify error output P3.5. P3.5 can be latched with the falling edge of ALE.

When the last byte of the internal ROM/OTP has been handled, the C513AO starts generating a PSEN signal. This PSEN signal or the Carry (CY) signal of the address counter indicates, to the verify detect logic, the end of the internal ROM/OTP verification.

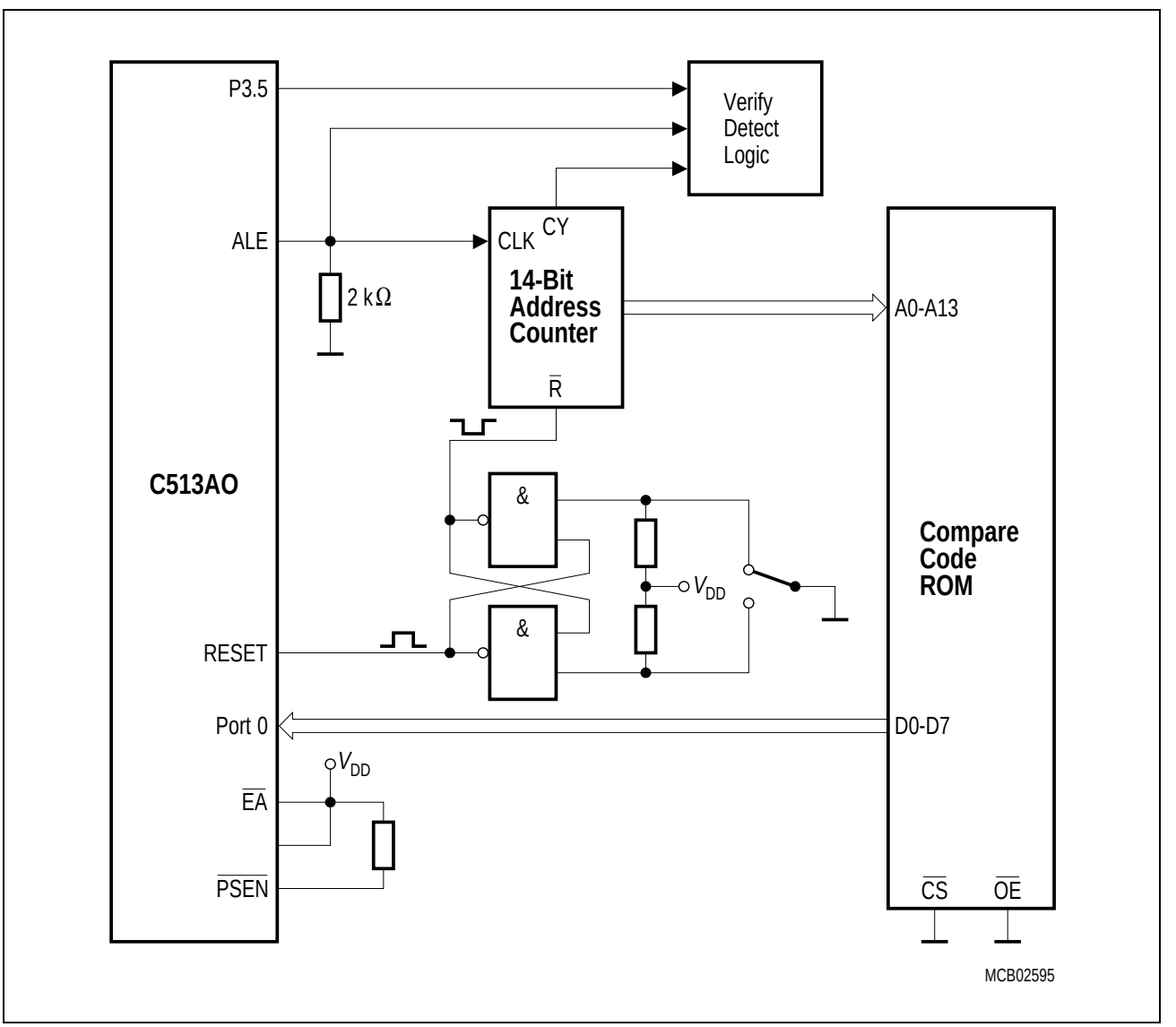


Figure 4-5
ROM/OTP Verification Mode 2 - External Circuitry Example

5 Reset and System Clock Operation

5.1 Hardware Reset Operation

The hardware reset function incorporated in the C513AO allows easy automatic start-up of a minimum set of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation to restart the device. This is particularly useful when the Power-down Mode is to be terminated.

In addition to the hardware reset, which is applied externally to the C513AO, there are two internal reset sources: the Watchdog Timer and the Oscillator Watchdog. This chapter deals with the external hardware reset only.

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

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

The time required for a reset operation is the oscillator start-up time plus 2 machine cycles. Under normal conditions, this must be at least 10 - 20 ms for a crystal oscillator. This requirement is typically met using a capacitor of 4.7 to 10 µF. The same considerations apply if the reset signal is generated externally (see **Figure 5-1 (b)**). In each case, it must be assured that the oscillator has started up properly and that at least two machine cycles have passed before the reset signal goes inactive.

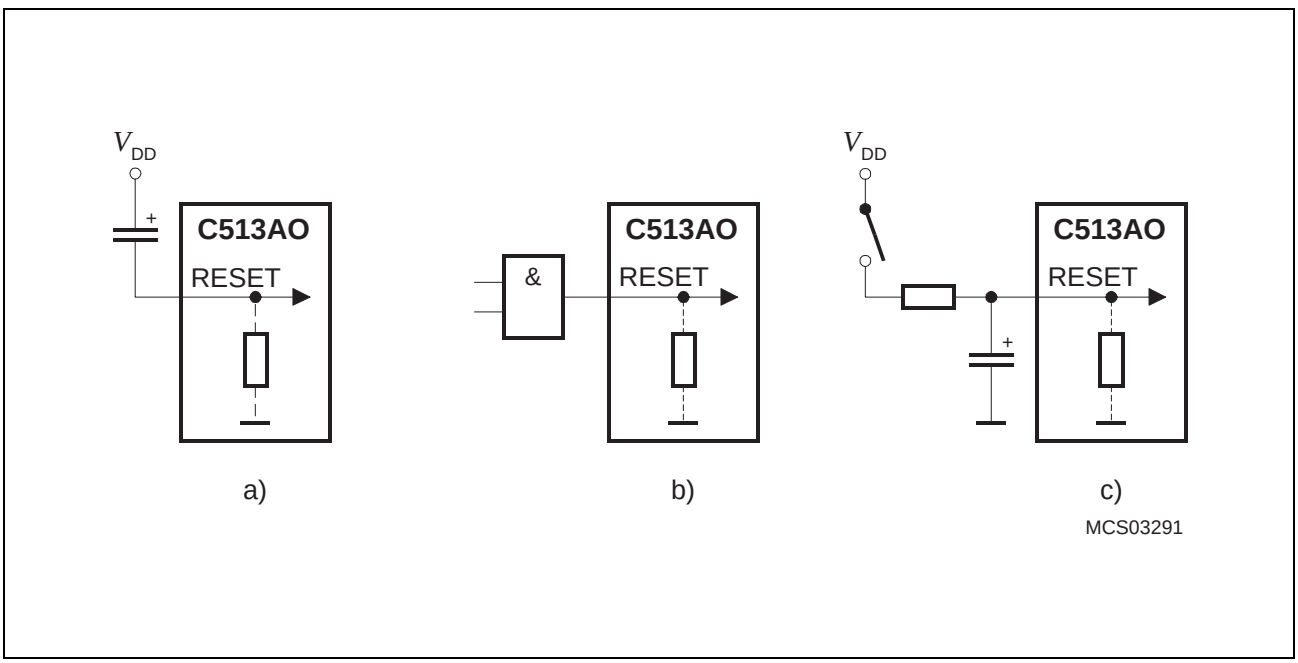


Figure 5-1
Reset Circuitries

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 to 3) output “1”s.

The contents of the internal RAM and XRAM of the C513AO are not affected by a reset. After power-up, the contents are undefined, while it remains unchanged during a reset if the power supply is not turned off.

5.2 Fast Internal Reset after Power-On

The C513AO uses the Oscillator Watchdog unit for a fast internal reset procedure after power-on. **Figure 5-2** shows the power-on sequence under control of the Oscillator Watchdog.

Normally, devices in the 8051 microcontroller family do not enter their default reset state before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed to bring the device into the correct reset state. Especially if a crystal is used, the start up time of the oscillator is relatively long (typ. 10 ms). During this time period, the pins have an undefined state which could have severe effects, especially to actuators connected to port pins.

In the C513AO, the Oscillator Watchdog unit avoids this problem. With this device, after power-on, the Oscillator Watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 ms). Following this, the watchdog circuitry detects a failure condition for the on-chip oscillator because this has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is detected, the watchdog uses the RC oscillator output as clock source for the chip rather than the on-chip oscillator's output. This allows correct resetting of the part and brings all ports to the defined state (see **Figure 5-2**).

Under worst case conditions (fast V_{DD} rise time, such as 1µs, measured from V_{DD} = 4.25 V up to stable port condition), the delay between power-on and the correct port reset state is:

- Typical: 18 µs
- Maximum: 34 µs

The RC oscillator will already run at a V_{DD} below 4.25V (lower specification limit). Therefore, at slower V_{DD} rise times, the delay time will be less than the two values given above.

After the on-chip oscillator has finally started, the Oscillator Watchdog detects the correct function; then the watchdog still holds the reset active for a time period of max. 768 cycles of the RC oscillator clock to allow the oscillation of the on-chip oscillator to stabilize (**Figure 5-2, II**). Subsequently, the clock is supplied by the on-chip oscillator and the oscillator watchdog's reset request is released (**Figure 5-2, III**). However, an externally applied reset still remains active (**Figure 5-2, IV**) and the device does not start program execution (**Figure 5-2, V**) before the external reset is also released.

Although the Oscillator Watchdog provides a fast internal reset, it is additionally necessary to apply the external reset signal when powering up for the following reasons:

- Termination of software Power-Down Mode
- Reset of the status flag OWDS which is set by the Oscillator Watchdog during the power up sequence.

Using a crystal or ceramic resonator for clock generation, the external reset signal must be held active at least until the on-chip oscillator has started and the internal watchdog reset phase is completed (after Phase III in **Figure 5-2**). When an external clock generator is used, Phase II is very short. Therefore, an external reset time of typically 1 ms is sufficient in most applications.

For reset time generation at power-on, an external capacitor can be applied to the RESET pin.

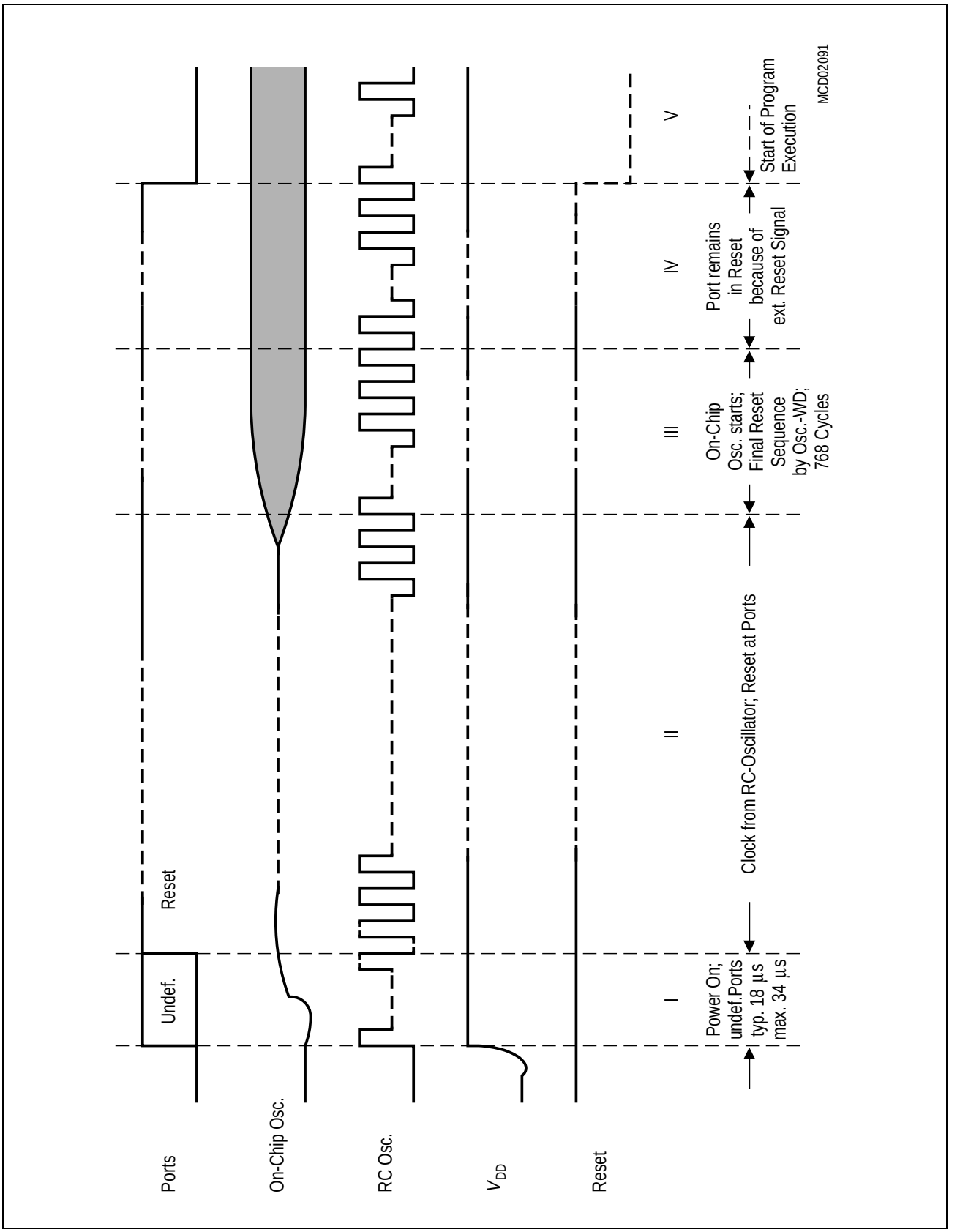


Figure 5-2
Power-On Reset Timing of the C513AO

5.3 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 occurs in State 5 Phase 2 (S5P2). Thus, the external reset signal is synchronized to the internal CPU timing. When the RESET signal is detected to be active at S5P2, the internal reset procedure is started. It needs two complete machine cycles to put the complete device into its correct reset state; that is, all special function registers contain their default values, the port latches contain “1”s, etc. Note that this reset procedure is also performed if there is no clock available at the device. (This is done by the Oscillator Watchdog, which provides an auxiliary clock for performing a perfect reset without clock at the XTAL1 and XTAL2 pins). The RESET signal must be active for at least two machine cycles; after this interval, the C513AO remains in its reset state as long as the signal is active. When the signal goes inactive, this transition is recognized in the subsequent S5P2 of the machine cycle. Then, the processor starts its address output (when configured for external ROM) in the subsequent State 5 Phase 1. One phase later (State 5, Phase 2), the first falling edge at pin ALE occurs.

Figure 5-3 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 interval of at least one machine cycle but less than two machine cycles.

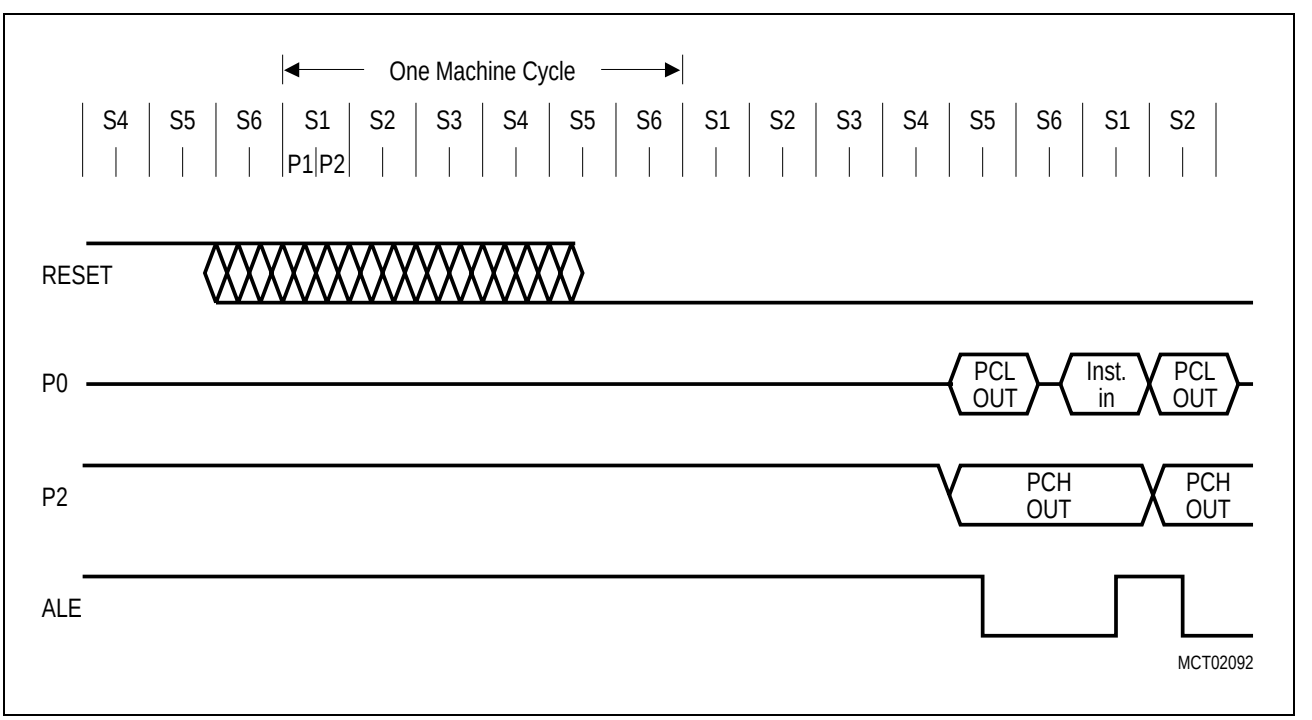


Figure 5-3
CPU Timing after Reset

5.4 Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components as a pierce oscillator. The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clock signals to the chip. These signals define the internal phases, states, and machine cycles.

Figure 5-4 shows the recommended oscillator circuit.

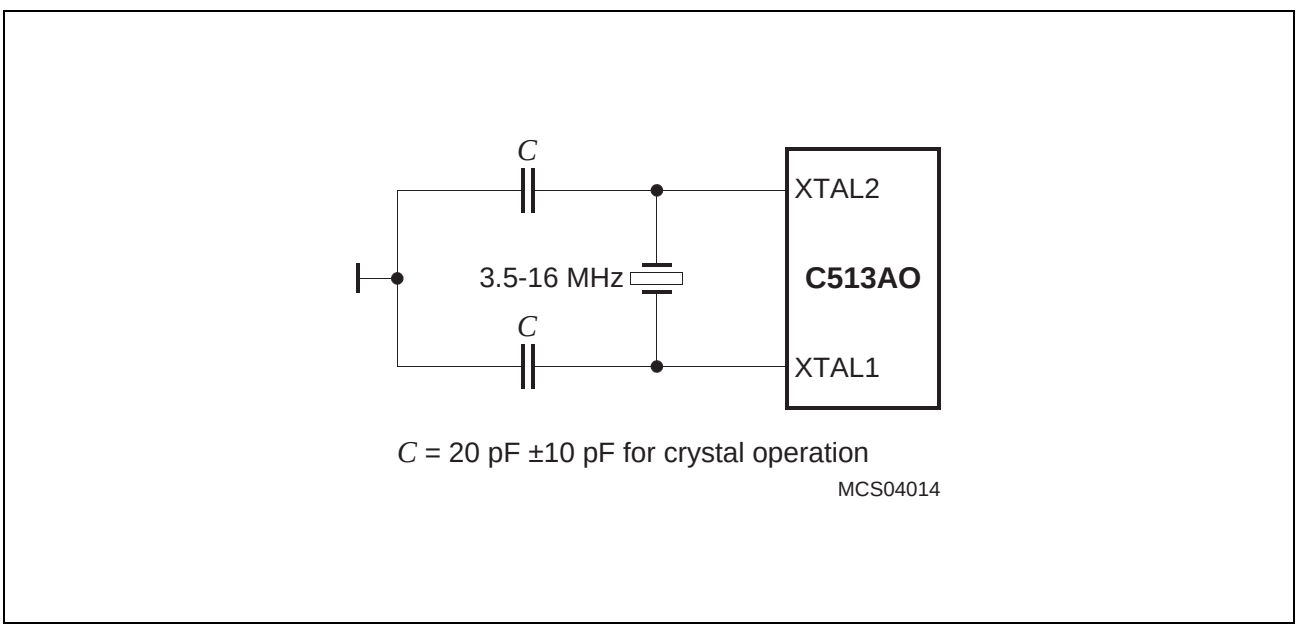


Figure 5-4
Recommended Oscillator Circuit

In this application, the on-chip oscillator is used as a crystal-controlled, positive-reactance oscillator (a more detailed schematic is given in **Figure 5-5**). It is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. The crystal specifications and capacitances are non-critical. In this circuit, 20 pF can be used as single capacitance at any frequency together with a good quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. If a ceramic resonator is used, the two capacitors normally will have different values, dependent on the oscillator frequency. We recommend consulting the manufacturer of the ceramic resonator for value specifications of these capacitors.

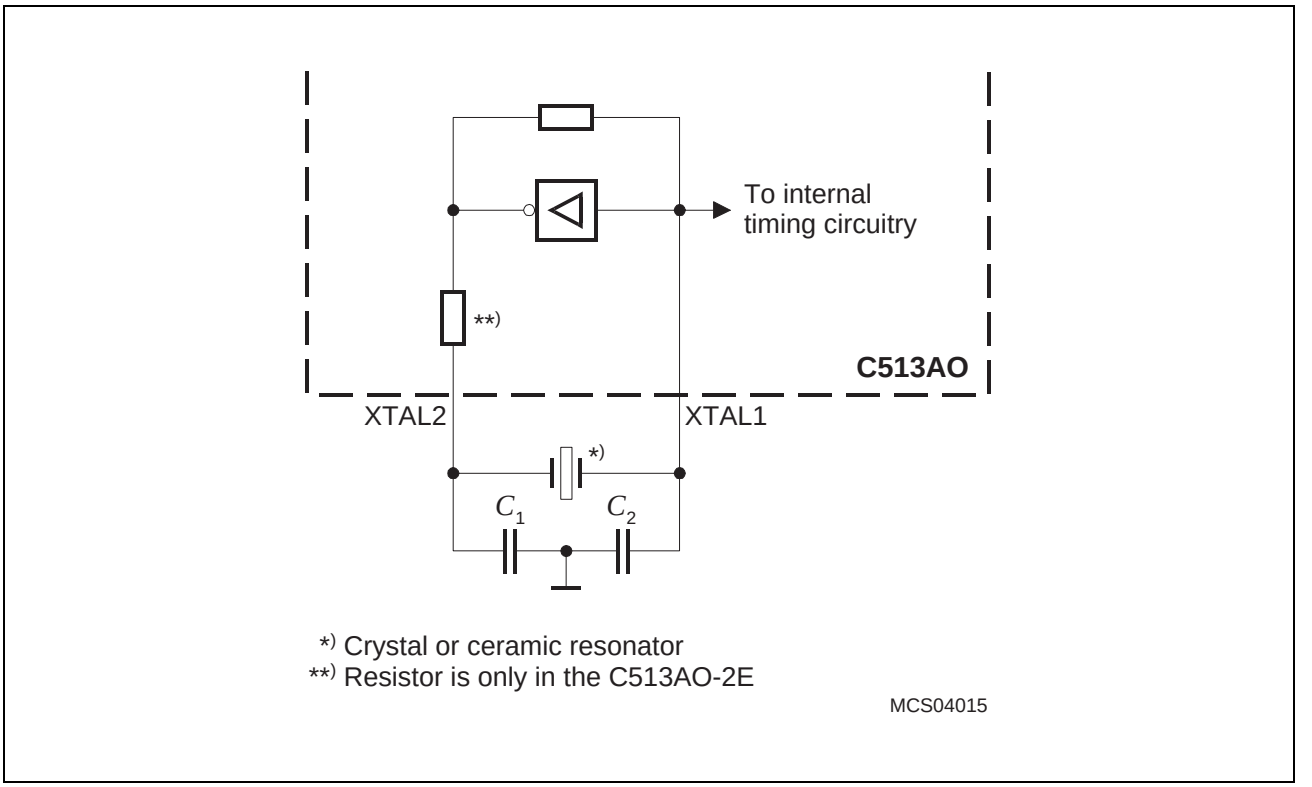


Figure 5-5
On-Chip Oscillator Circuitry

To drive the C513AO with an external clock source, the external clock signal must be applied to XTAL1, as shown in **Figure 5-6**. XTAL2 must be left unconnected. A pull-up resistor is suggested to increase the noise margin, but is optional if V_{OH} of the driving gate corresponds to the V_{IH3} specification of XTAL1.

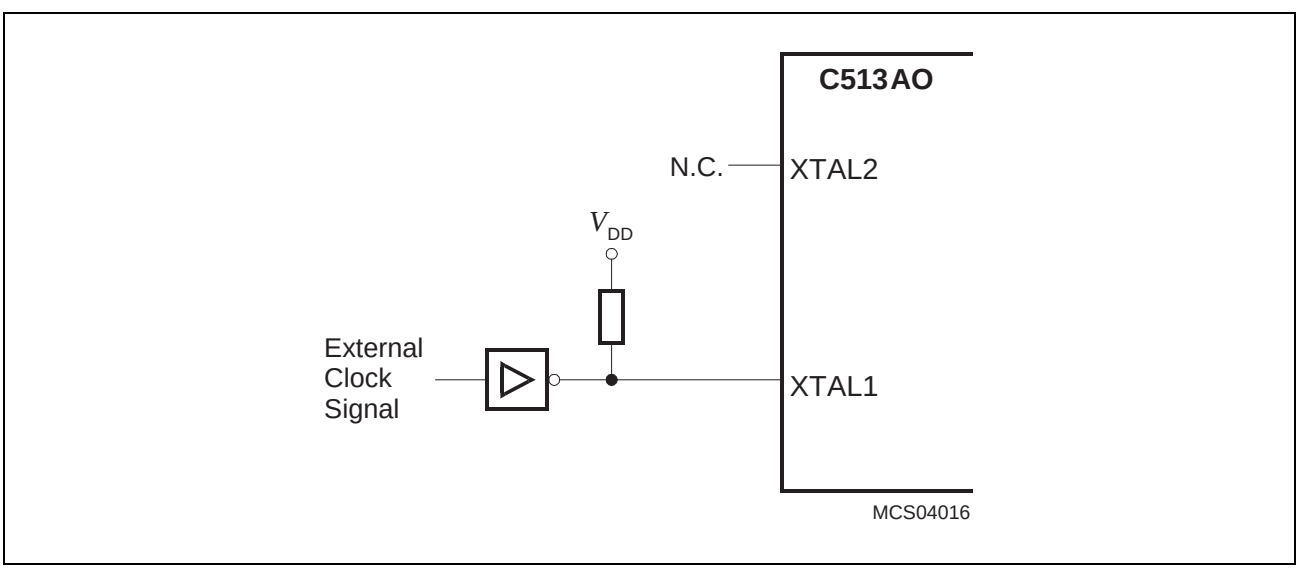


Figure 5-6
External Clock Source

6 On-Chip Peripheral Components

6.1 Parallel I/O

The C513AO has four 8-bit I/O ports. Port 0 is an open-drain bidirectional I/O port, while Ports 1, 2, and 3 are quasi-bidirectional I/O ports with internal pull-up resistors. Thus, 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 Port 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 to emit the P2 SFR contents. In this case, Port 0 is not an open-drain port, but uses a strong internal pull-up Field Effect Transistors (FETs).

Port 1 pins used for Synchronous Serial Channel (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 6-1 shows a functional diagram of a typical 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 and the actual state of the port pins can be read, depending on the instruction used for accessing the port.

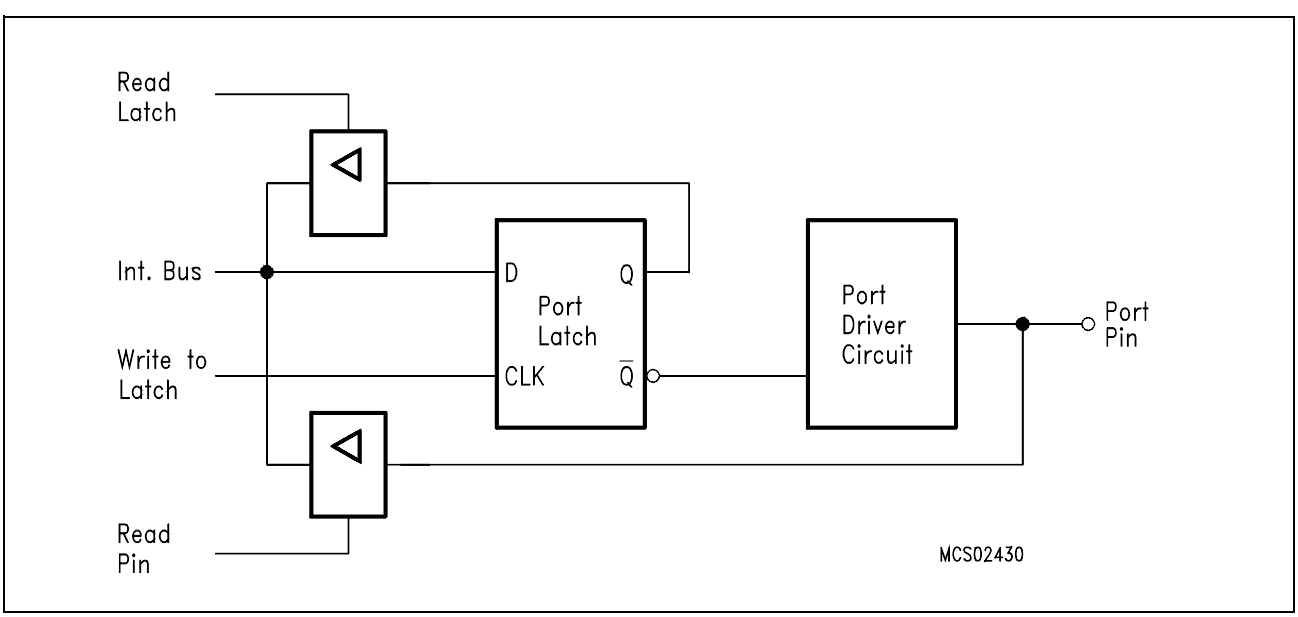


Figure 6-1
Basic Structure of Port Circuitry

Ports 1, 2, and 3 output drivers have internal pull-up FETs (see **Figure 6-2**). 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 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 pull-ups, 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”.

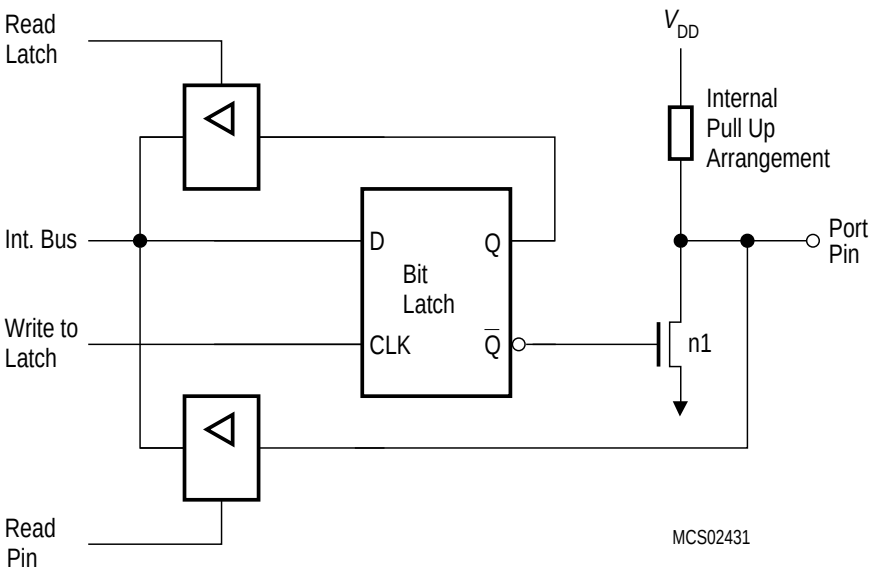


Figure 6-2
Basic Output Driver Circuit of Ports 1, 2, and 3

In fact, the pull-ups mentioned before, and included in **Figure 6-2,** are pull-up arrangements shown in **Figure 6-3**.

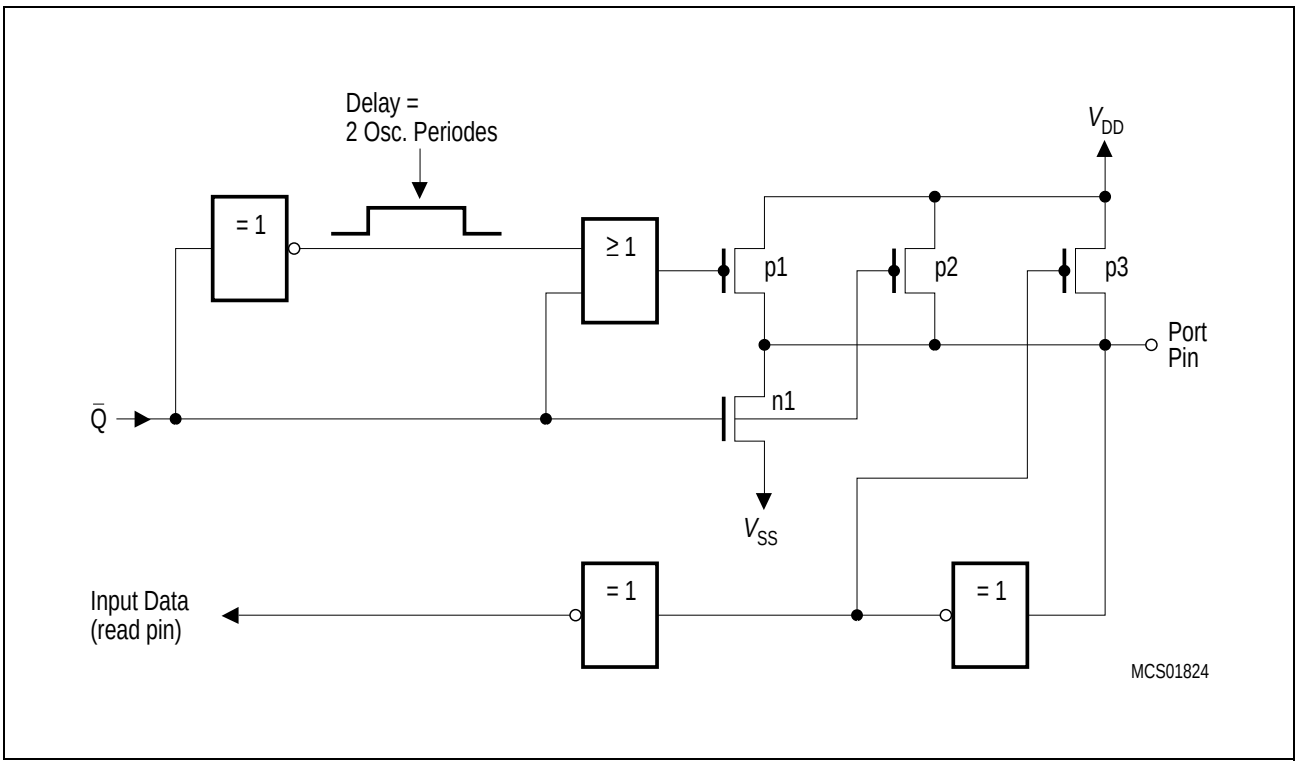


Figure 6-3
Output Driver Circuit of Ports 1, 2 and 3 (except P1.2, P1.3, P1.4 and P1.5)

One n-channel pull-down FET and three pull-up FETs are used in the example shown in **Figure 6-3**.

- The **pull-down FET n1** is of n-channel type. It is a very strong 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_{DD} 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 **pull-up FET p1** is of p-channel type. It is activated for one state (S1) if a 0-to-1 transition is programmed to the port pin; that is, a “1” is programmed to the port latch which contained a “0”. The extra pull-up can drive a current similar to the pull-down FET n1. This provides a fast transition of the logic levels at the pin.
- The **pull-up 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 pull-up FET sources a much lower current than p1. Therefore, the pin may also be tied to the ground; for example, when used as input with logic low input level.
- The **pull-up FET p3** is of p-channel type. It is activated only if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pull-up 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; that is, when used as input. In this configuration, only the weak pull-up FET p2 is active, which sources the current I_{IL}. If, in addition, the pull-up 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 activating and deactivating of the four different transistors results in one of these four states:

- 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 a “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 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 weak “1” until the first 0-to-1 transition on the latch occurs. Until this occurs, the output level might stay below the trip point of the external circuitry.

The same is true if a pin is used as a 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 have been modified to provide the following features when used for the alternate functions of the SSC:

- 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 pull-ups
- 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 **Figure 6-4** and **Figure 6-5**.

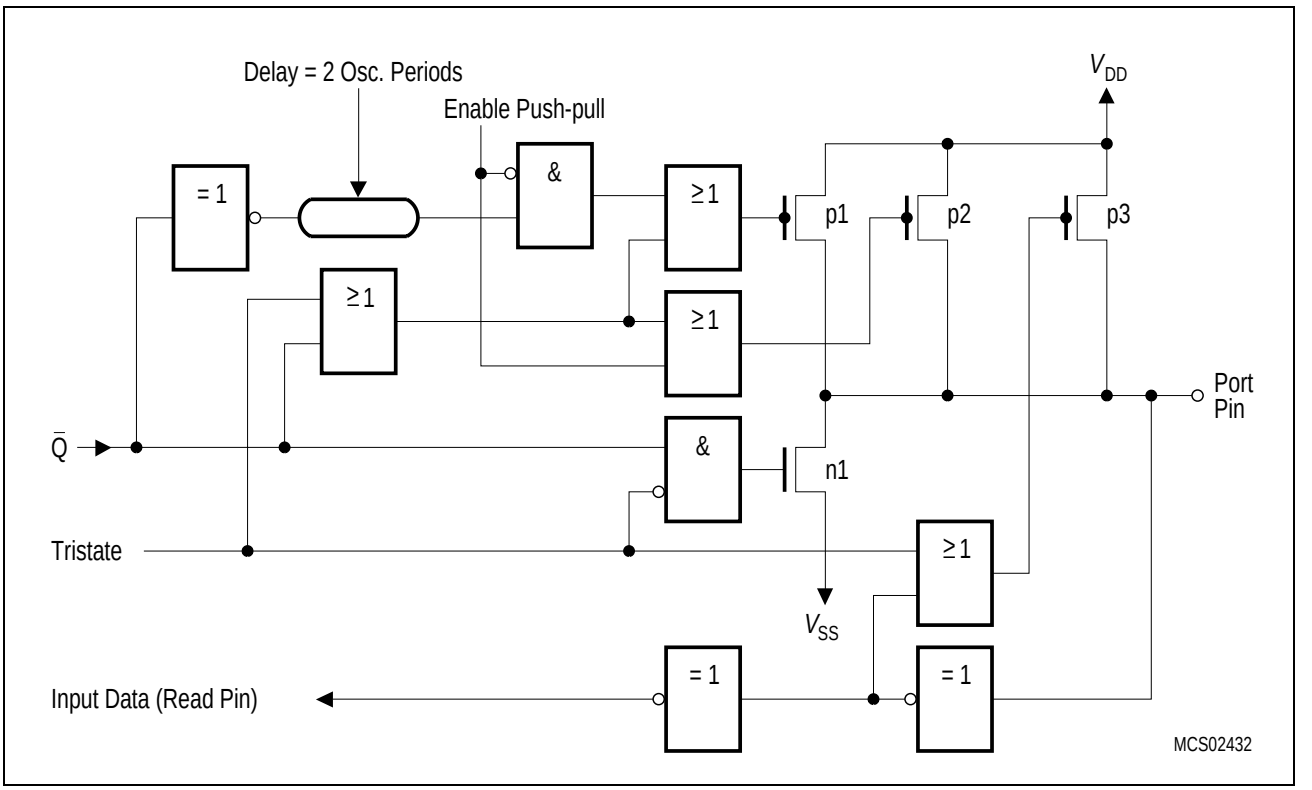


Figure 6-4
Driver Circuit of Port 1 Pins P1.2 and P1.4 (when used for SLCK and STO)

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

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

Pin Control for SCO

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

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

In Slave Mode with 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.

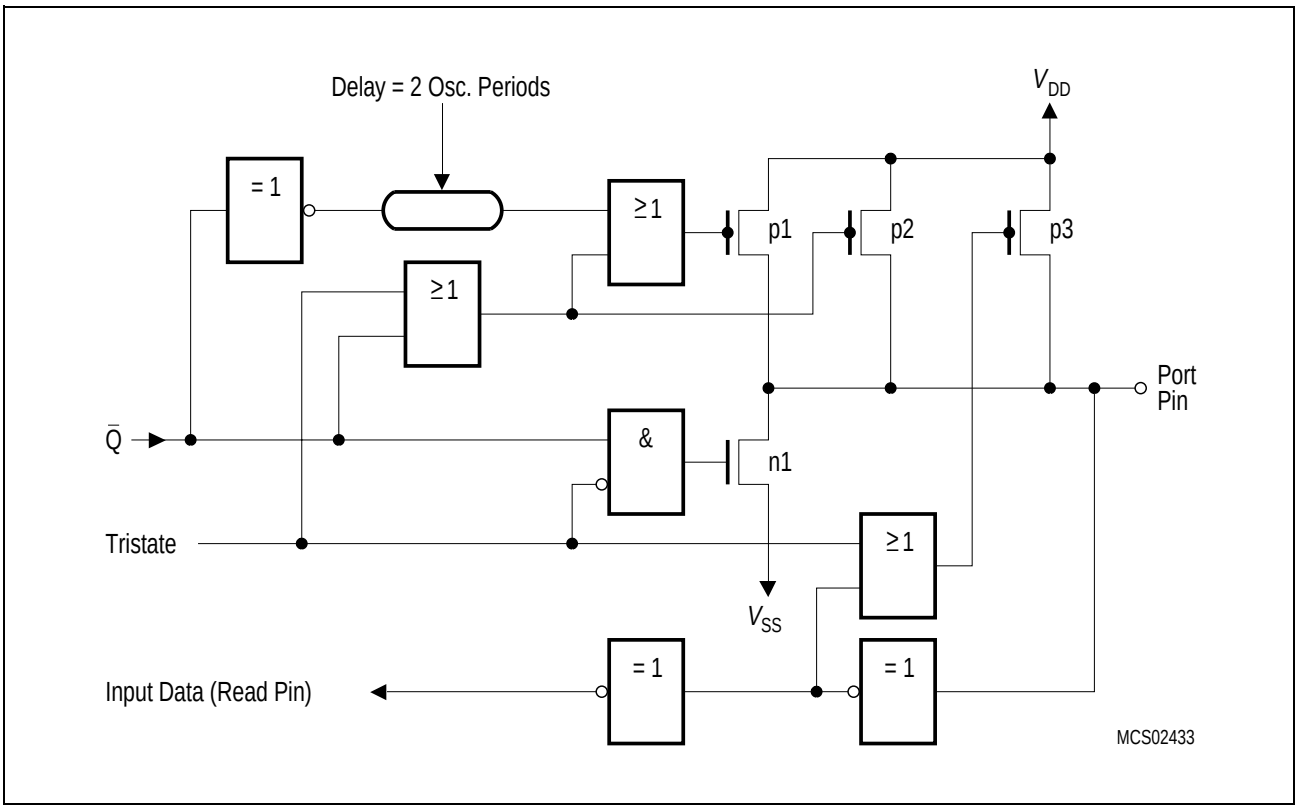


Figure 6-5
Driver Circuit of Port 1 Pins P1.3 and P1.5 (when used for SRI and $\overline{SLS}$)

When enabling the SSC, inputs used for the SSC will be switched into 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 pull-ups. The pullup FET in the P0 output driver (see **Figure 6-6**) is used only when the port is emitting “1”s during the external memory accesses. Otherwise, the pull-up 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 a general I/O port and has to emit logic high-level (1), external pull-ups are required.

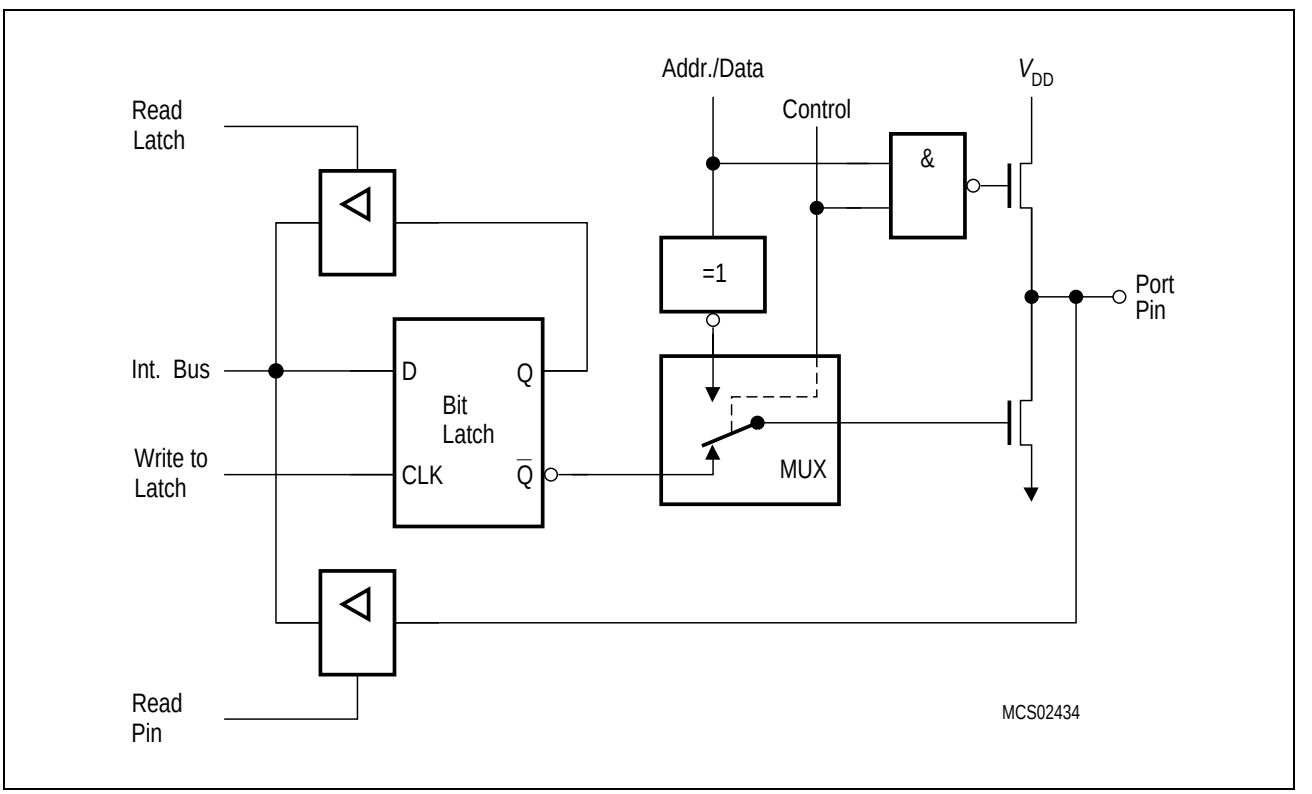


Figure 6-6
Port 0 Circuitry

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

As shown in **Figure 6-6** and in **Figure 6-7** respectively, the output drivers of Port 0 and Port 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 6-7**. 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.

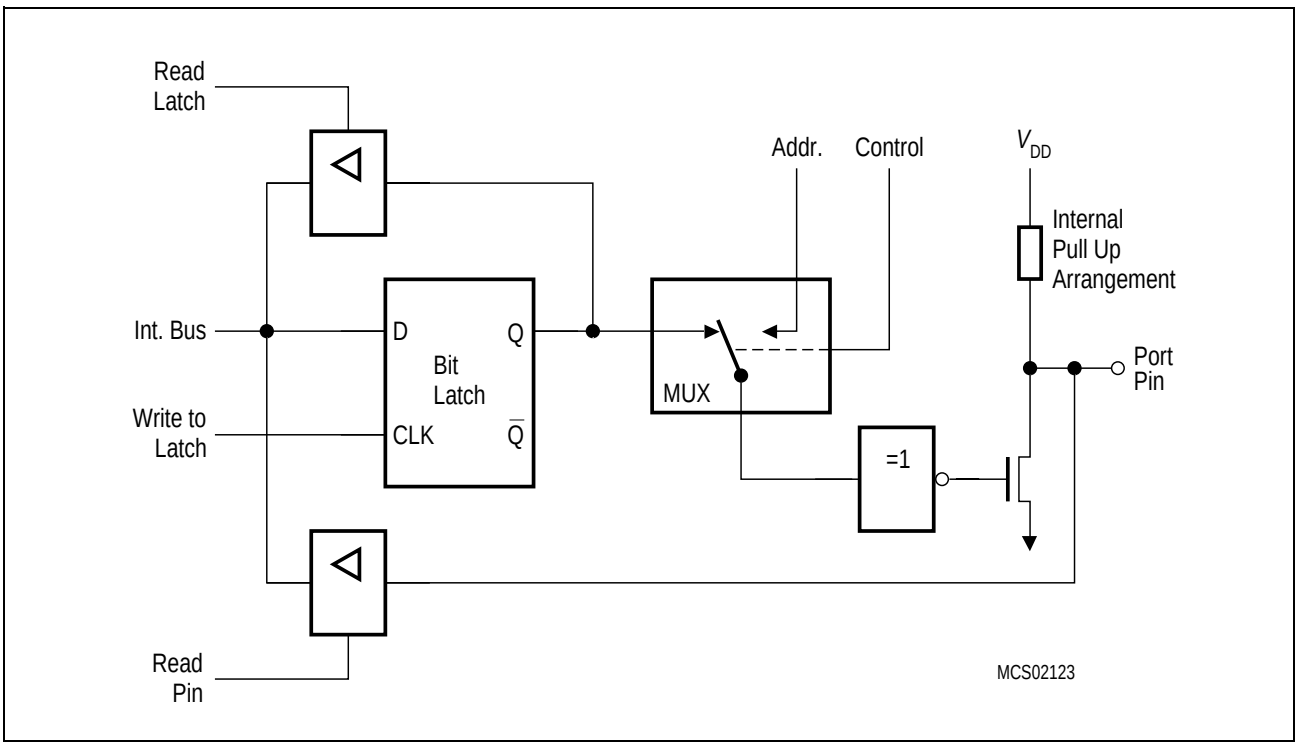


Figure 6-7
Port 2 Circuitry

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 6-1**.

Figure 6-7 shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, the gate between the latch and the driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR must contain a “1”; otherwise the pull-down FET is on and the port pin is stuck at “0”. After reset, all port latches contain “1”s.

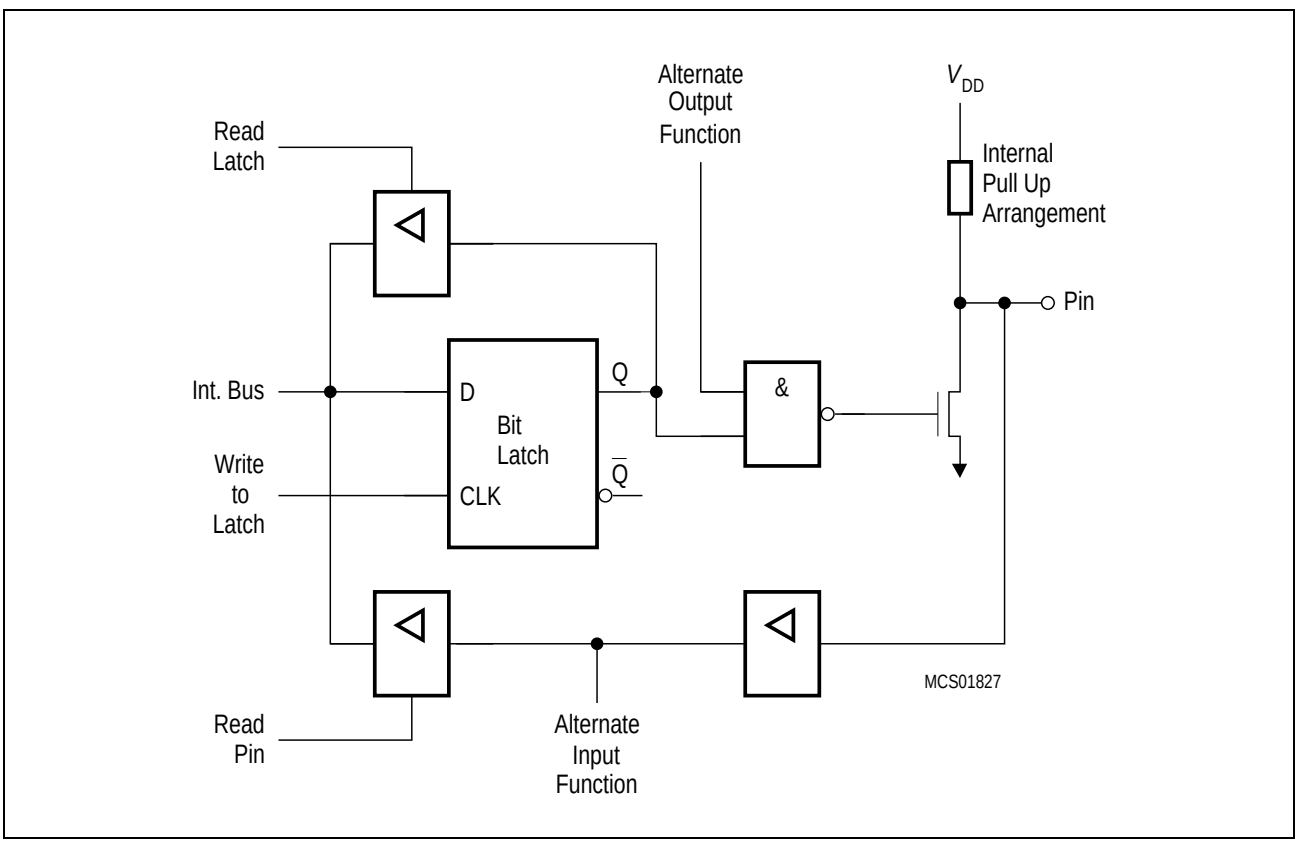


Figure 6-8
Ports 1 and 3

Different structures apply for port pins P1.2 to P1.5, see **Figure 6-9** and **Figure 6-10**.

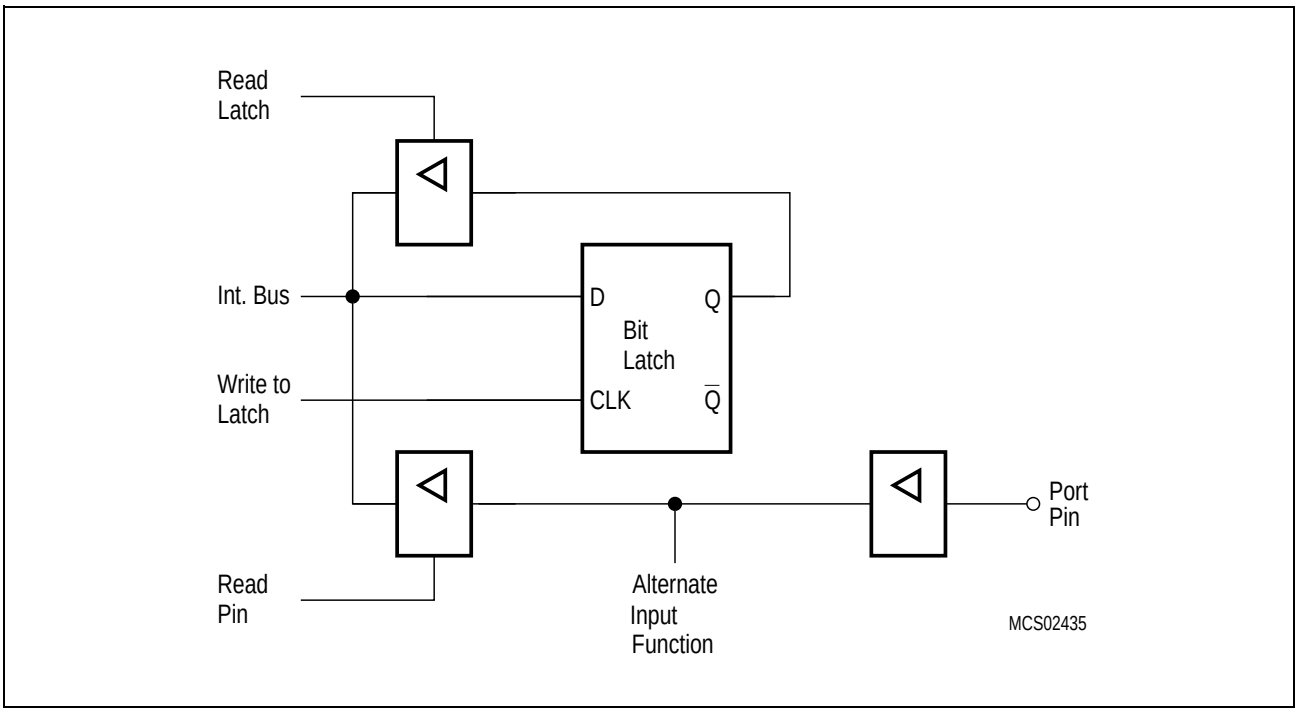


Figure 6-9
Port Pins P1.2, P1.3 and P1.5 (when used as SSC inputs)

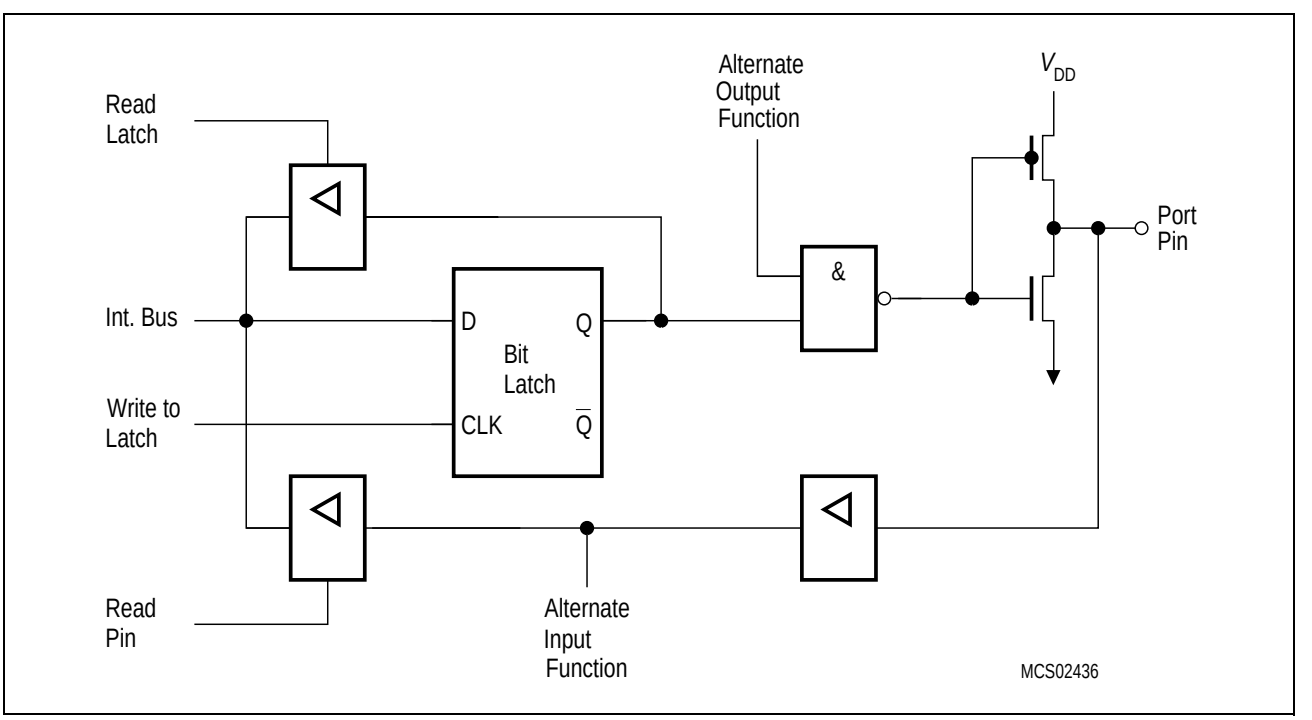


Figure 6-10
Port Pins P1.2 and P1.4 (when used as SSC outputs)

Ports 1 and 3 provide several alternate functions as listed in **Table 6-1**.

Table 6-1
Alternate Functions of Ports 1 and 3

Port	Symbol	Function
P1.0	T2	Input to Counter 2
P1.1	T2EX	Capture-reload trigger of Timer 2/up-down count
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
P3.1	TXD	Serial port's transmitter data output (asynchronous) or data clock input
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 Timing

When executing an instruction which 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 (such as MOV A, P1), the port pin is actually sampled in state 5 phase 1 or phase 2 depending on port and alternate functions. **Figure 6-11** 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”; for example, when used as a counter input. In this case, an “edge” is detected when the sampled value differs from the value that was sampled in the previous cycle. Therefore, certain requirements must be met on the pulse length of signals 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.

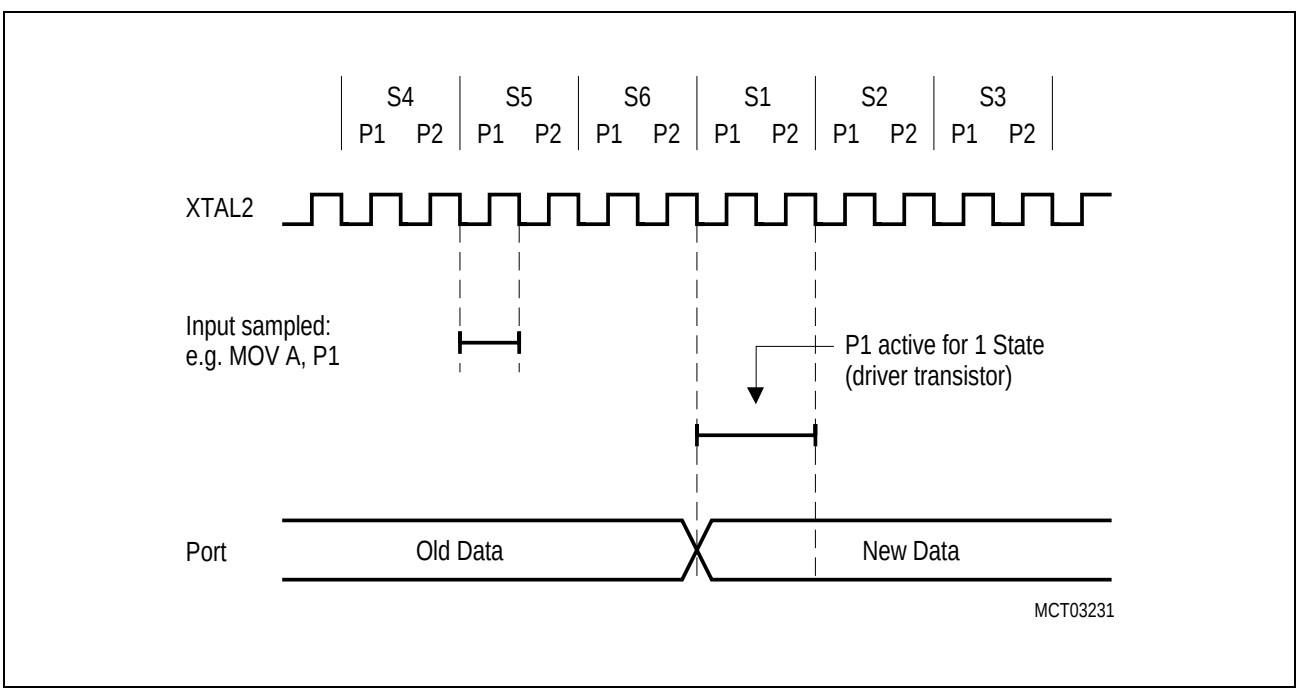


Figure 6-11
Port Timing

6.1.5 Port Loading and Interfacing

The output buffers of Ports 2 and 3 can drive TTL inputs directly. Refer to the DC characteristics in the Data Sheet of the C513AO for the maximum port load which still guarantees correct logic output levels. The corresponding parameters are V_{OL} and V_{OH}.

The output buffers of Port 0 can also drive TTL inputs directly. They do, however, require external pull-ups to drive floating inputs, except when used as the address/data bus.

It must be noted that when used as inputs, Ports 2 and 3 are not floating but have internal pull-up transistors. The driving devices must be capable of sinking a sufficient current if a logic low level is applied to the port pin (Parameters I_{TL} and I_{IL} in the DC Characteristics of the C513AO Data Sheet specify these currents). Port 0 has floating inputs when used for digital input.

6.1.6 Read-Modify-Write Feature of Ports 2 and 3

Some port-reading instructions read the latch and others read the pin. 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, and are listed in **Table 6-2**. 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 rather than the latch. In all cases, whether from the latch or the pin, the instruction is performed by reading the SFR P0, P2 and P3 as in the op-code; for example, “MOV A, P3” reads the value from Port 3 pins, while “ANL P3, #0AAH” reads from the latch, modifies the value, and writes it back to the latch.

Although it is not obvious, the last three instructions in **Table 6-2** are “read-modify-write” instructions. They are such because they read all 8 bits of the port byte, modify the addressed bit, then write the complete byte back to the latch.

Table 6-2
“Read-Modify-Write” Instructions

Instruction	Function
ANL	Logic AND; for example, ANL P1, A
ORL	Logic OR; for example, ORL P2, A
XRL	Logic exclusive OR; for example, XRL P3, A
JBC	Jump if bit is set and clear bit; for example, JBC P1.1, LABEL
CPL	Complement bit; for example, CPL P3.0
INC	Increment byte; for example, INC P1
DEC	Decrement byte; for example, DEC P1
DJNZ	Decrement and jump if not zero; for example, DJNZ P3, EL
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

“Read-modify-write” instructions are directed to the latch rather than the pin 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; that is, 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 rate than the pin will return the correct value of “1”.

6.2 Timers/Counters

The C513AO contains three 16-bit timers/counters which are useful in many applications.

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 twelve oscillator periods, the counter rate is 1/12th 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, respectively). In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high level in one cycle and a low level in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. It takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition; therefore, the maximum count rate is 1/24th 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 Timer/Counter 1 of the C513AO are fully compatible with Timer/Counter 0 and Timer/Counter 1 of the C501 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 configured as one 8-bit timer/counter and one 8-bit timer
(In this mode, Timer/counter 1 holds its count. The effect is the same as setting TR1 = 0).

External inputs INT0 and INT1 can be programmed to function as a gate for Timer/Counters 0 and 1 to facilitate pulse-width measurements.

Each timer/counter consists of two 8-bit registers (TH0 and TL0 for Timer/Counter 0; TH1 and TL1 for Timer/Counter 1). They may be combined into 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, TH0 and TL0 are used to specify the high-byte and the low-byte of Timer 0; TH1 and TL1 are used to specify that of Timer 1. The operating modes are described and shown for Timer 0, and apply also to Timer 1 if not explicitly noted.

6.2.1.1 Timer/Counter 0 and 1 Registers

Six special function registers control Timer/Counter 0 and 1 operation:

- TL0/TH0 and TL1/TH1 are counter registers with low and high bytes.
- TCON and TMOD are control and mode select registers.

Special Function Register TL0 (Address $8A_H$) **Reset Value: 00_H**
Special Function Register TH0 (Address $8C_H$) **Reset Value: 00_H**
Special Function Register TL1 (Address $8B_H$) **Reset Value: 00_H**
Special Function Register TH1 (Address $8D_H$) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$8A_H$	.7	.6	.5	.4	.3	.2	.1	.0	TL0
$8C_H$	.7	.6	.5	.4	.3	.2	.1	.0	TH0
$8B_H$	.7	.6	.5	.4	.3	.2	.1	.0	TL1
$8D_H$	.7	.6	.5	.4	.3	.2	.1	.0	TH1

Bit	Function	
TLx.7-0 x = 0-1	**Timer/Counter 0/1 Low Register**	
	Operating Mode	**Description**
	0	“TLx” holds the 5-bit prescaler value.
	1	“TLx” holds the lower 8-bit part of the 16-bit timer/counter value.
	2	“TLx” holds the 8-bit timer/counter value.
	3	TL0 holds the 8-bit timer/counter value; TL1 is not used.
THx.7-0 x = 0-1	**Timer/Counter 0/1 High Register**	
	Operating Mode	**Description**
	0	“THx” holds the 8-bit timer/counter value.
	1	“THx” holds the higher 8-bit part of the 16-bit timer/counter value
	2	“THx” holds the 8-bit reload value.
	3	TH0 holds the 8-bit timer value; TH1 is not used.

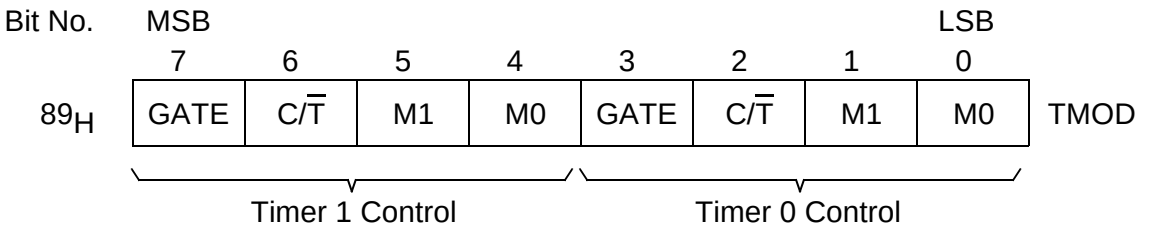
Special Function Register TCON (Address 88_H) **Reset Value: 00_H**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
	$8F_H$	$8E_H$	$8D_H$	$8C_H$	$8B_H$	$8A_H$	89_H	88_H	
88_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

The shaded 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 “INT x” 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	Mode select bits

M1	M0	Function
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. Timer 1: Timer/Counter 1 stops

6.2.1.2 Mode 0

Putting either Timer/Counter 0 or Timer/Counter 1 into Mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. **Figure 6-12** shows 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. 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 INT0 for the corresponding Timer 1 signals shown in **Figure 6-12**. There are, however, two different gate bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

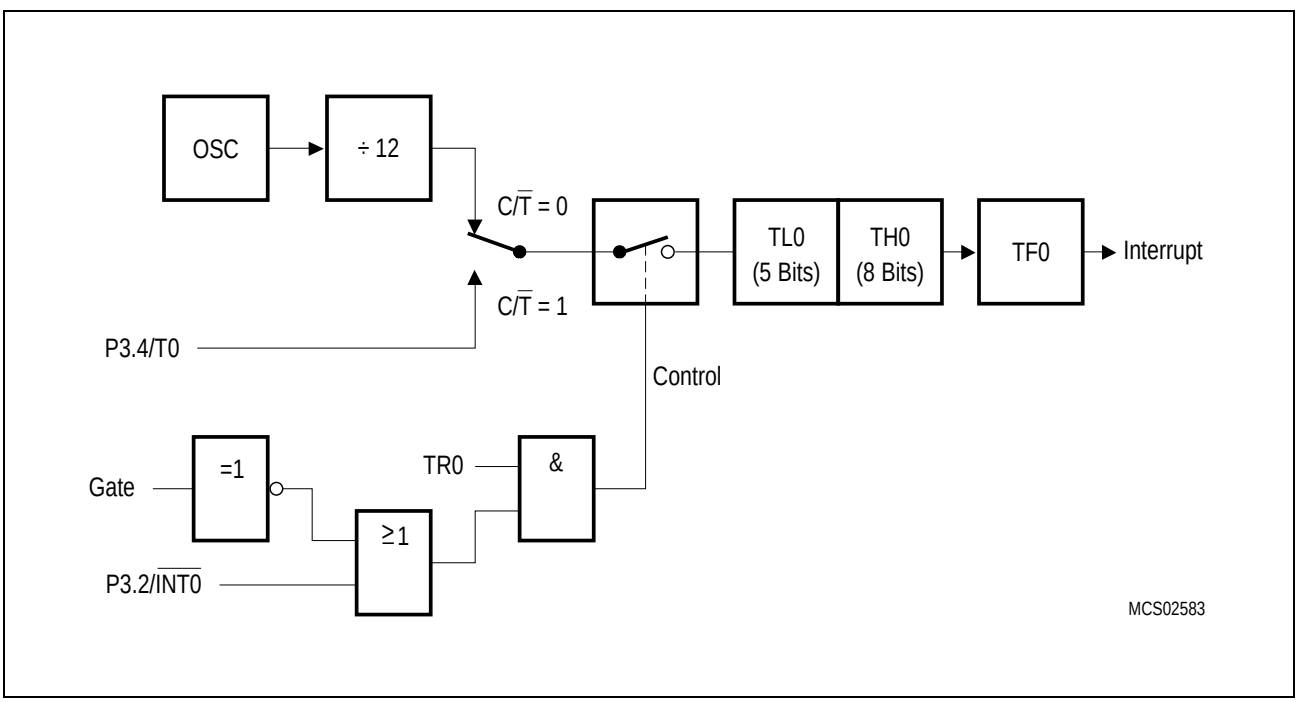


Figure 6-12
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

6.2.1.3 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 6-13**.

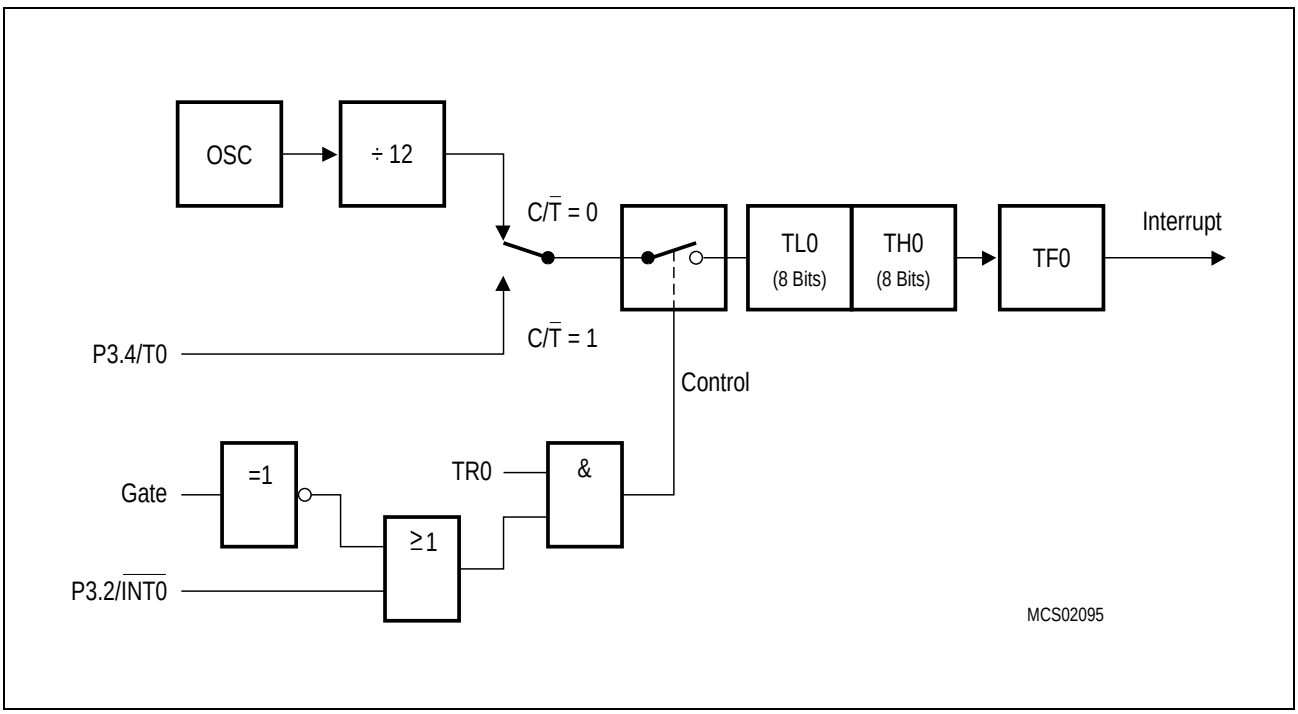


Figure 6-13
Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

6.2.1.4 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in **Figure 6-14**. 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.

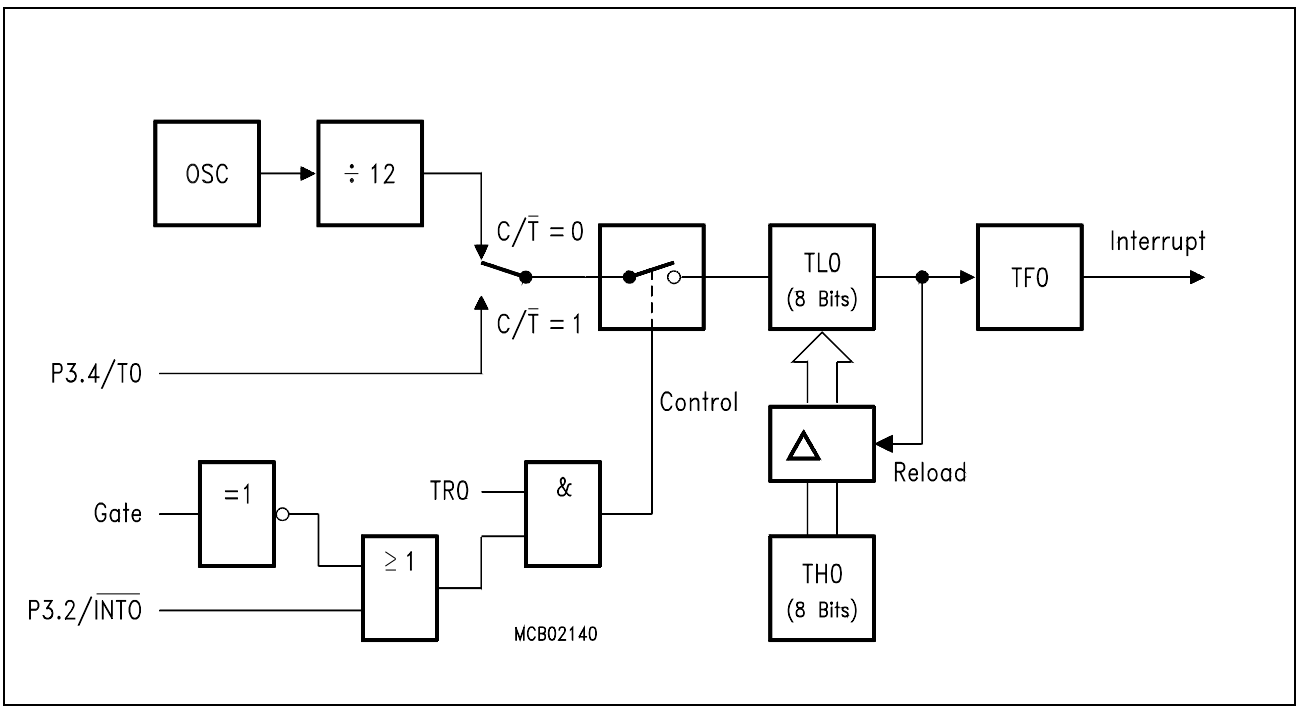


Figure 6-14
Timer/Counter 0,1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

6.2.1.5 Mode 3

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in **Figure 6-15**. TL0 uses the Timer 0 control bits: C/T, Gate, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over 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 in and out of its own Mode 3. Alternatively, Timer 1 can still be used by the serial channel as a baudrate generator in this or any application not requiring an interrupt from Timer 1.

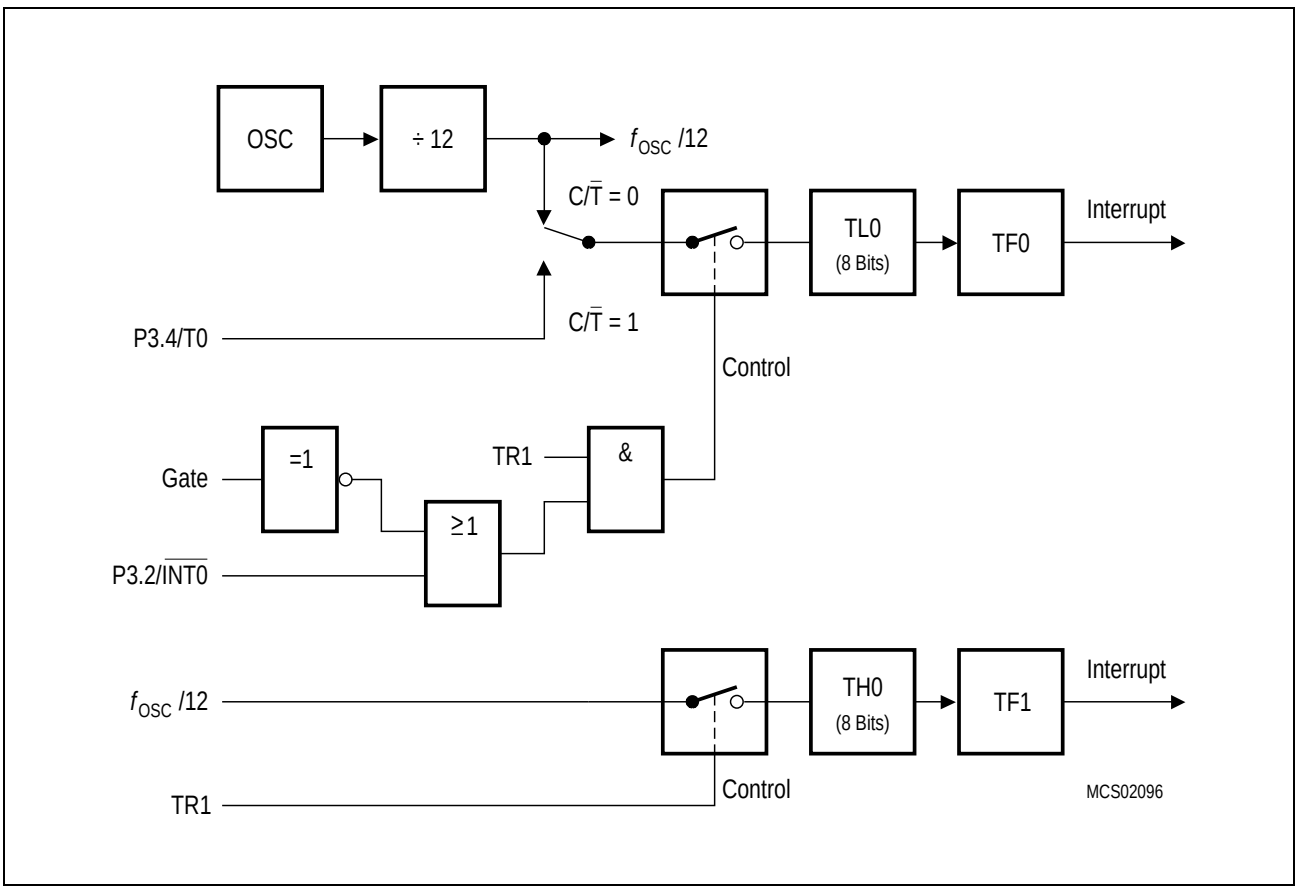


Figure 6-15
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

6.2.2 Timer/Counter 2

Timer 2 is a 16-bit timer/counter and has three operating modes:

- 16-bit auto-reload mode (up or down counting)
- 16-bit capture mode
- Baudrate generator (see **Section 6.3.3** “Baudrates”)

The modes are selected by bits in the SFR T2CON as shown in **Table 6-3**:

Table 6-3
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	Baudrate 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 twelve oscillator periods, the count rate is 1/12th 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 level in one cycle and a low level 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. As two machine cycles are required to recognize a 1-to-0 transition, the maximum count rate is 1/24th of the oscillator 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.

6.2.2.1 Timer/Counter 2 Registers

Six special function registers control Timer/Counter 0 and 1 operation:

- TL2/TH2 and RC2L/RC2H are counter and reload/capture registers with low and high bytes.
- T2CON and T2MOD are control and mode select registers.

Special Function Register TL2 (Address CC_H) **Reset Value: 00_H**
Special Function Register TH2 (Address CD_H) **Reset Value: 00_H**
Special Function Register RC2L (Address CA_H) **Reset Value: 00_H**
Special Function Register RC2H (Address CB_H) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
CC_H	.7	.6	.5	.4	.3	.2	.1	.0	TL2
CD_H	.7	.6	.5	.4	.3	.2	.1	.0	TH2
CA_H	.7	.6	.5	.4	.3	.2	.1	.0	RC2L
CB_H	.7	.6	.5	.4	.3	.2	.1	.0	RC2H

Bit	Function
TL2.7-0	**Timer 2 value low byte** The TL2 register holds the 8-bit low part of the 16-bit Timer 2 count value.
TH2.7-0	**Timer 2 value high byte** The TH2 register holds the 8-bit high part of the 16-bit Timer 2 count value.
RC2L.7-0	**Reload / capture Timer 2 register low byte** RC2L holds the 8-bit low byte of the 16-bit Timer 2 reload or capture value.
RC2H.7-0	**Reload / capture Timer 2 register high byte** RC2H holds the 8-bit high byte of the 16-bit Timer 2 reload or capture value.

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

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	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.

Special Function Register T2MOD (Address $C9_H$) **Reset Value: $XXXX\ XXX0_B$**

Bit No.	MSB							LSB	
	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.

6.2.2.2 Auto-Reload (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, SFR T2MOD, $0C9_H$). When DCEN is set, Timer 2 can count up or down depending on the value of pin T2EX (P1.1).

Figure 6-16 shows Timer 2 automatically counting up when DCEN = 0. In this mode, there are two options selectable by bit EXEN2 in SFR T2CON.

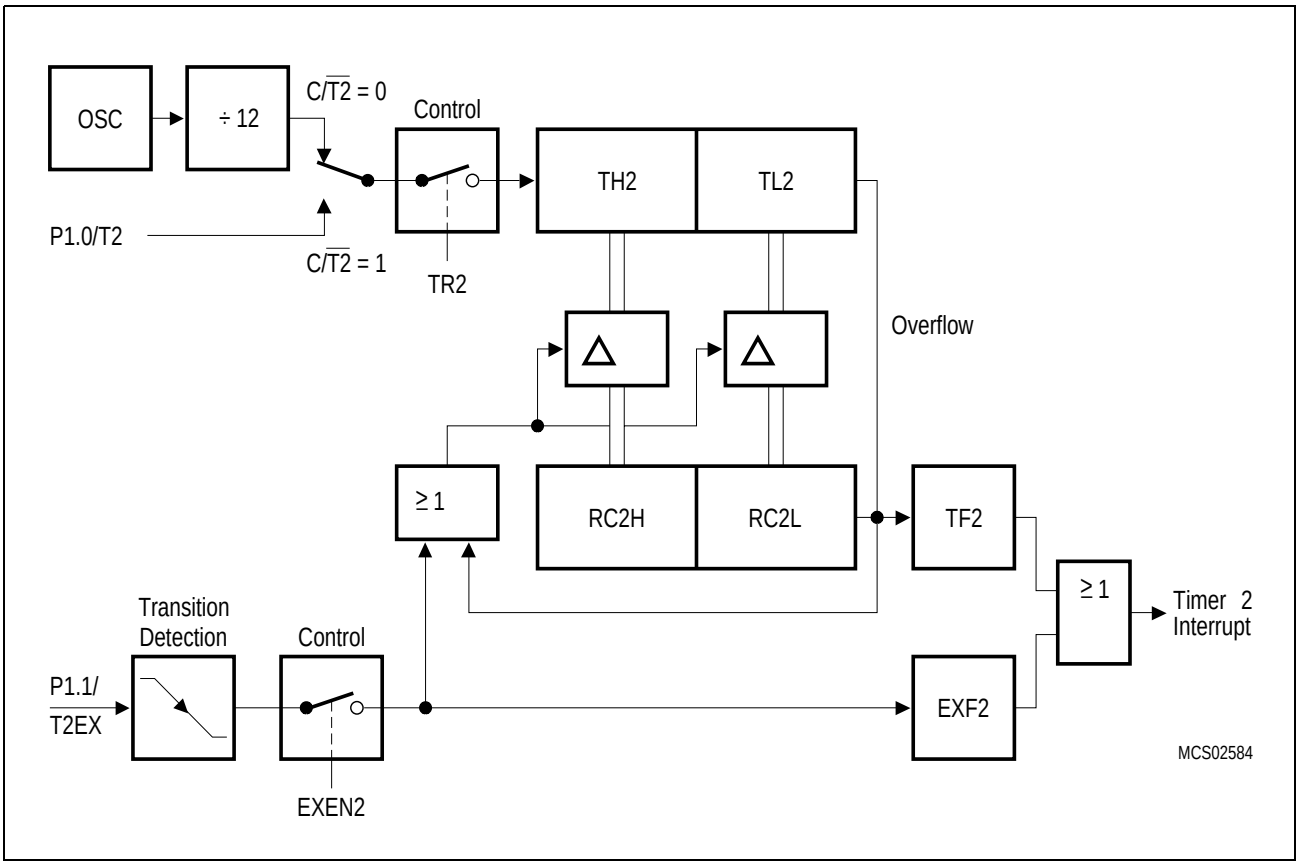


Figure 6-16
Timer 2 Auto-Reload Mode (DCEN = 0)

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 a Timer 2 interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down as shown in **Figure 6-17**. In this mode, the T2EX pin controls the direction of count.

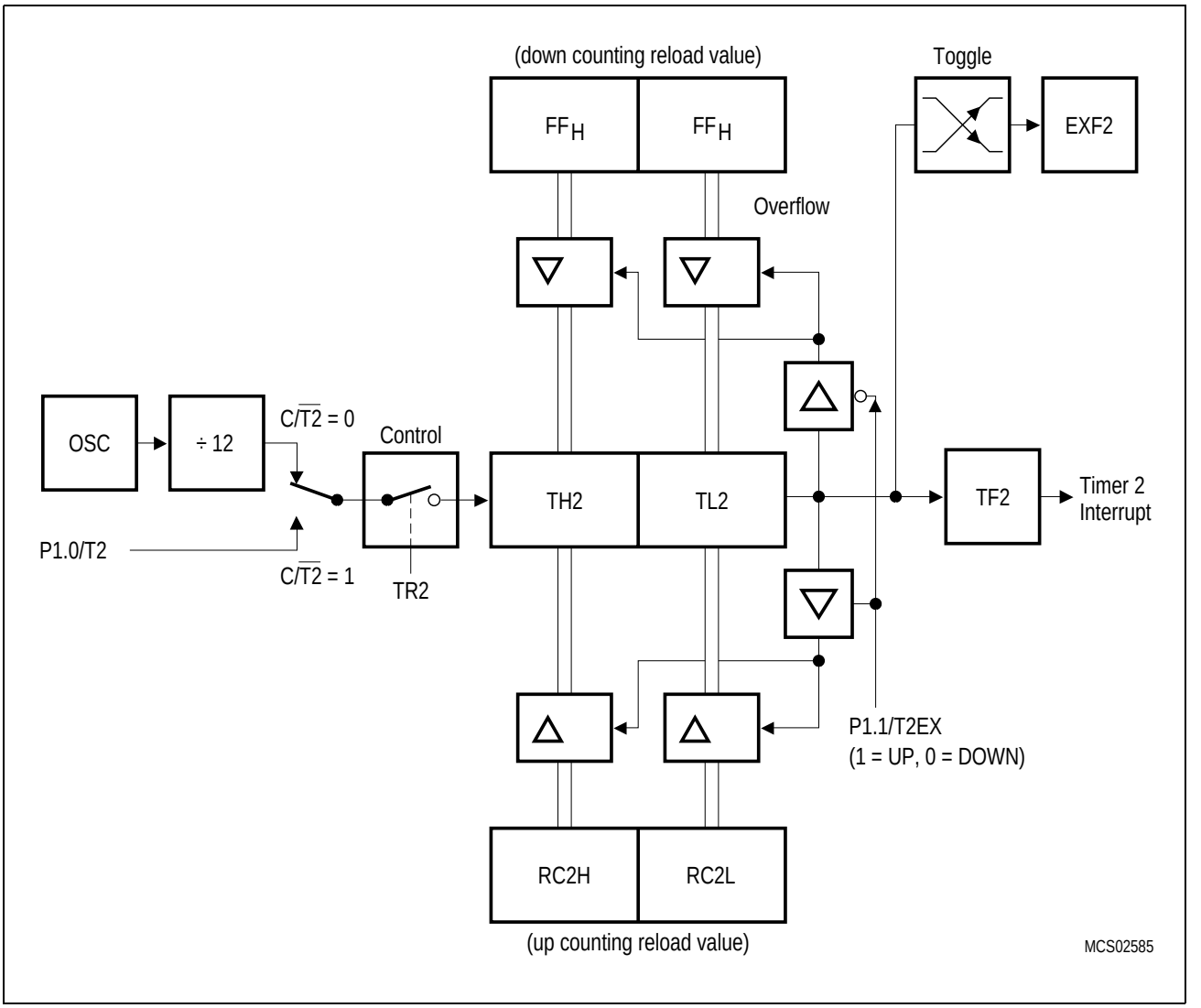


Figure 6-17
Timer 2 Auto-Reload Mode (DCEN = 1)

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.

6.2.2.3 Capture Mode

In 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 sets bit TF2 in SFR T2CON on overflow. This bit can be used to generate an interrupt.

If EXEN2 = 1, as well as setting bit TF2 as above, Timer 2 has the 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. Additionally, a transition at T2EX causes bit EXF2 in SFR T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. Capture Mode is illustrated in **Figure 6-18**.

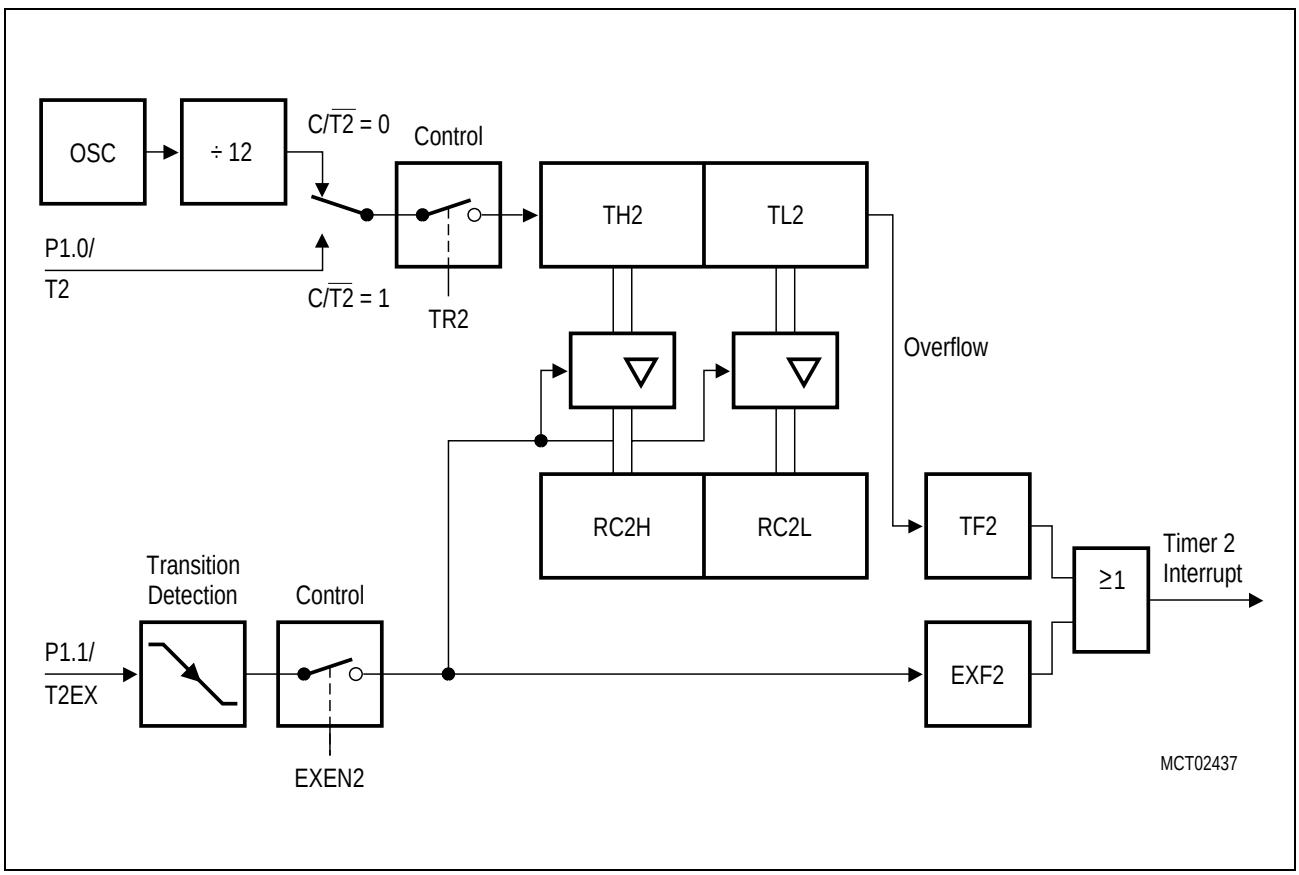


Figure 6-18
Timer 2 in Capture Mode

Note: Baudrate Generator Mode is selected by RCLK = 1 and/or TCLK = 1 in SFR T2CON. It is described in conjunction with the serial port.

6.3 Serial Interface (USART)

The serial port is a full duplex port capable of simultaneous transmit and receive functions. It is also receive-buffered; 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 has not been read before reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are 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 and three asynchronous):

Mode 0, Shift Register (Synchronous) Mode:

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 data bits are transmitted/ received with the Least Significant Bit (LSB) first. The baudrate is fixed at $^1/_{12}$th of the oscillator frequency.

Mode 1, 8-Bit USART, Variable Baudrate:

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 special function register SCON. The baudrate in Mode 1 is variable.

Mode 2, 9-Bit USART, Fixed Baudrate:

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 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 baudrate in Mode 2 is programmable to either $^1/_{32}$nd or $^1/_{64}$th of the oscillator frequency.

Mode 3, 9-Bit USART, Variable Baudrate:

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). In fact, Mode 3 is the same as Mode 2 in all respects except the baudrate. The baudrate in Mode 3 is variable.

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 incoming start bit if REN = 1.

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 incoming start bit if REN = 1. The serial interface also provides interrupt requests when transmission or reception of a frame is completed. The corresponding interrupt request flags for the serial interface are TI or RI, respectively. See **Chapter 7** for more details about the interrupt structure. The interrupt request flags, TI and RI, can also be used for polling the serial interface if the serial interrupt is not to be used (that is, serial interrupt is not enabled).

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 one 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. One use of this feature in multiprocessor systems is described here.

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 to determine if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive data bytes. The slaves which weren't being addressed keep their SM2s set and ignore the incoming data bytes.

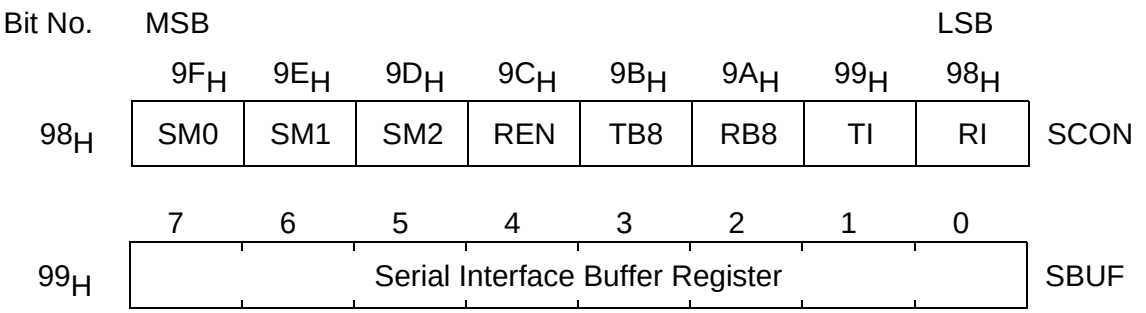
SM2 has no effect in Mode 0; in Mode 1, it can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

6.3.2 Serial Port Registers

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).

SBUF is the receive and transmit buffer of the serial interface. Writing to SBUF loads the transmit register and initiates transmission. Reading out SBUF accesses a physically separate receive register.

Special Function Register SCON (Address 98_H) **Reset Value: 00_H**
Special Function Register SBUF (Address 99_H) **Reset Value: XX_H**



Bit	Function
SM0 SM1	**Serial port operating Mode selection bits** **SM0** / **SM1** / **Selected Operating Mode** 0 / 0 / Serial Mode 0: Shift register, fixed baudrate (f_{OSC}/12) 0 / 1 / Serial Mode 1: 8-bit UART, variable baudrate 1 / 0 / Serial Mode 2: 9-bit UART, fixed baudrate (f_{OSC}/32 or f_{OSC}/64) 1 / 1 / Serial Mode 3: 9-bit UART, variable baudrate
SM2	**Enable Serial port Multiprocessor communication 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	**Enable Receiver of serial port** Enables serial reception. Set by software to enable serial reception. Cleared by software to disable serial reception.
TB8	**Serial port Transmitter Bit 9** TB8 is the 9th data bit that will be transmitted in Modes 2 and 3. Set or cleared by software as desired.
RB8	**Serial port Receiver Bit 9** In Modes 2 and 3, RB8 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	**Serial port tRansmitter Interrupt flag** TI is 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. TI must be cleared by software.
RI	**Serial port Receiver Interrupt flag** RI is 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 (exception see SM2). RI must be cleared by software.

6.3.3 Baudrates

There are several possibilities for generating the baudrate clock for the serial interface, depending on the mode in which it is operated.

To clarify the terminology, something should be said about the differences between “baudrate clock” and “baudrate”.

The serial interface requires a clock rate which is 16 times the baudrate for internal synchronization. Therefore, the baudrate generators must provide a “baudrate clock” to the serial interface results in the actual “baudrate” after being divided by 16 by the serial interface.

Note: All formulae given below already include the factor and calculate the final baudrate.

The baudrate generation is further controlled by bit SMOD which is located in SFR PCON.

Special Function Register PCON (Address $87_{H)}$ **Reset Value: $0XX00000_B$**

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

The functions of the shaded bits are not described in this section.

Symbol	Function
SMOD	**Baudrate double bit** When set, the baudrate of the serial channel in Mode 1,2,or 3 is doubled.

Mode 0

The baudrate in Mode 0 is fixed:

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

Mode 2

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

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

Modes 1 and 3

The baudrates in Mode 1 and 3 are determined by the timer overflow rate. These baudrates can be determined by Timer 1 or by Timer 2 or both (one for transmit, the other for receive).

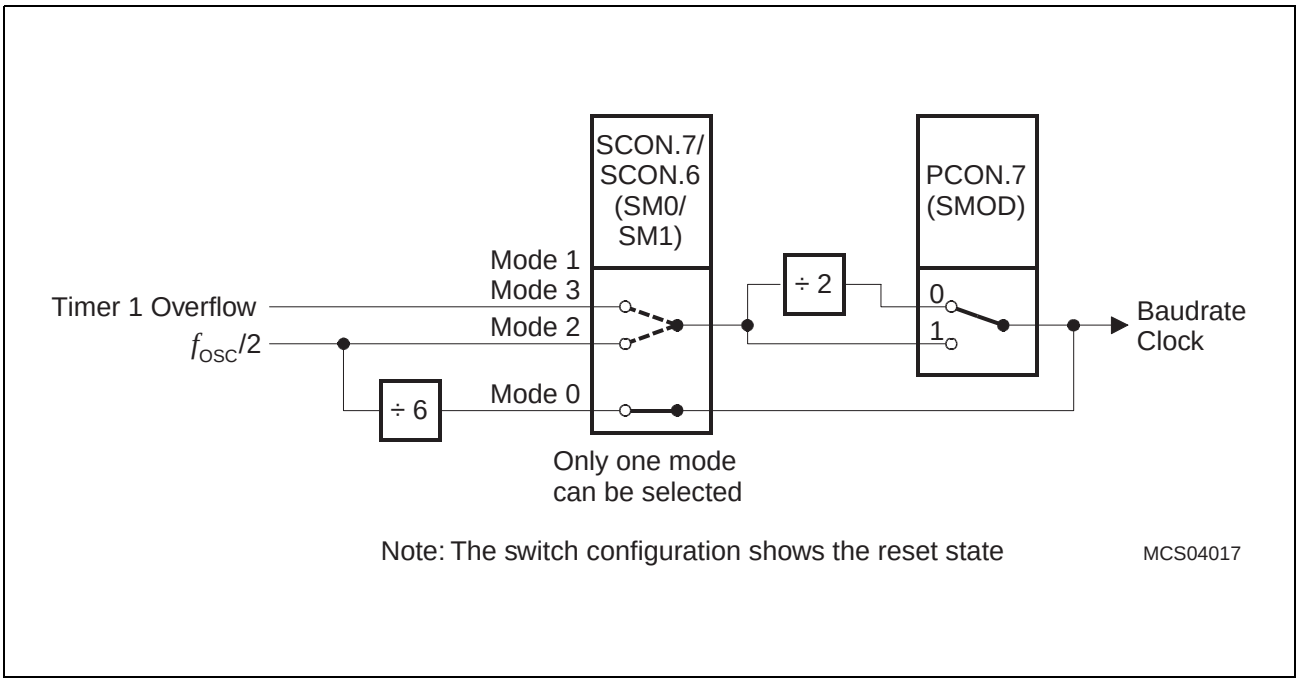


Figure 6-19
Baudrate Generation for Serial Channel

Figure 6-19 shows the configuration for the baudrate generation for the serial channel.

6.3.3.1 Using Timer 1 to Generate Baudrates

When Timer 1 is used as the baudrate generator, the baudrates in Modes 1 and 3 are determined by the timer 1 overflow rate and the value of SMOD as follows:

Modes 1 and 3 baudrate = $2^{SMOD}/32 \times$ (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 = 0010B). In that case, the baudrate is given by the formula

Modes 1 and 3 baudrate = $2^{SMOD}/32 \times f_{OSC}/[12 \times (256 - TH1)]$

One can achieve very low baudrates 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 = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Commonly used baudrates, and how they can be obtained from Timer 1, are listed in **Table 6-4**.

Table 6-4
Timer 1 Generated Commonly Used Baudrates

Baudrate	f_{osc}	SMOD	Timer 1		
			C/T	Mode	Reload Value
Mode 0 max: 1 MHz	12 MHz	X	X	X	X
Mode 2 max: 375 K	12 MHz	1	X	X	X
Modes 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 Baudrates

Timer 2 is selected as the baudrate generator by setting TCLK and/or RCLK in T2CON. Note that, simultaneously, the baudrates can be different for transmit and receive. Setting RCLK and/or TCLK puts Timer 2 into its Baudrate Generator Mode, as shown in **Figure 6-20**.

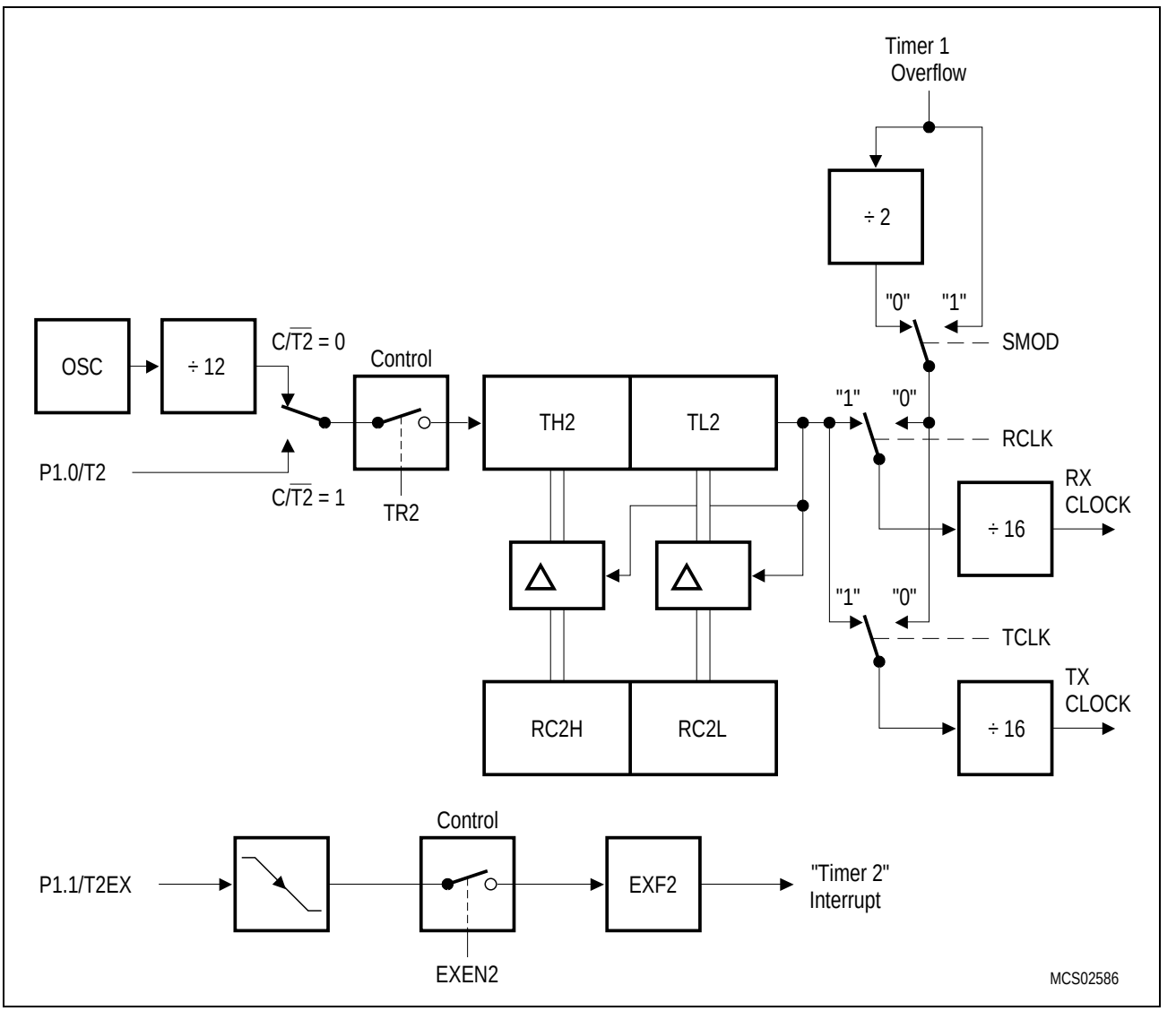


Figure 6-20
Timer 2 in Baudrate Generator Mode

Baudrate 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.

The baudrates in Modes 1 and 3 are determined by the overflow rate of Timer 2 as follows:

Modes 1 and 3 baudrate = 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 is used as a baudrate generator. Normally, as a timer it would increment every machine cycle

(thus at $f_{OSC}/12$). As a baudrate generator, however, it increments every state time ($f_{OSC}/2$). In that case, the baudrate is given by the formula

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

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 need not be disabled when Timer 2 is in Baudrate 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.

Note also that when Timer 2 is running (TR2 = 1) in “timer” function in Baudrate Generator Mode, TH2 or TL2 should not be read or written to. Under these conditions, the timer is incremented every state time, and the results of a read or write may not be accurate. The RC registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. In such a case, turn the timer off (clear TR2) before accessing Timer 2 or RC registers.

6.3.4 Details about Mode 0

Serial data enters and exists through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received with LSB first. The baudrate is fixed at $f_{OSC}/12$. **Figure 6-21** shows a simplified functional diagram of the serial port in Mode 0. The associated timing is illustrated in **Figure 6-22**.

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 output of the shift register to the alternate output function line of P3.0, and 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, “0”s come in from the left. When the Most Significant Bit (MSB) of the data byte is at the output position of the shift register, the “1” which was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain “0”s. 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 a data bit comes in from the right, “1”s shift out to the left. When the “0” which 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.

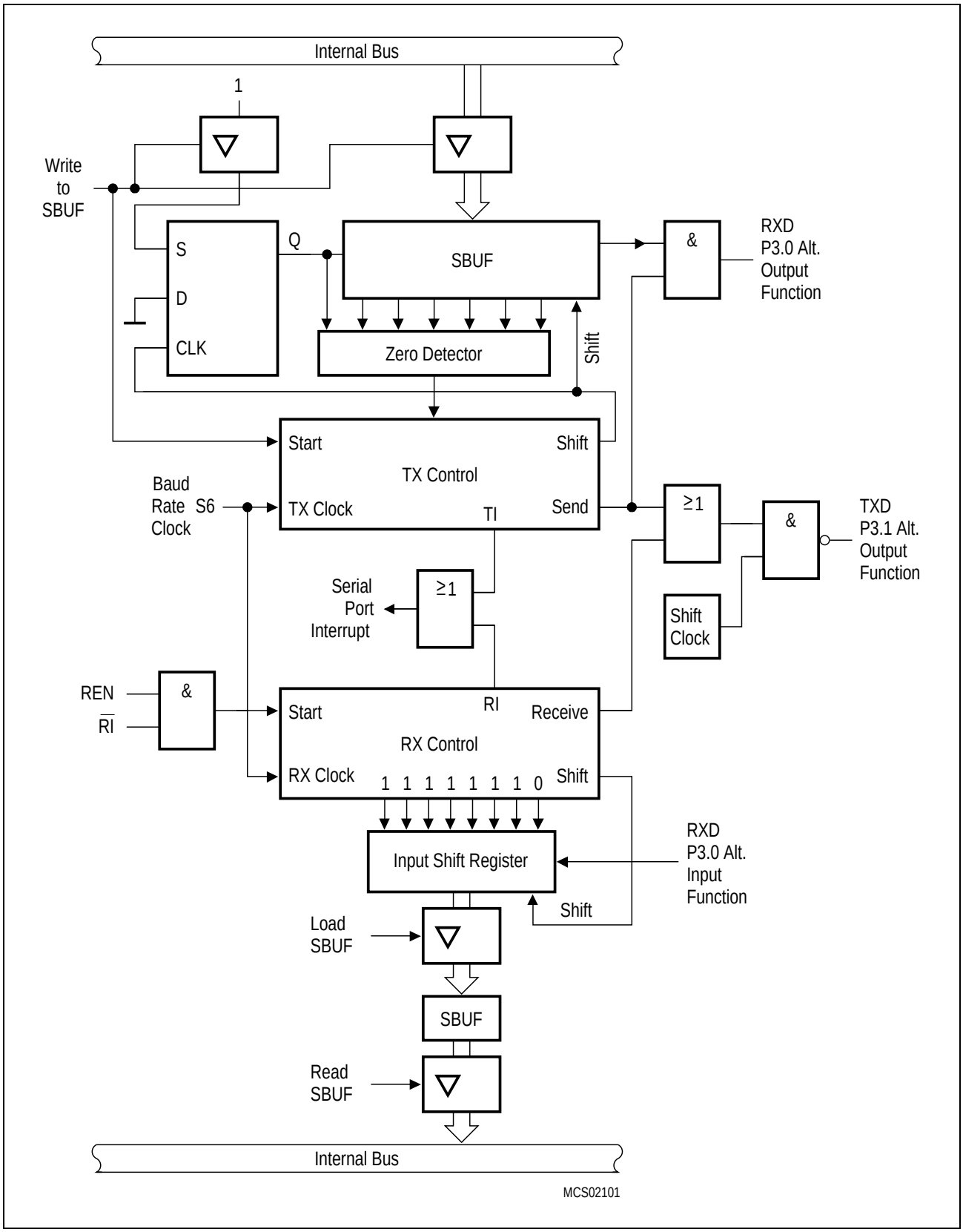


Figure 6-21
Serial Interface, Mode 0, Functional Diagram

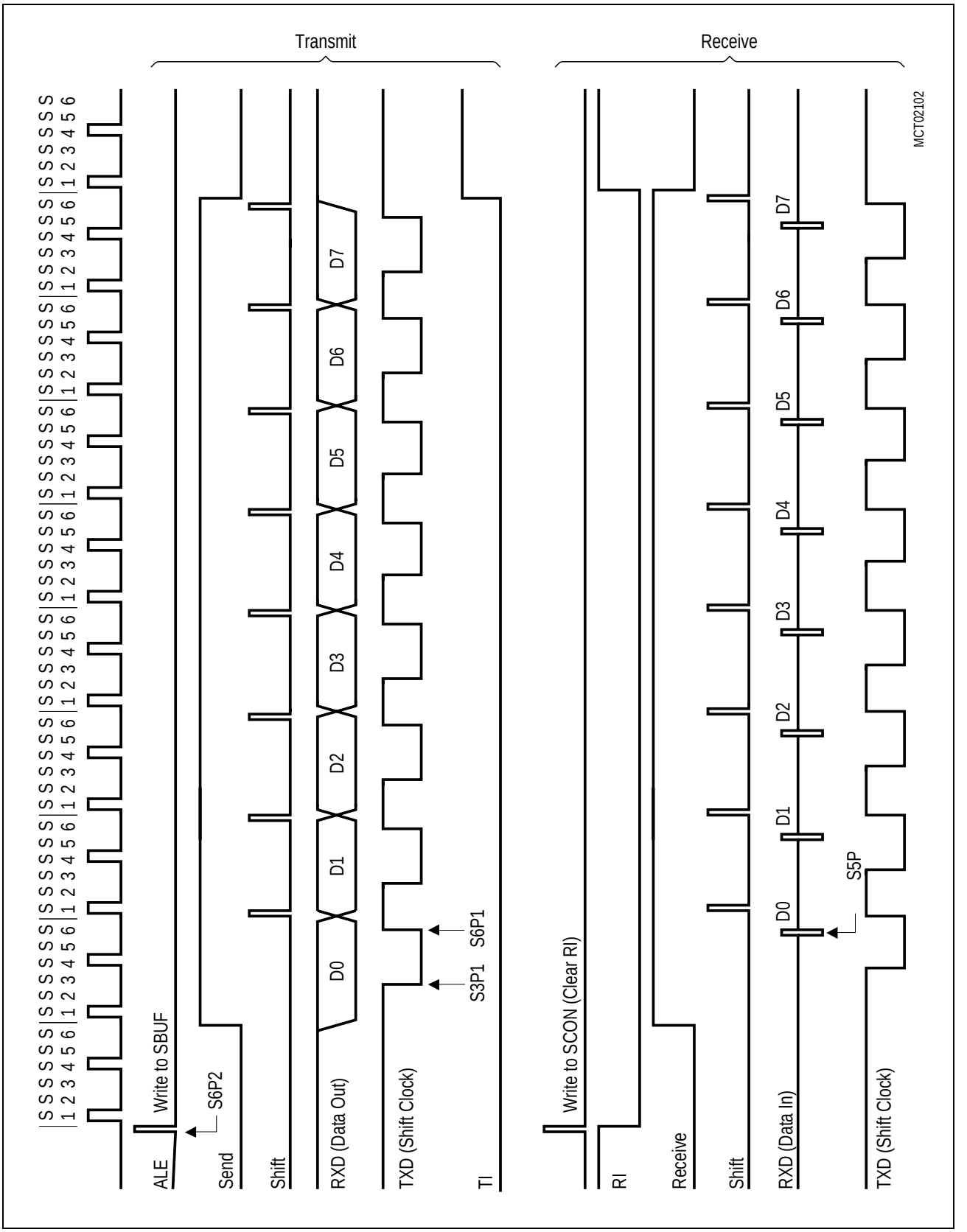


Figure 6-22
Serial Interface, Mode 0, Timing Diagram

6.3.5 Details about Mode 1

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

Figure 6-23 shows a simplified functional diagram of the serial port in Mode 1. Timing associated with transmit and receive is illustrated in **Figure 6-24**.

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 starts at 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, “0”s are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the “1” which was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain “0”s. 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 sixteen times whatever baudrate 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 sixteen 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 two of the three 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 enables rejection of 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, “1”s shift out to the left. When the start bit arrives at the leftmost position in the shift register (in Mode 1 this 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 conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the eight data bits go into SBUF, and RI is activated. At this point, whether the above conditions are met or not, the unit resumes looking for a 1-to-0 transition in RXD.

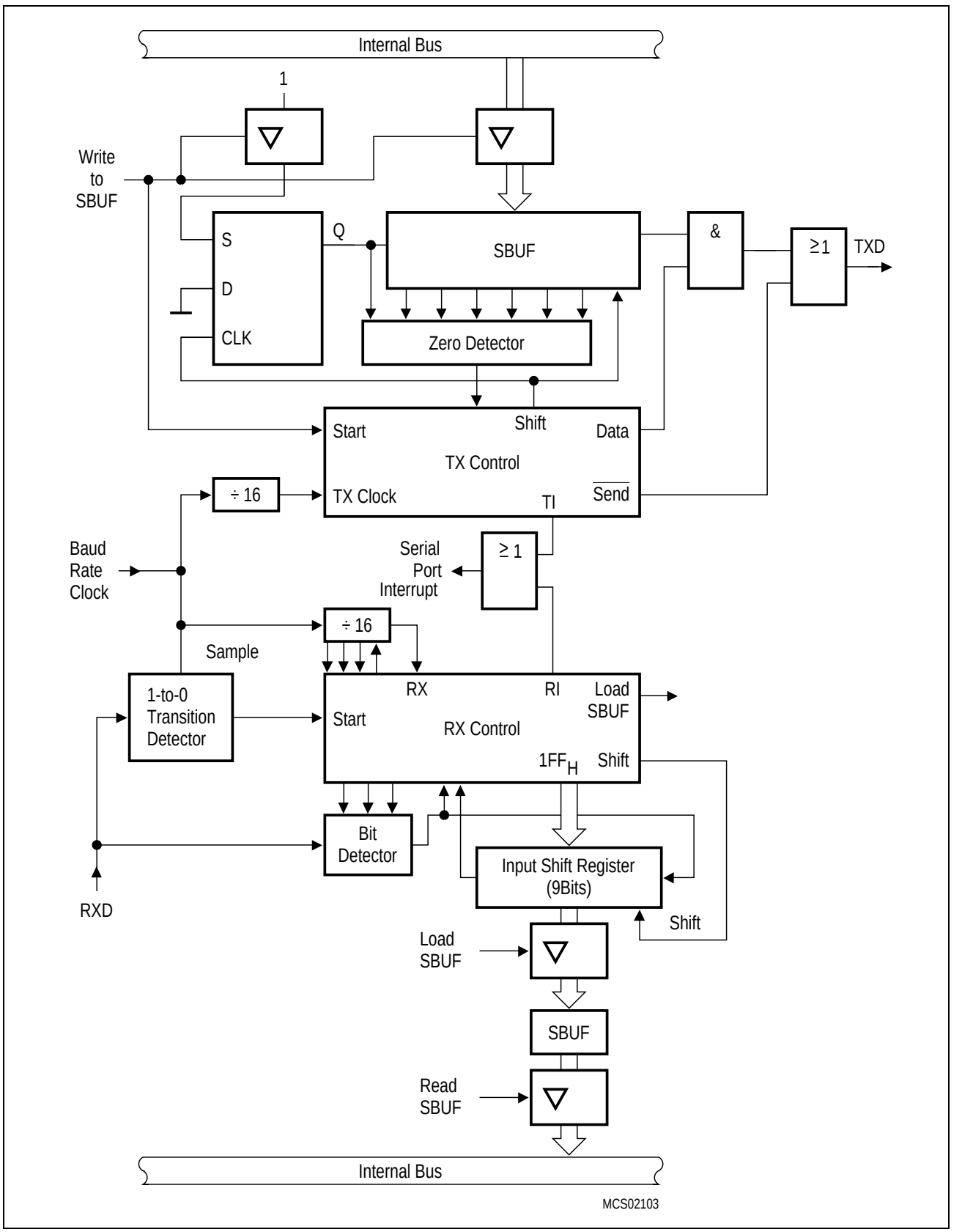


Figure 6-23
Serial Interface, Mode 1, Functional Diagram

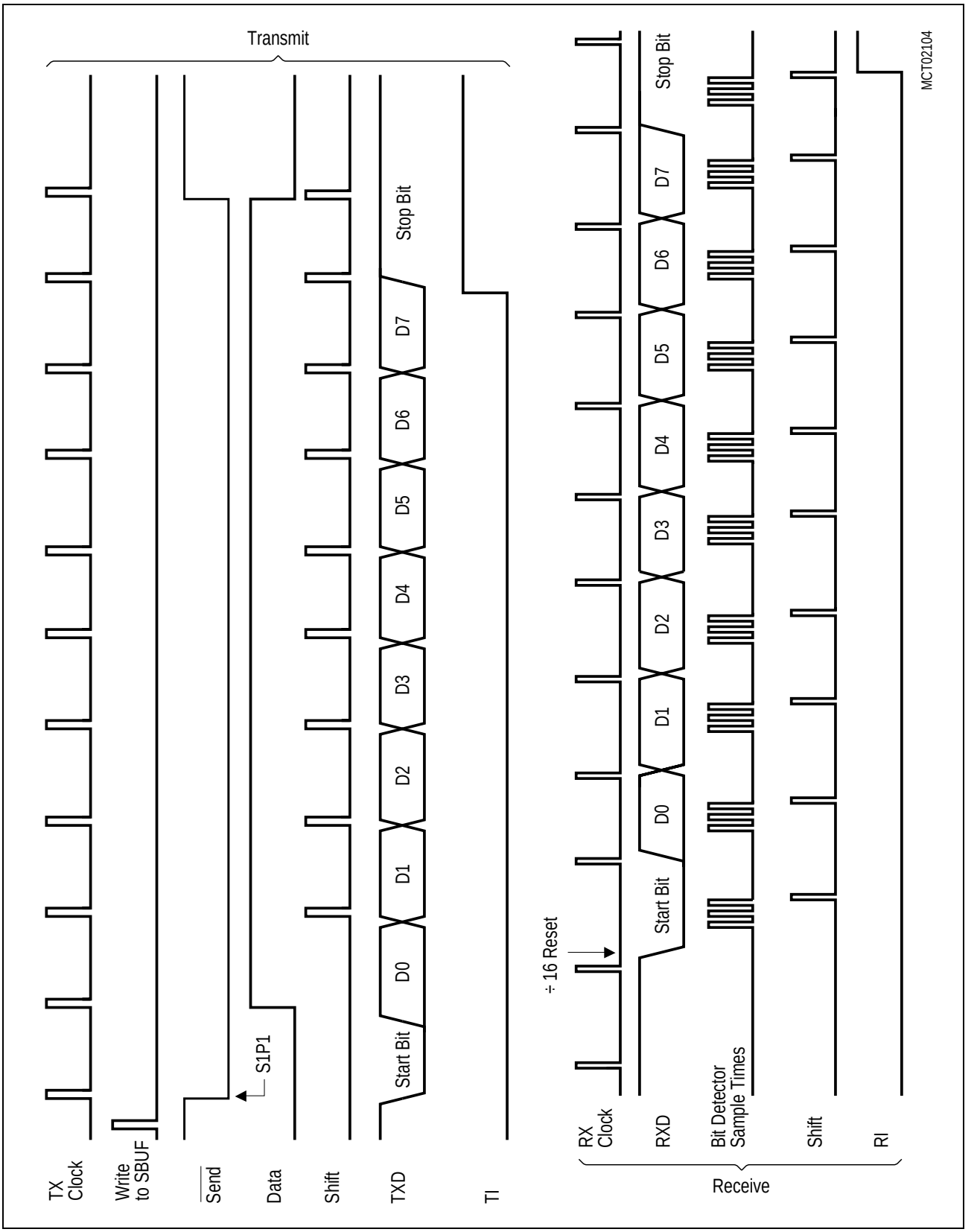


Figure 6-24
Serial Interface, Mode 1, Timing Diagram

6.3.6 Details about Modes 2 and 3

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

Figure 6-25 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 timing associated with transmit/receive is illustrated in **Figure 6-26**.

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 starts at 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).

Transmission 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 “0”s are clocked in. Thus, as data bits shift out to the right, “0”s 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 “0”s. This condition 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 sixteen times whatever baudrate 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 two of the three 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 bits come from the right, “1”s shift out to the left. When the start bit arrives at the leftmost position in the shift register (in Modes 2 and 3 this 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 eight data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit resumes 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.

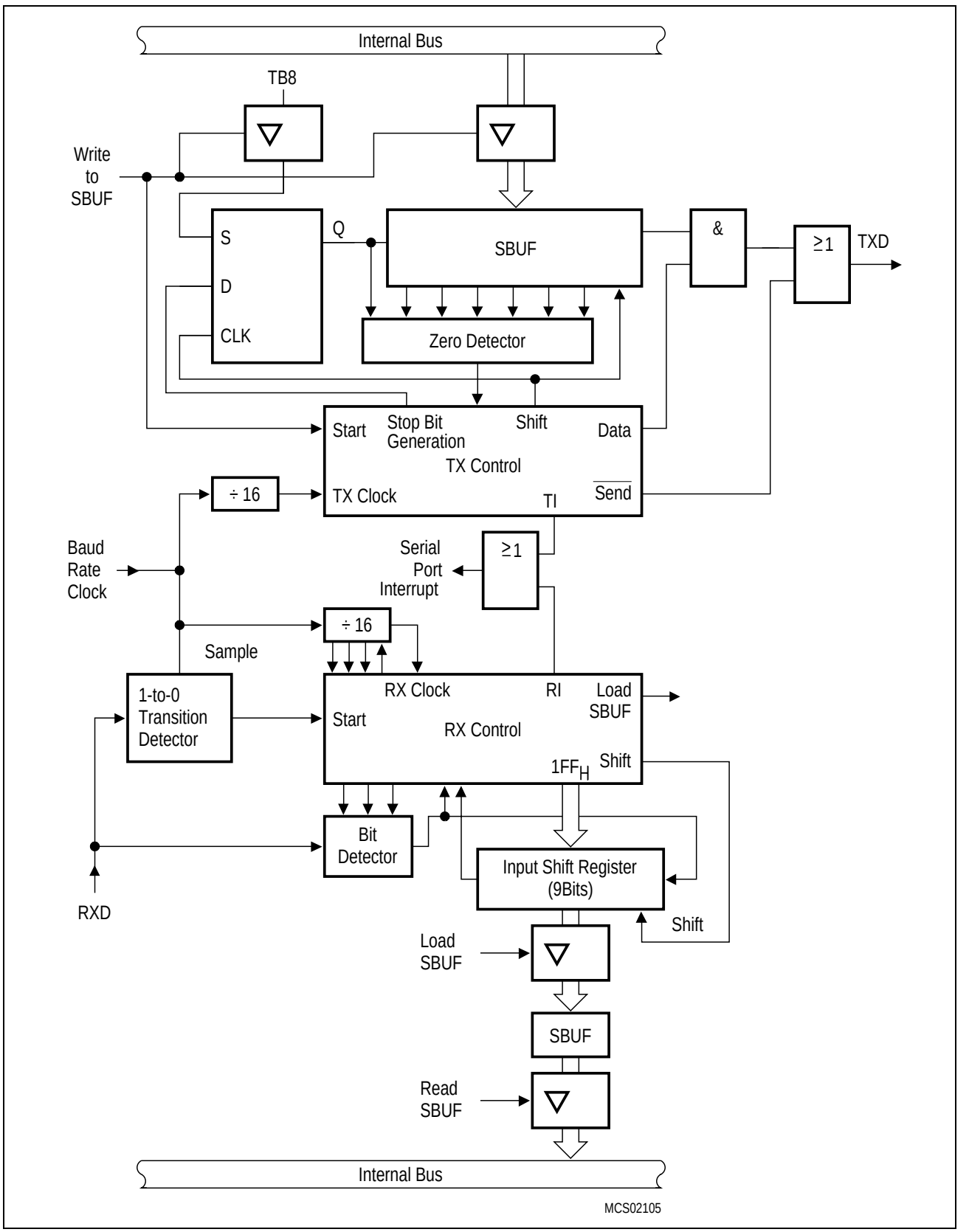


Figure 6-25
Serial Interface, Mode 2 and 3, Functional Diagram

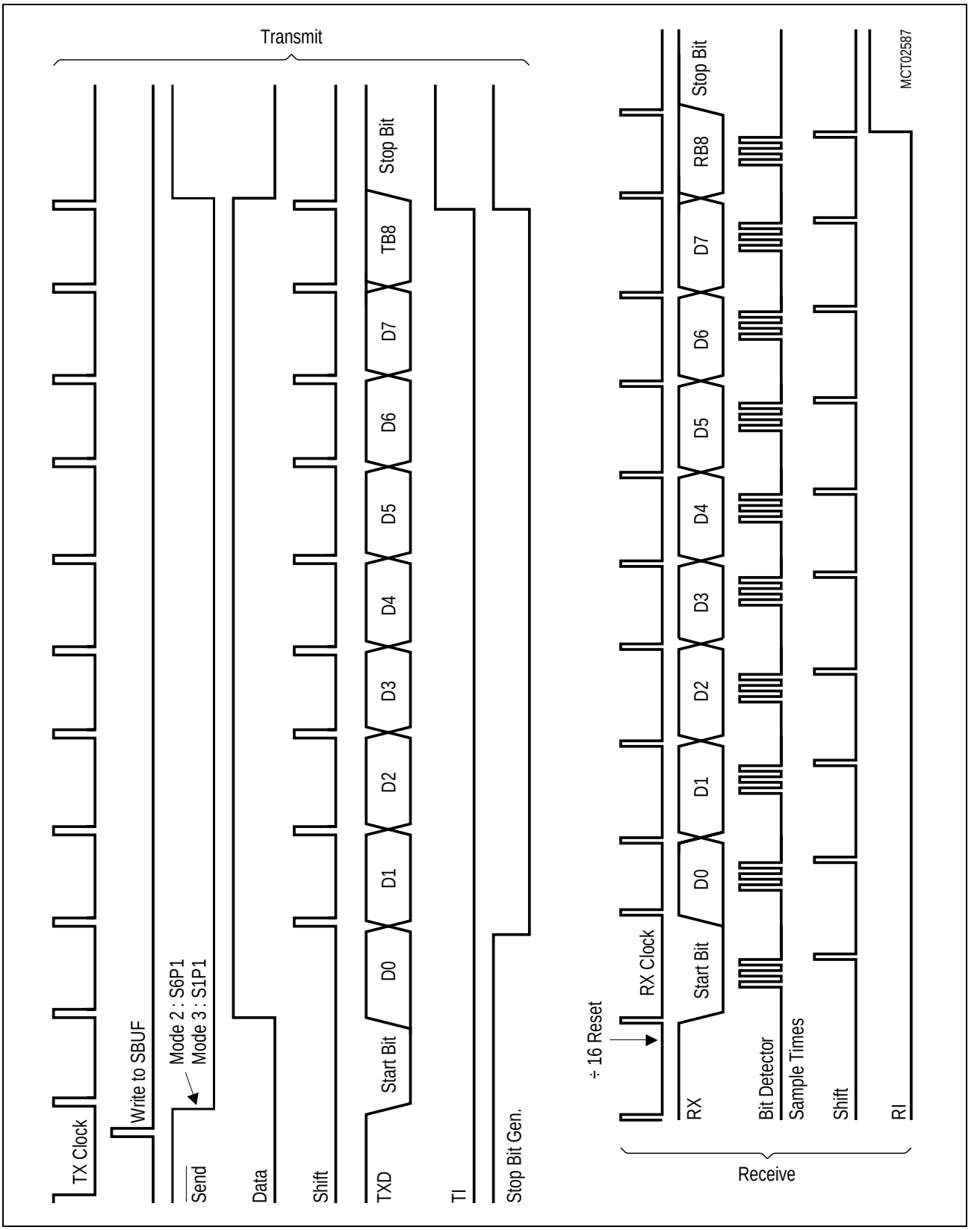


Figure 6-26
Serial Interface, Mode 2 and 3, Timing Diagram

6.4 SSC Interface

The C513AO microcontroller provides a Synchronous Serial Channel (SSC) unit. This interface is compatible with the popular SPI serial bus interface. It can be used for simple I/O expansion via shift registers, for connection with a variety of peripheral components (such as A/D converters, EEPROMs etc.), or interconnection of several microcontrollers in a master/slave structure. The SSC unit supports full-duplex or half-duplex operation and can run in Master Mode or Slave Mode.

Figure 6-27 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.

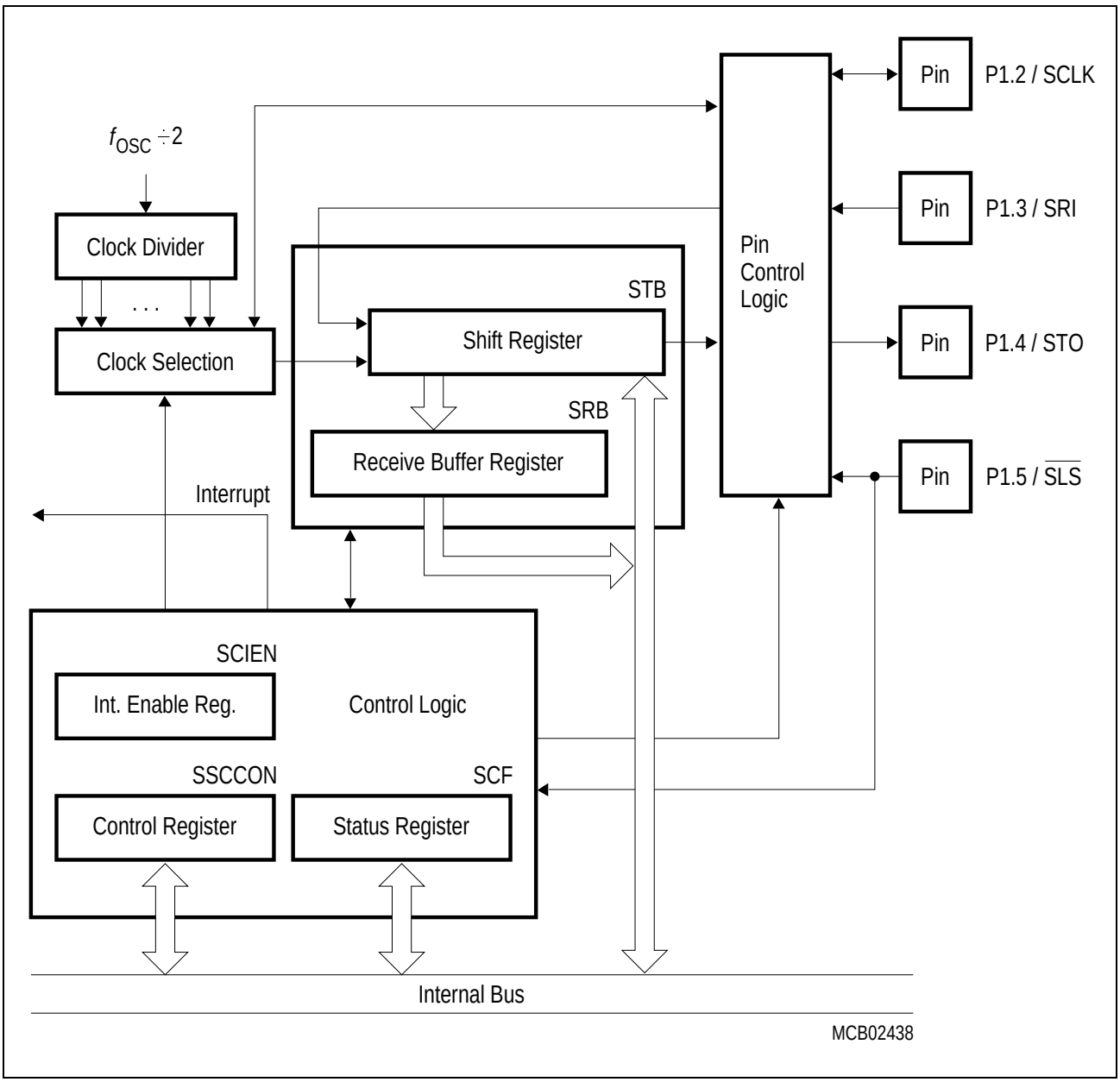


Figure 6-27
SSC Block Diagram

Because the SSC is a synchronous serial interface, a dedicated clock signal sequence must be provided for each transfer. The SSC has implemented a clock control circuit, which can generate the clock via a baudrate generator in Master Mode, or receive the transfer clock in 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 $\overline{SLS}$ enables the SSC interface and also controls the transmitter output. The pin used for this is P1.5/$\overline{SLS}$. There is an additional option to control 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.

6.4.1 General Operation of the SSC

After initialization of the SSC, the data to be transmitted must 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 6-29**).

While the transmit data in the shift register is shifted out bit-by-bit (starting with either the MSB or LSB), the incoming receive data are shifted in. The shifting of transmit and receive data is synchronized with the clock signal at pin SCLK. When the eight bits are shifted out (and an equal number are shifted in), the contents of the shift register are 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 has been 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 and CPHA bits. After eight bits have been shifted in, the contents of the shift register are transferred to the receive buffer register and the transmission complete flag TC is set. At the same time, if the transmitter is enabled in Slave Mode (TEN bit set to 1), the SSC will shift out at STO 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 contents of the shift register will not be modified.

6.4.2 Enable/Disable Control

Bit SSCEN of the SSCCON register globally enables or disables the Synchronous Serial Interface. Setting SSCEN to “0” stops the baudrate 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 for operation 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; and P1.5/$\overline{SLS}$ will remain as a regular I/O pin.

Note: The output latches of port pins which are 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 tristate until the transmitter is enabled by the $\overline{SLS}$ input being low and the TEN control bit is set to “1”. This allows more than one slave to be connected to one select line. The final selection of the slave will be done by a software protocol.

6.4.3 Baudrate Generation (Master Mode only)

The baudrate clock is generated from the processor clock ($f_{osc} \div 2$). This clock is fed into a resetable divider with seven outputs for different baudrate clocks ($f_{osc}/8$ to $f_{osc}/512$). One of these 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 such a way that the first SCLK clock transition will not occur before one half the transmit clock cycle time after the register load. This ensures that there is sufficient setup time between MSB or LSB valid on the data output and the first sample clock edge; and ensures that the MSB or LSB has the same length as the other bits (No special care is necessary in case of CPHA = 1, because the first clock edge will be used for shifting).

6.4.4 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 from the STB register.

There are different definitions for a transfer to be considered “in progress”. The definition depends on the operating mode:

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: 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 length 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

6.4.5 Master/Slave Mode Selection

Selection of the SSC unit for Master Mode or Slave Mode is dependent on the hardware configuration and must be made before the SSC will be enabled.

Normally, a specific device will operate as either master or slave. The SSC has no on-chip support for multi-master configurations (switching between Master and Slave Mode operation). The SSC can operate as a master in a multi-master environment if external circuitry is provided for swapping transmit and receive lines.

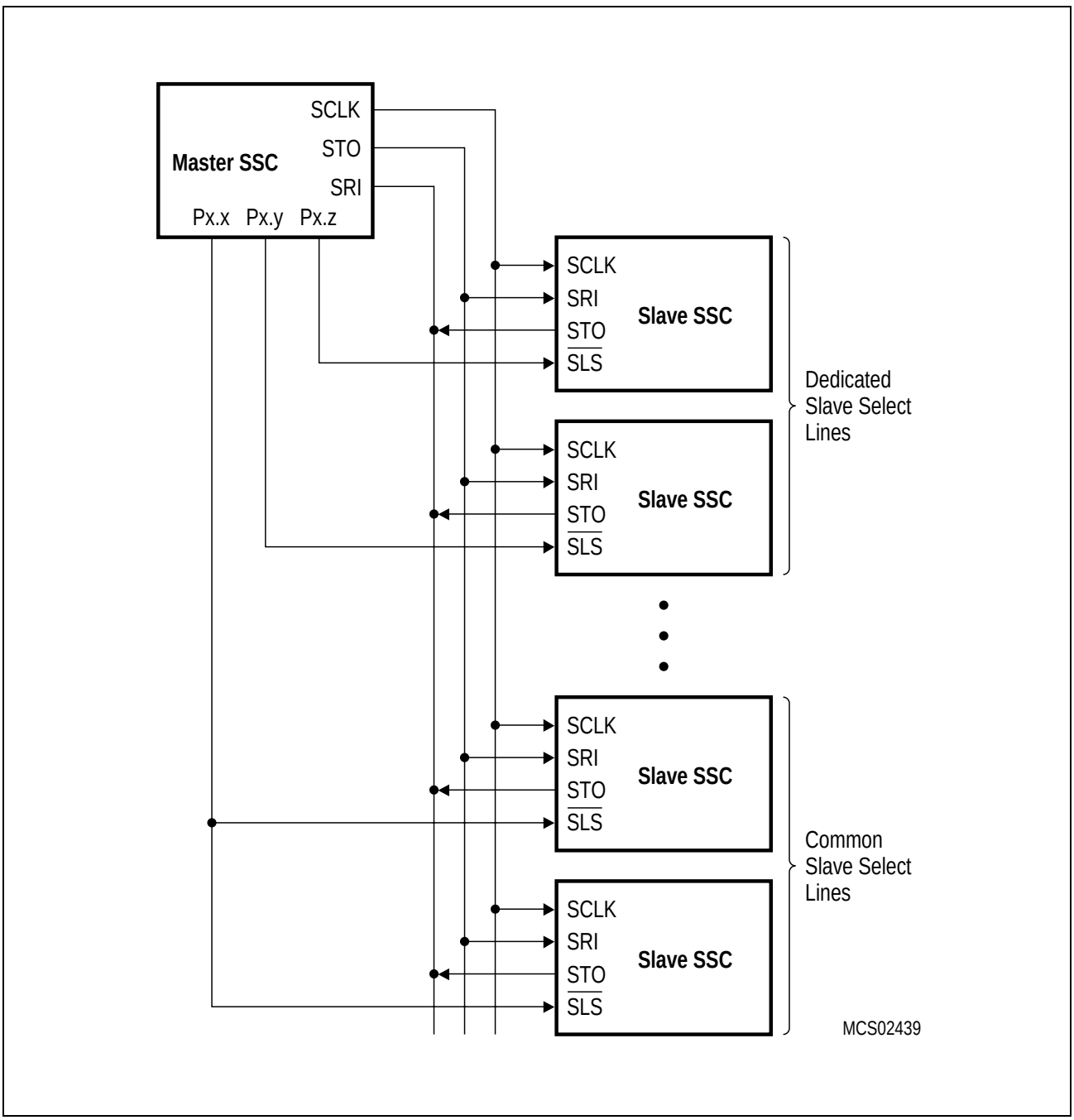


Figure 6-28
Typical SSC System Configuration

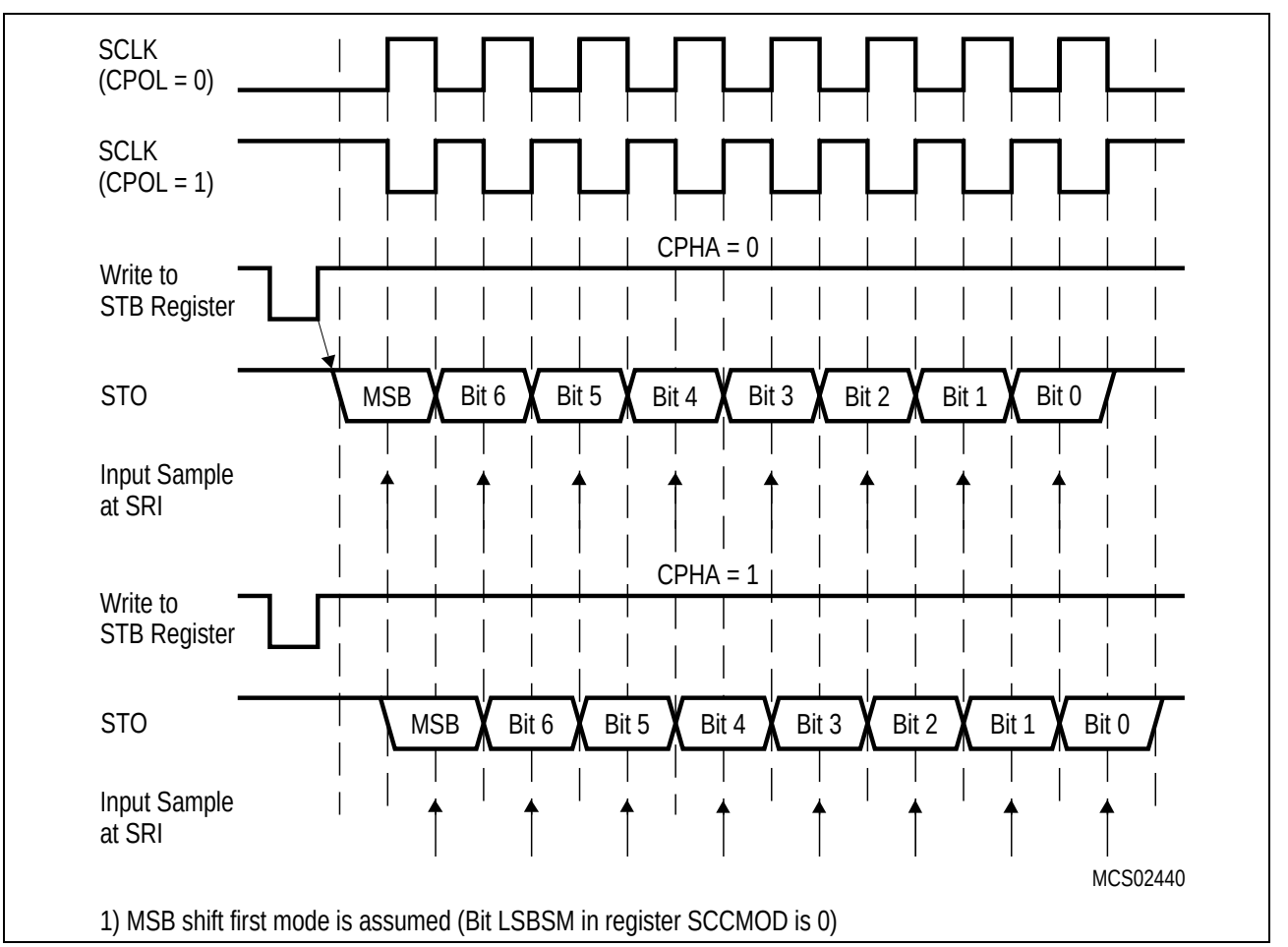
6.4.6 Data/Clock Timing Relationships

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

6.4.6.1 Master Mode Operation

Figure 6-29 shows the clock-data/control relationship of the SSC in Master Mode. When CPHA is set to “1”, the MSB (or LSB) of the data that was written into the shift register will be provided on the transmitter output after the first clock edge; and the receiver input will sample with the next clock edge. The direction (rising or falling) of the respective clock edge depends 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 (or LSB) will output immediately after the data is 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 transmissions, the data output STO will be “1”.



1) MSB shift first mode is assumed (Bit LSBSM in register SCCMOD is 0)

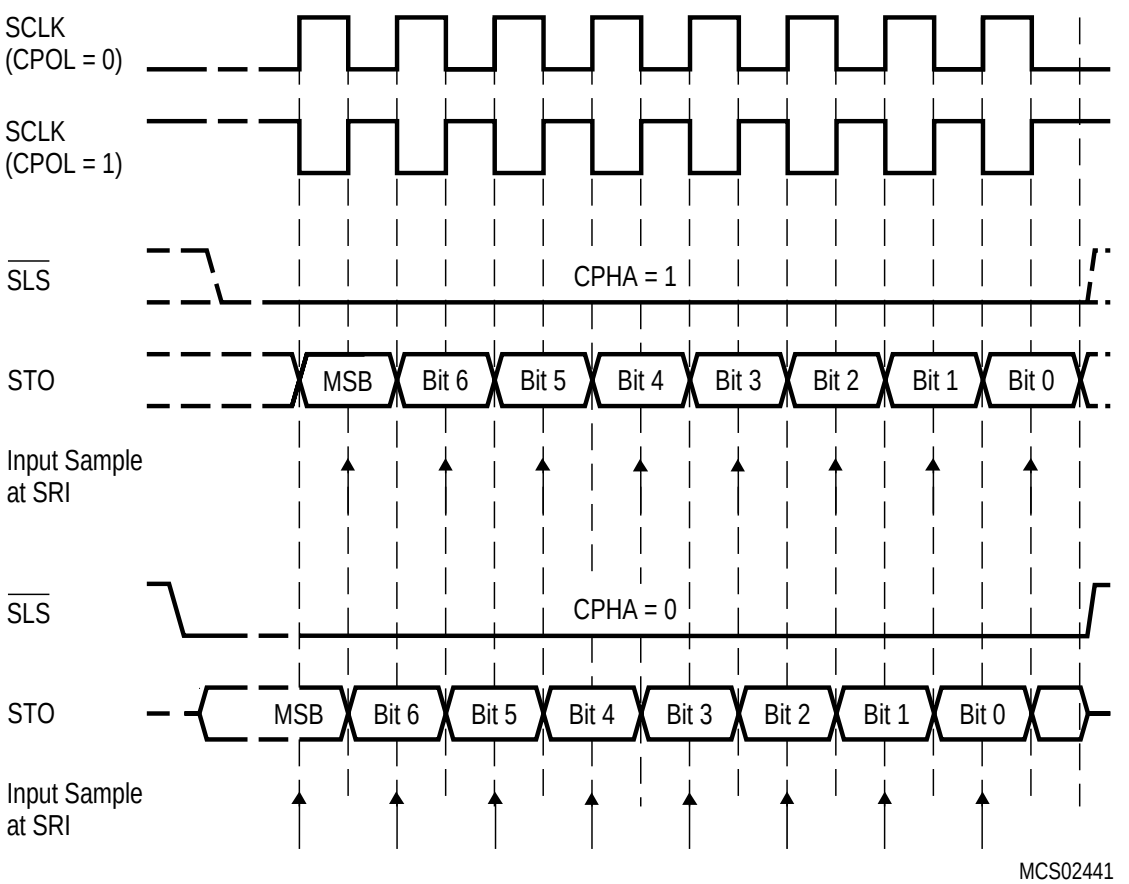
Figure 6-29
Master Mode Operation of SSC

6.4.6.2 Slave Mode Operation

Figure 6-30 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. Then, the receiver input will sample the input data with the next clock edge. The direction (rising or falling) of the respective clock edge depends 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 (or LSB) 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 impedance 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 $\overline{SLS}$ goes inactive before all eight bits are received during a transmission, 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.



1) MSB shift first mode is assumed (Bit LSBSM in register SCCMOD is 0)

Figure 6-30
Slave Mode Operation of SSC

6.4.7 Register Description

The SSC interface has six Special function Registers (SFRs) which are listed in **Table 6-5**.

Table 6-5
Special Function Registers of the COMP Unit

Symbol	Description	Address
SSCCON	SSC Control Register	$E8_H$
SCIEN	SSC Interrupt Enable Register	$F9_H$
SCF	SSC Status Register	$F8_H$
STB	SSC Transmit Buffer Register	$E9_H$
SRB	SSC Receive Buffer Register	EA_H
SSCMOD	SSC Mode Test Register	EB_H

The SSCCON Register provides basic control of the SSC functions, such as general enable/ disable, mode selections, and transmitter control.

Special Function Register 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 mode, 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 selection** MSTR = 0: Slave Mode is selected MSTR = 1: Master Mode is selected This bit must be set to the correct value (dependent 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.

Bit	Function
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.
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 from the microcontroller clock rate f_{osc}. The baudrate is defined by Baudrate = f_{osc}/Devidefactor = $f_{osc}/(4 \times 2^{BRS(2-0)})$, for BRS (2-0) ≠ 0.

BRS(2-0)	Divide Factor	Baudrate for f_{osc} = 7.68 MHz	Baudrate for f_{osc} = 12 MHz
0	reserved	reserved	reserved
1	8	960 kBaud	1.5 MBaud
2	16	480 kBaud	750 kBaud
3	32	240 kBaud	375 kBaud
4	64	120 kBaud	187,5 kBaud
5	128	60 kBaud	93,75 kBaud
6	256	30 kBaud	46,875 kBaud
7	512	15 kBaud	23,44 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.

The SCIEN Register enables or disables interrupt request for the status bits. SCIEN must be written only when the SSC interrupts are disabled in the general Interrupt Enable Register IE ($A8_H$) using bit ESSC, otherwise, unexpected interrupt requests may occur.

Special Function Register 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
–	Reserved for future use.
WCEN	**SSC 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	**SSC 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 interrupt behaviour is also affected by bit ESSC in the Interrupt Enable Register IE and by bit PSSC in the Interrupt Priority Register IP.

Special Function Register STB (Address $E9_H$) **Reset Value: XX_H**
Special Function Register SRB (Address EA_H) **Reset Value: XX_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$E9_H$	.7	.6	.5	.4	.3	.2	.1	.0	STB
EA_H	.7	.6	.5	.4	.3	.2	.1	.0	SRB

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

The register SSCMOD is used to enable test modes during factory test. It must not be written or modified during normal operation of the C513AO.

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

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

Bit	Function
LOOPB	**SSC Loopback Enable** This bit should be used for test purposes only. LOOPB = 0: The SSC operates as specified. LOOPB = 1: The STO output is connected internally via an inverter to the SRI input, allowing to check the transfer locally without a second SSC device.
TRIO	**SSC disable Tristate Mode of SSC inputs** This bit should be used for test purposes only. TRIO = 0: The SSC operates as specified. TRIO = 1: The SSC inputs will be connected to the output latch of the corresponding port pin. This allows a test of the SSC in Slave Mode by simulating a transfer via a program setting the port latches accordingly.
5-1	All bits of this register are set to “0” after reset. When writing SSCMOD, these bits must be written with “0”.
LSBSM	**SSC LSB Shift Mode** If LSBSM is cleared, the SSC will shift out the MSB of the data first and LSB last. If LSBSM is set, the SSC will shift out LSB first and MSB last.

7 Interrupt System

The C513AO provides seven interrupt sources with two priority levels. Five of the interrupts can be generated by the on-chip peripherals (Timer 0, Timer 1, Timer 2, USART, and SSC) and three of the interrupts may be triggered externally (P1.1/T2EX, P3.2/INT0, P3.3/INT1). A non-maskable eighth interrupt is reserved for external wake-up from power-down mode.

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

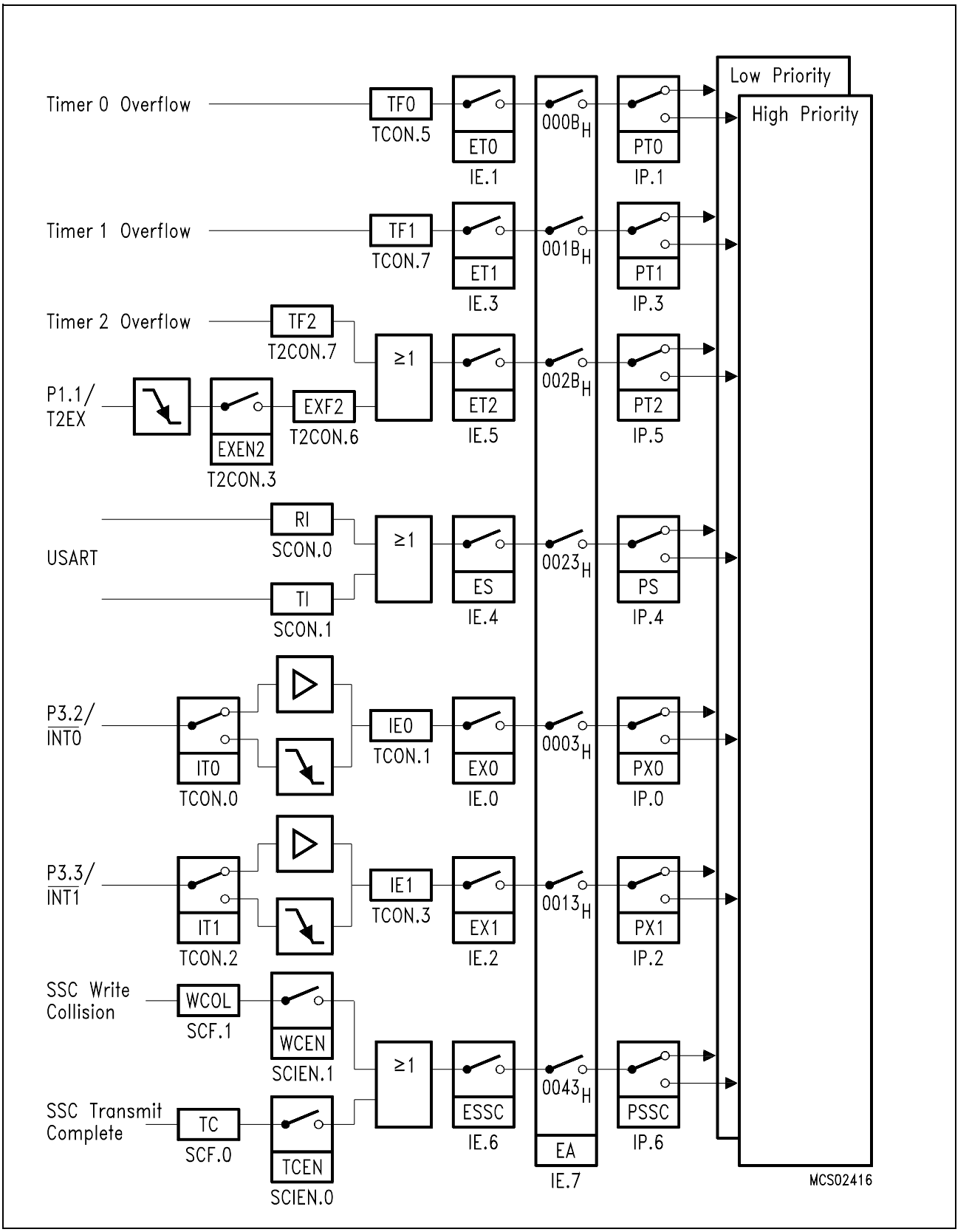


Figure 7-1
Interrupt Request Sources

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 (SFR). Examples include TCON, T2CON and, 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 “1”. But the interrupt is not necessarily serviced.

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

Table 7-1
Interrupt Sources and Vector

Interrupt Source	Vector Address	Request Flags
External interrupt 0	0003_H	IE0
Timer 0 interrupt	$000B_H$	TF0
External interrupt 1	0013_H	IE1
Timer 1 interrupt	$001B_H$	TF1
USART serial port interrupt	0023_H	RI + TI
Timer 2 interrupt	$002B_H$	TF2 + EXF2
Synchronous Serial Channel interrupt (SSC)	0043_H	WCOL + TC
Wake-up from power-down mode	$007B_H$	–

7.2 Interrupt Registers

7.2.1 Interrupt Enable Registers

Each interrupt vector can be enabled or disabled individually 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.

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	**Enable/Disable 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 (SSC) is disabled.
ET2	**Timer 2 Interrupt Enable.** If ET2 = 0, the Timer 2 interrupt is disabled.
ES	**USART Serial Channel Interrupt Enable.** 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 Request Flags

The request flags for the different interrupt sources are located in several special function registers. This section describes the locations and meanings of these interrupt request flags in detail.

External Interrupts 0 and 1 (P3.2/$\overline{\text{INT0}}$ and P3.3/$\overline{\text{INT1}}$) each can be either level-activated or negative transition-activated, depending on bits IT0 and IT1 in SFR TCON. The flags which generate these interrupts are bits IE0 and IE1 in SFR TCON. When an external interrupt is generated, the flag that of this interrupt is cleared by hardware when the service routine is vectored to; but, only if the interrupt was transition-activated. If the interrupt was level-activated, the requesting external source directly controls the request flag, rather than the on-chip hardware.

Timer 0 and Timer 1 interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective timer/counter registers (all except Timer 0 in Mode 3). When a timer interrupt is generated, the flag which generated it is cleared by the on-chip hardware when the service routine is vectored to.

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

	MSB							LSB	
Bit No.	$8F_H$	$8E_H$	$8D_H$	$8C_H$	$8B_H$	$8A_H$	89_H	88_H	
88_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

The shaded bits are not used for interrupt request control.

Bit	Function
TF1	**Timer 1 overflow flag** Set by hardware on Timer/Counter 1 overflow. Cleared by hardware when the processor vectors to the interrupt routine.
TF0	**Timer 0 overflow flag** Set by hardware on Timer/Counter 0 overflow. Cleared by hardware when the processor vectors to the interrupt routine.
IE1	**External interrupt 1 request flag** Set by hardware. Cleared by hardware when the processor vectors to the interrupt routine (if IT1 = 1) or released by external source (if IT1 = 0).
IT1	**External interrupt 1 level/edge-trigger control flag** If IT1 = 0, level triggered external Interrupt 1 is selected. If IT1 = 1, negative edge-triggered external Interrupt 1 is selected.
IE0	**External interrupt 0 request flag** Set by hardware. Cleared by hardware when the processor vectors to the interrupt routine (if IT0 = 1) or released by external source (if IT0 = 0).
IT0	**External interrupt 0 level/edge trigger control flag** If IT0 = 0, level triggered external interrupt 0 is selected. If IT0 = 1, negative edge-triggered external Interrupt 0 is selected.

Interrupt of the serial interface is generated by the request flags RI and TI in SFR SCON. The two request flags of the serial interface are logically OR-ed together. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine of each interface will normally need to determine whether it was the receive interrupt flag or the transmission interrupt flag which generated the interrupt, and the bit will need to be cleared by software.

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

	MSB							LSB	
Bit No.	$9F_H$	$9E_H$	$9D_H$	$9C_H$	$9B_H$	$9A_H$	99_H	98_H	
98_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

The shaded bits are not used for interrupt request control.

Bit	Function
TI	**Serial port Transmitter Interrupt flag** TI is set by hardware at the end of the eighth bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. TI must be cleared by software.
RI	**Serial port Receiver Interrupt flag** RI is set by hardware at the end of the eighth bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (exception see SM2). RI must be cleared by software.

Timer 2 interrupt is generated by the logical OR of bit TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may need to determine whether it was TF2 or EXF2 which generated the interrupt, and the bit will need to be cleared by software.

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

	MSB							LSB	
Bit No.	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	T2CON

The shaded bits are not used for interrupt request control.

Bit	Function
TF2	**Timer 2 Overflow Flag**. Must be cleared by software. Set by a Timer 2 overflow. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	**Timer 2 External Flag**. Must be cleared by software. 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 does not cause an interrupt in up/down counter mode (DCEN = 1, SFR T2MOD).

SSC Interrupt is generated by a logical OR of flag WCOL and TC in SFR SCF. Both bits can be cleared by software when a “0” is written to the bit location. WCOL is reset by hardware when the SSC transmit data register STB is written with data after a proceeding read operation of the SCF register. TC is reset by hardware when the receive data register SRB is read the next time after a proceeding read operation of the SCF register. The interrupt service routine will normally need to determine whether it was the WCOL or the TC flag which generated the interrupt, and the bit will need to be cleared by software.

Special Function Register SCF (Address $F8_H$) **Reset Value: $XXXX\ XX00_B$**

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

SCF

Bit	Function
-	Reserved bits for future use. Read by CPU returns undefined values
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 erroneous write attempt. This bit can be reset in two different ways: Writing a “0” to the bit (bit access, byte access or read-modify-write access) 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: Writing a “0” to the bit (bit access, byte access or read-modify-write access) after the receive buffer register SRB has been read. 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.

7.2.3 Interrupt Priority Registers

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 Register 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
–	Reserved for future use.
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	**USART Serial Channel Interrupt Enable** 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 levels 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 7-2**.

Table 7-2
Interrupt Source Structure

Interrupt Source		Priority
External Interrupt 0	IE0	High
Synchronous Serial Channel	WCOL OR TC	
Timer 0 Interrupt	TFO	
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 can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

7.3 Interrupt Handling

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 proceeding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, if 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 IE0/IE1 or IP0/IP1.

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 at least one more instruction will be executed before any interrupt is vectored to, if the instruction in progress is RETI or any write access to registers IE or IP. 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 which 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 was 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**.

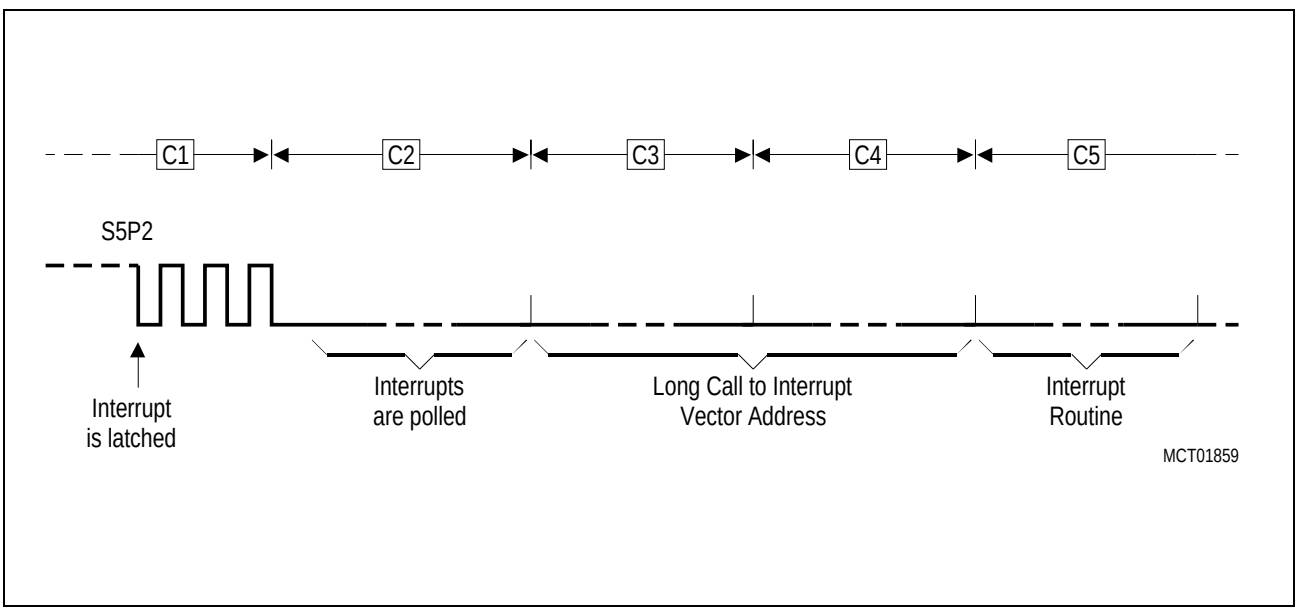


Figure 7-2
Interrupt Response Timing Diagram

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 rules described above, 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 has generated the interrupt. In other cases where it does not, the flag must be cleared 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 to.

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, the interrupt control system would have behaved as if an interrupt were 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, 1 can be programmed to be level-activated or 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 must hold the request active until the requested interrupt is generated. Then, it must 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.

Because the external interrupt pins are sampled once in each machine cycle, an input low should be held for at least twelve oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source must hold the request pin high for at least one cycle, and then hold it low for at least one cycle. This ensures that the transition is recognized, so the corresponding interrupt request flag will be set (see **Figure 7-3**). The external interrupt request flags will be cleared automatically by the CPU when the service routine is called.

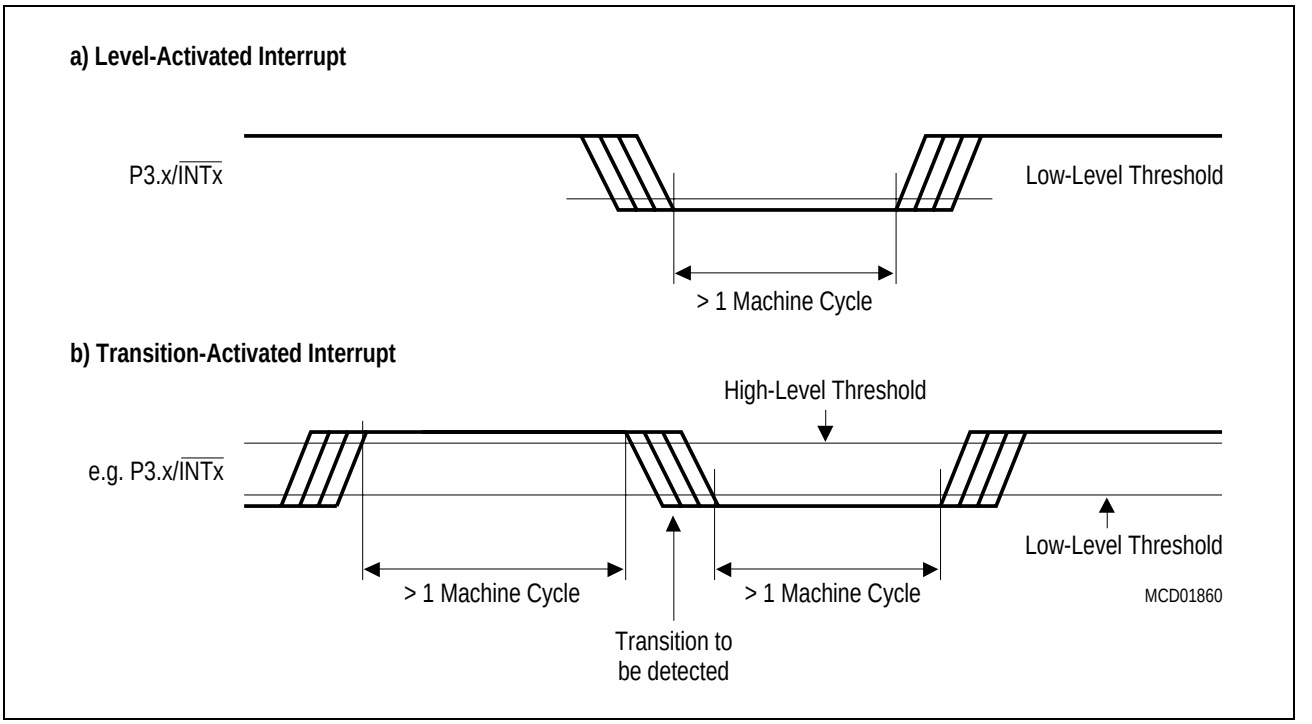


Figure 7-3
External Interrupt Detection

7.5 Interrupt 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 the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles will elapse between activation of the external interrupt request and the execution of the first instruction of the service routine.

A longer response time would occur if the request is blocked by one of the three conditions previously listed. If an interrupt of equal or higher priority is already in progress, the additional wait time is determined by 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 three cycles since the longest instructions (MUL and DIV) are only four cycles long. If the instruction in progress is RETI or a write access to the registers IEN or IP, the additional wait time cannot be more than five cycles (a maximum of one more cycle to complete the instruction in progress, plus four cycles to complete the next instruction, if the instruction is MUL or DIV). Thus, in a single interrupt system, the response time is always more than three cycles and fewer than nine cycles.

8 Fail Safe Mechanisms

The C513AO offers enhanced fail-safe mechanisms which allow automatic recovery from a software upset or a hardware failure:

- A programmable Watchdog Timer (WDT) has variable time-out period from 512 µs up to approx. 1.1 s at 12 MHz
- An Oscillator Watchdog (OWD) monitors the on-chip oscillator and forces the microcontroller into reset state if the on-chip oscillator fails. It also provides the clock for a fast internal reset after power-on.

8.1 Programmable Watchdog Timer

To protect the system against software failure, the user's program must clear the watchdog within a previously programmed time period. If the software fails to do this periodic refresh of the Watchdog Timer, an internal hardware reset will be initiated. The software can be designed such that the watchdog times out if the program does not work properly. It also times out if a software error is based on a hardware-related problem.

The Watchdog Timer in the C513AO is a 15-bit timer which is incremented by a count rate of either $f_{CYCLE}/2$ or $f_{CYCLE}/32$ ($f_{CYCLE} = f_{OSC}/12$). That is, the machine clock is divided by a fixed divide-by-two prescaler and an optional divide-by-16 prescaler arranged in series. The divide-by-16 prescaler is enabled by setting bit WDTPSEL (bit 7 of SFR WDTREL). From the 15-bit Watchdog Timer count value, only the upper seven bits can be programmed.

Figure 8-1 shows the block diagram of the programmable Watchdog Timer.

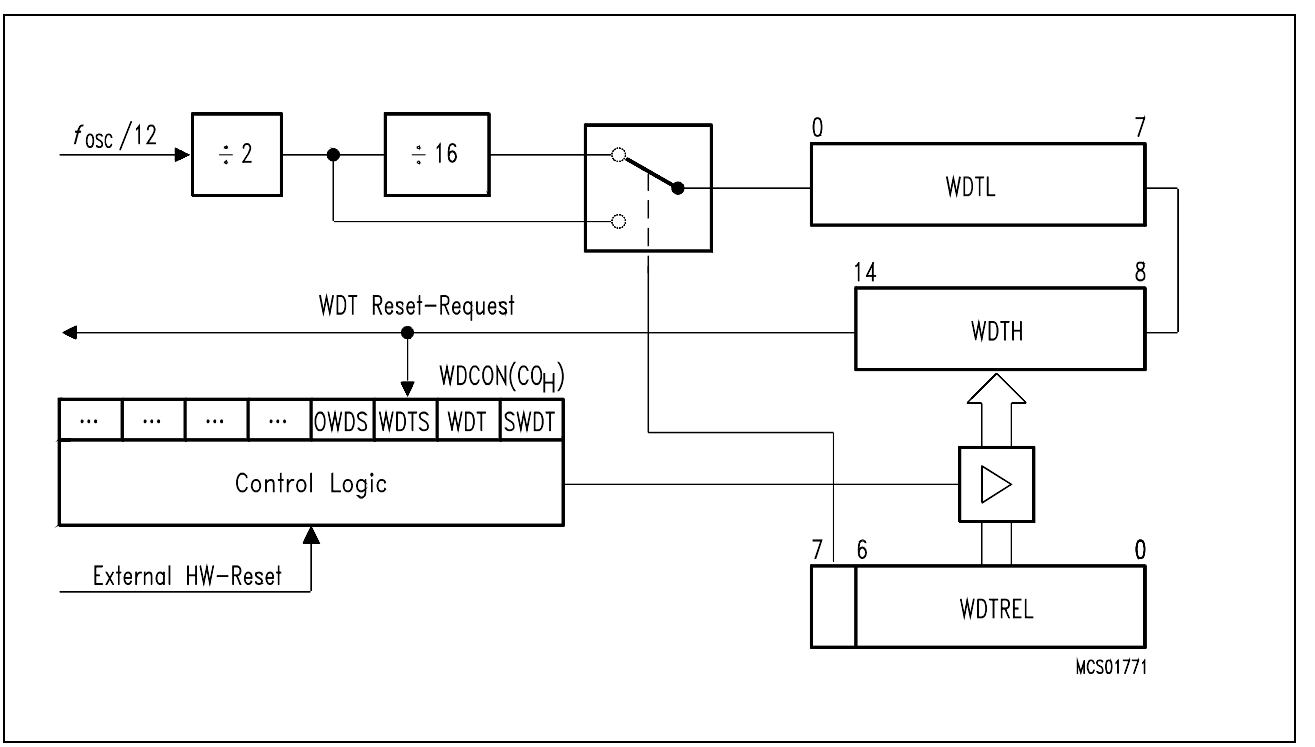
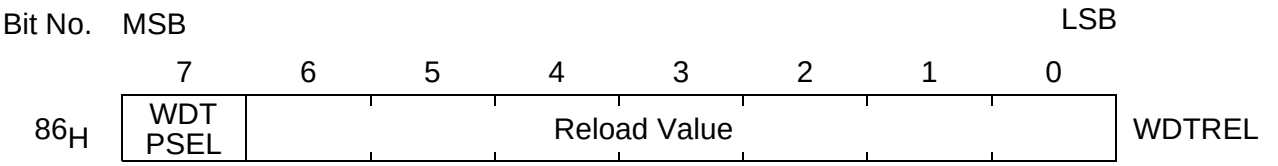


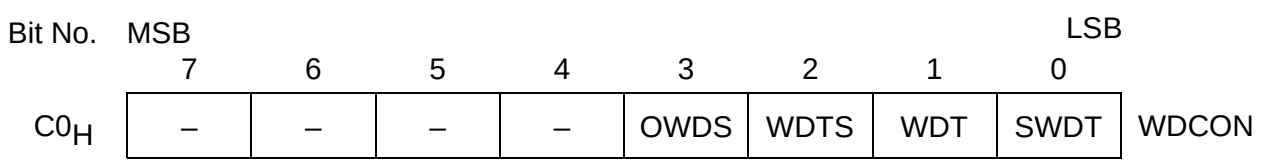
Figure 8-1
Block Diagram of the Programmable Watchdog Timer

Special Function Register WDTREL (Address 86_H) **Reset Value: 00_H**



Bit	Function
WDTPSEL	**Watchdog Timer Prescaler Select Bit**. When set, the Watchdog Timer is clocked through an additional divide-by-16 prescaler.
WDTREL.6 - 0	**7-bit Reload Value** This is used for the high-byte of the Watchdog Timer. This value is loaded to WDTH when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

Special Function Register WDCON (Address $C0_H$) **Reset Value: $XXXX\ 0000_B$**



Bit	Function
–	Not implemented. Reserved for future use.
OWDS	**Oscillator Watchdog Timer Status Flag.** Set by hardware when an oscillator watchdog reset occurs. Can be set and cleared by software.
WDTS	**Watchdog Timer Status Flag.** Set by hardware when a Watchdog Timer reset occurs. Can be cleared and set by software.
WDT	**Watchdog Timer Refresh Flag.** Set to initiate a refresh of the Watchdog Timer. Must be set directly before SWDT is set to prevent an unintentional refresh of the Watchdog Timer.
SWDT	**Watchdog Timer Start Flag.** Set to activate the Watchdog Timer. When directly set after setting WDT, a Watchdog Timer refresh is performed.

Immediately after start, the Watchdog Timer is initialized to the reload value programmed to WDTREL.0-WDTREL.6. Register WDTREL is cleared to 00_H after an external hardware reset, an Oscillator Watchdog power on reset, or a Watchdog Timer reset. The lower seven bits of WDTREL can be loaded by software at any time.

Examples (given for 12- and 16-MHz external oscillator frequency):

Table 8-1
Watchdog Timer Time-Out Periods

WDTREL	Time-Out Period		Comments
	f_{osc} = 12 MHz	f_{osc} = 16 MHz	
00_H	65.535 ms	49.152 ms	This is the default value
80_H	1.1 s	0.79 s	Maximum time period
$7F_H$	512 µs	384 µs	Minimum time period

Starting the Watchdog Timer

The Watchdog Timer can be started by software (bit SWDT in SFR WDCON); but, it cannot be stopped during active mode of the device. If the software fails to clear the Watchdog Timer, an internal reset will be initiated. The reset cause (either external reset or reset caused by the watchdog) can be examined by software (status flag WDTS in WDCON is set). A refresh of the Watchdog Timer is done by setting bits WDT (SFR WDCON) and SWDT consecutively. This double instruction sequence has been implemented to increase system security.

It must be noted, however, that the Watchdog Timer is halted during Idle Mode and Power-down Mode of the processor (see Chapter “Power Saving Modes”). Therefore, even the Watchdog Timer cannot reset the device when one of the power-saving modes has been entered accidentally.

8.1.1 Refreshing the Watchdog Timer

At the same time the Watchdog Timer is started, the 7-bit register WDTH is preset by the contents of WDTREL.0 to WDTREL.6. Once started, the Watchdog Timer cannot be stopped by software. However, it can be refreshed to the reload value only by first setting bit WDT (WDCON) and by the next instruction setting SWDT (WDCON). Bit WDT will be cleared automatically during the third machine cycle after having been set. This double-instruction refresh of the Watchdog Timer is implemented to minimize the chance of an unintentional reset of the watchdog unit.

When the Watchdog Timer is started or refreshed, its lower eight bits, stored in WDTL are reset to 00_H (see **Figure 8-1**).

The reload register, WDTREL, can be written at any time. Therefore, a periodic refresh of WDTREL can be included in the starting procedure of the Watchdog Timer. Thus, a wrong reload value caused by a possible distortion during the write operation to WDTREL can be corrected by software.

8.1.2 Watchdog Reset and Watchdog Status Flag (WDTS)

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state $7FFC_H$. The duration of the reset signal then depends on the prescaler selection (either 8 or 128 cycles). Unlike an external reset, in an internal reset the Watchdog Timer is not disabled and bit WDTS is set. The WDTS is a flip-flop which is set by a Watchdog Timer reset and can be cleared by an external hardware reset. Bit WDTS allows the software to identify the source from which the reset was activated. The bit WDTS can also be cleared by software.

8.2 Oscillator Watchdog Unit

The Oscillator Watchdog (OWD) unit is used for three functions:

- **Monitoring the on-chip oscillator's function**
  The watchdog supervises the on-chip oscillator's frequency. If the frequency is lower than the frequency of the auxiliary RC oscillator in the watchdog unit, the internal clock is supplied by the RC oscillator and the device is brought into reset. If the failure condition disappears (that is, if the on-chip oscillator has a higher frequency than the RC oscillator), the device executes a final reset phase of typically 1 ms to allow the oscillator to stabilize. Then, the oscillator watchdog reset is released and the device resumes program execution.
- **Fast internal reset after power-on**
  The oscillator watchdog unit provides a clock supply for reset before the on-chip oscillator has started. The oscillator watchdog unit reset works identically to the monitoring function.
- **Control of external wake-up from software power-down mode**
  When power-down mode is terminated by a low level at the $\overline{\text{INT0}}$ pin, the oscillator watchdog unit ensures that the microcontroller resumes operation (execution of the power-down wake-up interrupt) with the nominal clock rate. In power-down mode, the RC oscillator and the on-chip oscillator are stopped. Both oscillators are started again when power-down mode is terminated. When the on-chip oscillator has a frequency higher than the RC oscillator, the microcontroller starts operation after a final delay of typ. 1 ms to allow the on-chip oscillator to stabilize.

Note: The Oscillator Watchdog unit is always enabled.

8.2.1 Detailed Description of the Oscillator Watchdog Unit

Figure 8-2 shows the block diagram of the Oscillator Watchdog unit. It consists of an internal RC oscillator which provides the reference frequency for comparison with the frequency of the on-chip oscillator.

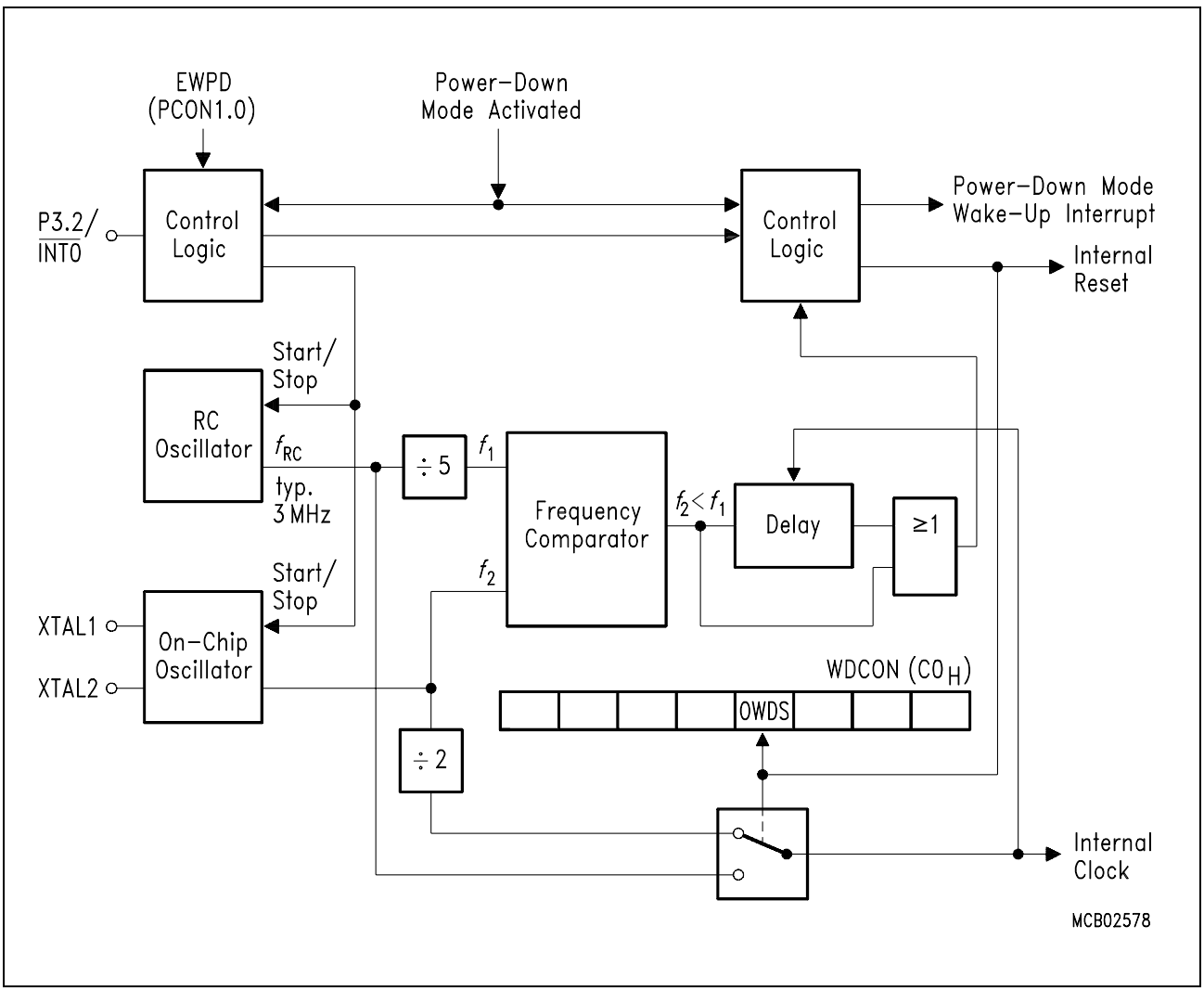


Figure 8-2
Functional Block Diagram of the Oscillator Watchdog

The frequency coming from the RC oscillator is divided by five and compared to the on-chip oscillator's frequency. If the frequency coming from the on-chip oscillator is lower than that derived from the RC oscillator, the watchdog detects a failure condition (oscillation at the on-chip oscillator could stop because of crystal damage etc.). In this case, it switches the input of the internal clock system to the output of the RC oscillator. This means that the device is being clocked even if the on-chip oscillator has stopped or has not yet started. At the same time, the watchdog activates the internal reset to bring the device into its defined reset state. The reset is performed because a clock is available from the RC oscillator. This internal watchdog reset has the same effect as an externally applied reset signal with the following exceptions: The Watchdog Timer Status flag WDTS is not reset (however, the Watchdog Timer is stopped); and bit OWDS is set. This allows the software to examine error conditions detected by the Watchdog Timer if an oscillator failure occurs.

The Oscillator Watchdog is able to detect a recovery of the on-chip oscillator after a failure. If the frequency derived from the on-chip oscillator is again higher than the reference, the watchdog starts a final reset sequence which takes typ. 1 ms. Within that time, the clock is still supplied by the RC oscillator and the device is held in reset. This allows a reliable stabilization of the on chip oscillator. After that, the watchdog switches the clock supply back to the on-chip oscillator and releases the reset request. If no external reset is applied in this moment, the part will start program execution. If an external reset is active, however, the device will keep the reset state until the external reset request is released.

Furthermore, the status flag OWDS is set if the Oscillator Watchdog reset was active. The status flag can be evaluated by software to detect a reset caused by the Oscillator Watchdog. The flag OWDS can be set or cleared by software. An external reset request, however, also resets OWDS (and WDTS).

If software power-down mode is activated, the RC oscillator and the on-chip oscillator are stopped. Both oscillators are started again in power-down mode when a low level is detected at the $\overline{\text{INT0}}$ input pin and when bit EWPD in SFR PCON1 is set (wake-up from power-down mode enabled). After start-up of the watchdog circuitry in power-down mode, a power-down mode wake-up interrupt is generated (instead of an internal reset).

8.2.2 Fast Internal Reset after Power-On

The C513AO can use the Oscillator Watchdog for a fast internal reset procedure after power-on. Normally, members of the 8051 family (for example, SAB 80C52) enter their default reset state not before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed to bring the device into the correct reset state. Especially if a crystal is used, the start up time of the oscillator is relatively long (typ. 10 ms). During this time period, the pins have an undefined state which could have severe effects (for example, to actuators connected to port pins).

In the C513AO, the Oscillator Watchdog unit avoids this situation. After power-on, the Oscillator Watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 µs). The watchdog circuitry detects a failure condition for the on-chip oscillator because it has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is valid, the watchdog uses the RC oscillator output as clock source for the chip. This allows correct resetting of the device and brings all ports into the defined state. The maximum delay time between power-on and correct reset state is 34 µs.

9 Power Saving Modes

The C513AO microcontroller provides three basic power-saving modes: Idle Mode, Slow-down Mode, and Power-down Mode.

The functions of the power-saving modes are controlled by bits located in the Special Function Register PCON. PCON is located at SFR address 87_H. PCON1 is located in the mapped SFR area at address 88_H and is accessed with RMAP = 1. Bit RMAP is located in SFR SYSCON ($B1_H$) bit 4.

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. Slow-down Mode is controlled by bit SD, located in SFR PCON.

Furthermore, register PCON contains two general purpose flags. For example, the flag bits GF0 and GF1 can be used to indicate that an interrupt occurred during normal operation or during an Idle Mode. Then, an instruction which activates Idle Mode can also set one or both flag bits. When Idle Mode is terminated by an interrupt, the interrupt service routine can examine the flag bits.

Special Function Register PCON (Address 87_H) **Reset Value: $0XX00000_B$**

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

The function of the shaded bit is not described in this section.

Symbol	Function
–	Not implemented. Reserved for future use.
SD	**Slow-down Mode Bit** When set, the Slow-down Mode is enabled (default is Slow-down disabled).
GF1	**General purpose flag**
GF0	**General purpose flag**
PDE	**Power-down Enable Bit** When set, power-down mode is entered.
IDLE	**Idle Mode Enable Bit** When set, idle mode is entered.

Special Function Register PCON1 (Mapped Address $88_{H)}$ **Reset Value: $0XXXXXXX_B$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
88_H	EWPD	–	–	–	–	–	–	–	PCON1

Symbol	Function
–	Reserved for future use.
EWPD	**External Wake-up From Power-down Enable Bit** Setting EWPD before entering Power-down Mode enables external wake-up from Power-down Mode capability via the pin $\overline{INT0}$.

9.1 Idle Mode

In Idle Mode, the oscillator of the C513AO continues to run; but, the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the Synchronous Serial Channel (SSC) interface, and all timers are still provided with the clock. 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 SSC and serial interfaces are not running, maximum power reduction can be achieved. This state is also the test condition for Idle Mode I_{DD}.

The user must take care which peripheral should continue to run and which are to be stopped during Idle Mode. The state of all port pins – both the pins controlled by their latches and those controlled by their secondary functions – also 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 alternative functions, they still continue to output during Idle Mode if the assigned function is on. This applies to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ are held at logic high levels.

Table 9-1
Status of External Pins During Idle Mode 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	Float	Float
PORT 1	Data/alternative outputs	Data/last output	Data/alternative outputs	Data/last output
PORT 2	Data	Data	Address	Data
PORT 3	Data/alternative outputs	Data/last output	Data/alternative 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.

As discussed below, 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.

Idle Mode is entered by setting the IDLE (PCON.0) bit. Since PCON is not a bit-addressable register, the setting of this bit is achieved by byte-handling instructions. For example:

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

The instruction which sets bit IDLE is the last instruction executed before going into Idle Mode.

There are two ways to terminate Idle Mode:

- Idle Mode can be terminated by activating any enabled interrupt. This interrupt will be serviced and, normally, 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 Idle Mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Note: The Watchdog Timer is the only peripheral which is stopped automatically during Idle Mode.

9.2 Slow-down Mode Operation

In some applications, where power consumption and dissipation are critical, the controller might run for a certain time at reduced speed (for example, if the controller is waiting for an input signal). As a CMOS device, the C513AO has an almost linear dependence of the operating frequency and the power supply current, so that a reduction of the operating frequency results in reduced power consumption.

In Slow-down Mode, all signal frequencies derived from the oscillator clock are divided by 8.

Slow-down Mode is activated by setting the bit SD in SFR PCON. If Slow-down Mode is enabled, the clock signals for the CPU and the peripheral units are reduced to 1/8th of the nominal system clock rate. The controller enters Slow-down Mode after a short synchronization period (max. two machine cycles). Slow-down Mode is terminated by clearing bit SD.

Slow-down Mode can be combined with Idle Mode by performing the following instruction:

```
ORL     PCON,#00010001B         ; entering idle mode combined with the slow down mode:
                                ; (IDLE and SD set)
```

There are two ways to terminate the combined Idle and Slow-down Mode:

- Idle Mode can be terminated by activation of any enabled interrupt. When CPU operation is resumed, the interrupt will be serviced, and the next instruction to be executed after the RETI instruction will be the one following the instruction that had set the bits IDLE and SD. Nevertheless, Slow-down Mode remains enabled and if required must be terminated by clearing the bit SD in the corresponding interrupt service routine or at any point in the program where the user no longer requires the Slow-down Mode power-saving benefit.
- The other way to terminate the combined Idle and Slow-down Mode is with a hardware reset. Since the oscillator is still running, the hardware reset must be held active for only two machine cycles for a complete reset.

9.3 Power-down Mode

In Power-down Mode, the on-chip oscillator which operates with the XTAL pins is stopped. Therefore, all functions of the microcontroller are stopped and only the contents of the on-chip RAM, XRAM, and the SFRs are maintained. The port pins, which are controlled by their port latches, output the values that are held by their SFRs. 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 $\overline{\text{PSEN}}$ are held at logic low level (see **Table 9-1**).

During Power-down Mode operation, V_{DD} can be reduced to minimize power consumption. It must be ensured, however, that V_{DD} is not reduced before Power-down Mode is invoked, and that V_{DD} is restored to its normal operating level before Power-down Mode is terminated.

9.3.1 Invoking Power-Down Mode

Power-down Mode is entered by setting bit PDE (PCON.1). Since PCON is not a bit-addressable register, setting the PDE bit is achieved by byte-handling instructions. For example:

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

The instruction which sets bit PDE is the last instruction executed before going into Power-down Mode.

9.3.2 Exit from Power-down Mode

The C513AO can recover from Power-down Mode in one of the following ways:

- Hardware reset
- Wake-up from power-down through pin P3.2/$\overline{\text{INT0}}$

If the bit EWPD in SFR PCON is “0” during power-down entry, the only way to exit from Power-down Mode is a hardware reset. This reset will restore the SFRs with their default values, but will not change the contents of the internal RAM and XRAM. The reset signal which terminates Power-down Mode also restarts the RC oscillator and the on-chip oscillator. The reset operation should not be activated before V_{DD} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

There is also the capability to wake-up from power-down. If this capability is used, its function must be enabled using the following instruction sequence prior to entering Power-down Mode.

```
ORL     SYSCON,#00010000B   ;set RMAP
ORL     PCON1,#80H          ;enable external wake-up from power-down by setting EWPD
ANL     SYSCON,#11101111B   ;reset RMAP (for future SFR accesses)
```

Note: Before entering Power-down Mode, the port latch of SFR P3.2 (P3.2/$\overline{\text{INT0}}$ pin) should contain a “1” (pin operates as input). Otherwise, the wake-up sequence will be started immediately when Power-down Mode is entered.

Figure 9-1 shows the procedure which must be executed when Power-down Mode is exited via the $\overline{\text{INT0}}$ wake-up capability of the C513AO.

If Power-down Mode wake-up capability has been enabled (bit EWPD in SFR PCON1 set) prior to entering Power-down Mode, the Power-down Mode can be exited via P3.2/$\overline{\text{INT0}}$ while executing the following procedure:

1. In Power-down Mode, pin P3.2/$\overline{\text{INT0}}$ must be held at high level.
2. Power-down Mode is terminated when P3.2/$\overline{\text{INT0}}$ goes low. With P3.2/$\overline{\text{INT0}}$ = low, the internal RC oscillator is started. The state of P3.2/$\overline{\text{INT0}}$ is then latched by the RC oscillator clock signal. Therefore, P3.2/$\overline{\text{INT0}}$ should be held low for at least 10 µs (latch phase). After this delay, P3.2/$\overline{\text{INT0}}$ can be set to high level again if required. Thereafter, the Oscillator Watchdog unit controls the wake-up procedure in its start up phase.
3. The Oscillator Watchdog unit starts operation as described in **Section 8.2.1**. When the on-chip oscillator clock is detected for stable nominal frequency, the microcontroller waits for a delay (typically 5 ms) and then starts again with its operation initiating the power-down wake-up interrupt. The interrupt address of the first instruction to be executed after wake-up is $007B_H$.
4. After the RETI instruction of the power-down wake-up interrupt routine has been executed, the instruction which follows the one which initiated Power-down Mode will be executed. The peripheral unit’s timer 0/1/2, serial interface, SSC interface, and WDT are frozen until end of Phase 4.

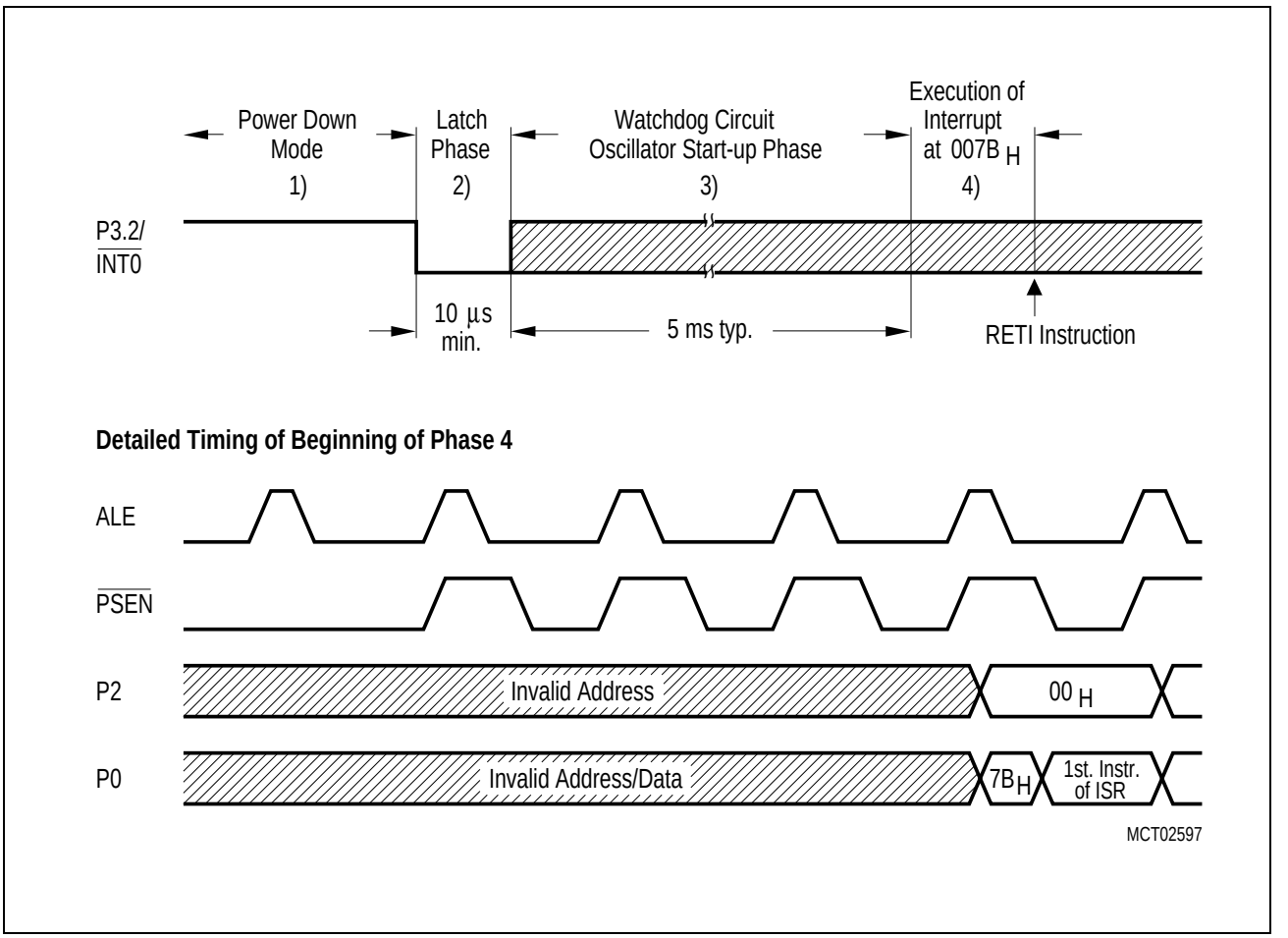


Figure 9-1
Wake-up from Power-Down Mode Procedure

All interrupts of the C513AO are disabled from Phase 2 until the end of Phase 4. Other interrupts can be handled after the RETI instruction of the wake-up interrupt routine.

Note: To avoid any unintentional external interrupt request, the user should ensure that P3.2/$\overline{INT0}$ is set back to high level after a wake-up request, prior to completion of the wake-up sequence.

10 OTP Memory Operation

The C513AO-2E is the One-Time Programmable (OTP) version of the C513AO microcontroller with a 16K byte OTP program memory. Fast programming cycles can be achieved with the C513AO-2E (1 byte in 100 µs). Several levels of OTP memory protection can be selected as well. The basic functionality of the C513AO-2E is identical to the C513AO-2R (ROM part) or the C513AO-L (ROMless part). Therefore, the programmable C513AO-2E typically can be used for prototype system design as a replacement for the ROM-based C513AO-2R microcontroller.

10.1 Programming Configuration

During normal program execution, the C513AO-2E behaves like the C513AO-2R/C513AO-L. To program the device, the C513AO-2E must be put into Programming Mode. This typically is done not in-system, but in special programming hardware. In Programming Mode, the C513AO-2E operates as a slave device similar to an EPROM stand-alone memory device and must be controlled with address/data information, control lines, and an external 11.5 V programming voltage.

In Programming Mode, Port 0 provides the bi-directional data lines and Port 2 is used for the multiplexed address inputs. The upper address information at Port 2 is latched with the signal PALE. For basic programming mode selection, the inputs RESET, $\overline{\text{PSEN}}$, $\overline{\text{EA}}/V_{PP}$, ALE and $\overline{\text{PSEL}}$ are used. Further, the inputs PMSEL1,0 are required in Programming Mode to select the access types (such as program/verify data, write lock bits, etc.). In Programming Mode, V_{DD}/V_{SS} and a clock signal at the XTAL pins must be applied to the C513AO-2E. The 11.5 V external programming voltage is input through the $\overline{\text{EA}}/V_{PP}$ pin.

Figure 10-1 shows the signals of the C513AO-2E which are required for controlling the OTP device's Programming Mode.

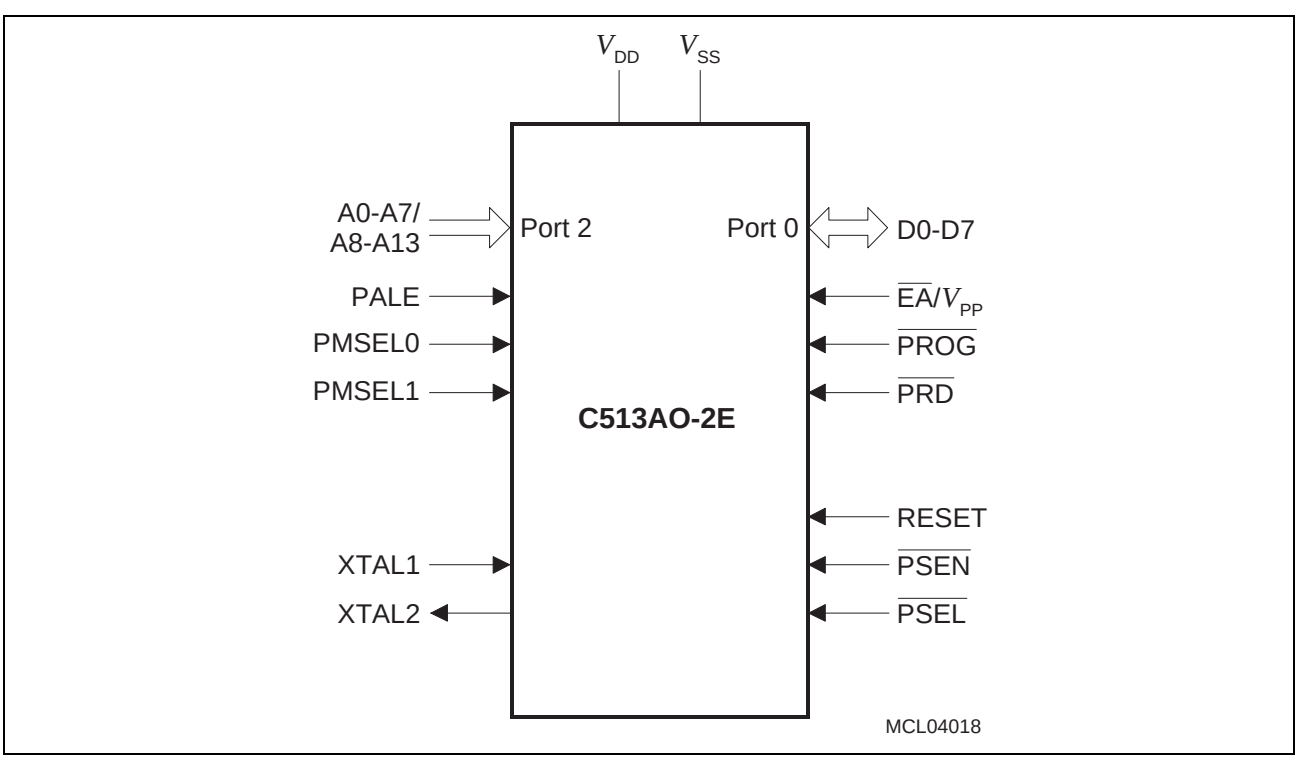


Figure 10-1
Programming Mode Configuration

10.2 Pin Configuration

Figure 10-2 to **Figure 10-4** show the detailed pin configurations of the C513AO-2E in different packages in Programming Mode.

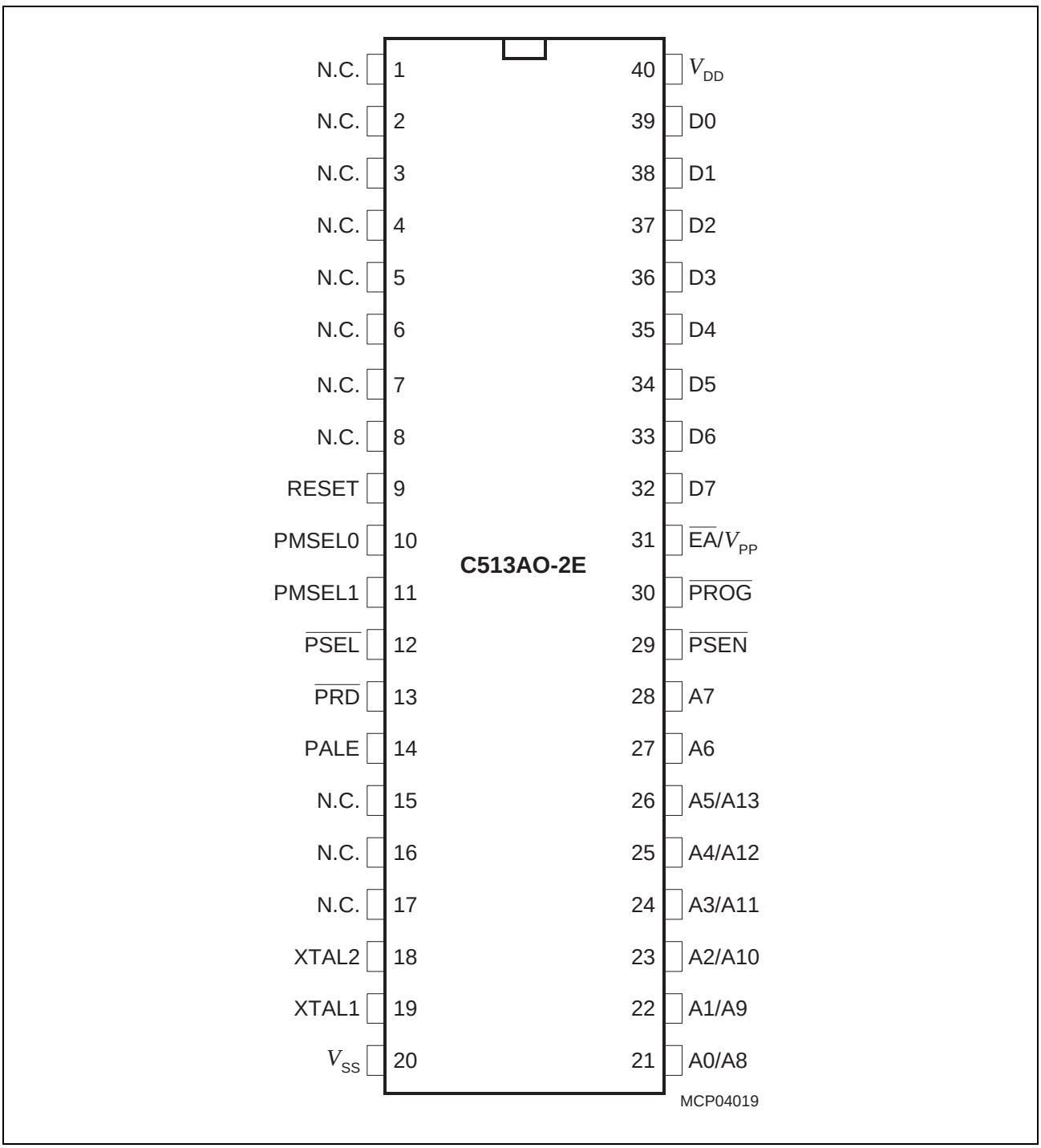


Figure 10-2
OTP Programming Mode Pin Configuration: P-DIP-40 (top view)

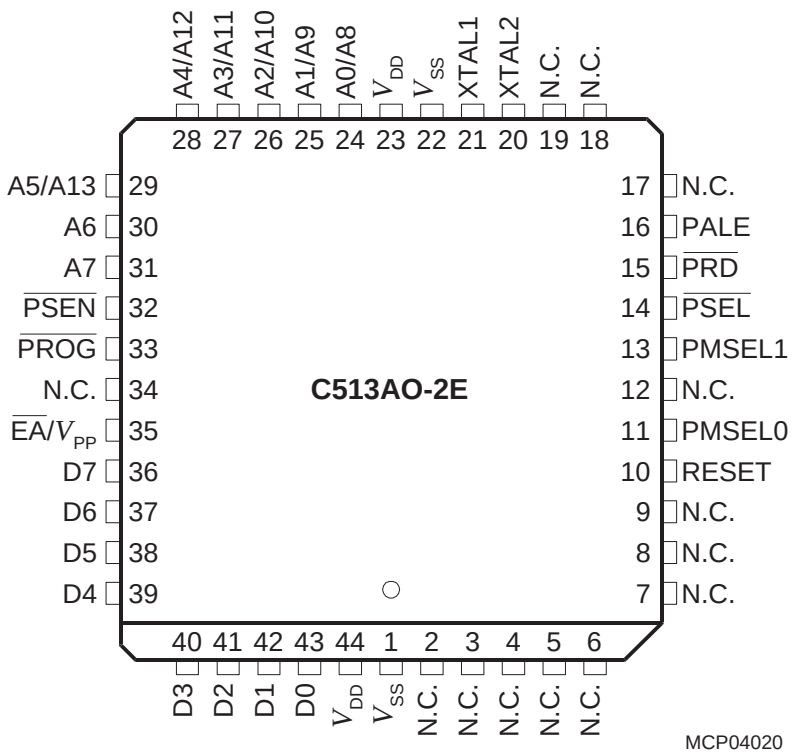


Figure 10-3
OTP Programming Mode Pin Configuration: P-LCC-44 (top view)

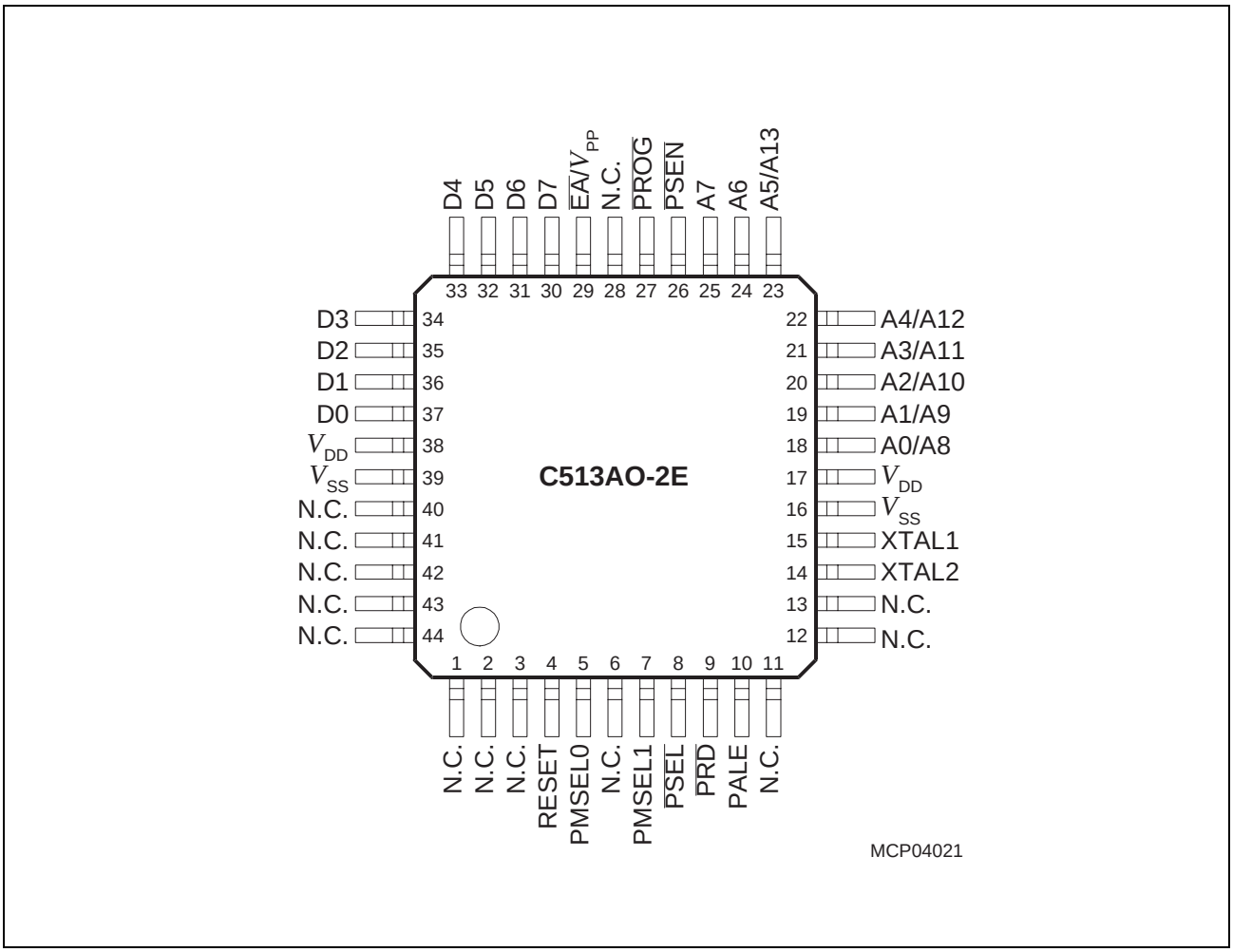


Figure 10-4
OTP Programming Mode Pin Configuration: P-MQFP-44 (top view)

10.3 OTP Programming Mode - Pin Definitions

The functional description of all C513AO-2E pins which are required for OTP memory programming are provided in **Table 10-1**.

Table 10-1
Pin Definitions and Functions of the C513AO-2E in Programming Mode

Symbol	Pin Number			I/O *)	Function
	P-DIP-40	P-LCC-44	P-MQFP-44		
RESET	9	10	4	I	**Reset** This input must be at static “1” (active) level throughout Programming Mode.
PMSEL0 PMSEL1	10, 11	11, 13	5 7	I I	**Programming Mode Selection Pins** These pins are used to select the different access modes in Programming Mode. PMSEL1,0 must satisfy a set up time to the rising edge of PALE. When the logic level of PMSEL1,0 is changed, PALE must be at low level.
$\overline{\text{PSEL}}$	12	14	8	I	**Basic Programming Mode Select** This input is used for the basic Programming Mode selection and must be switched according to **Figure 10-5**.
$\overline{\text{PRD}}$	13	15	9	I	**Programming Mode Read Strobe** This input is used for read access control for OTP memory read, Version Register read, and lock bit read operations.
PALE	14	16	10	I	**Programming Address Latch Enable** PALE is used to latch the high address lines. The high address lines must satisfy a set up and hold time to/from the falling edge of PALE. PALE must be at low level when the logic level of PMSEL1,0 is changed.
XTAL1	19	21	15	I	**XTAL1** Input to the oscillator amplifier.

PMSEL1	PMSEL0	Access Mode
0	0	Reserved
0	1	Read signature bytes
1	0	Program/read lock bits
1	1	Program/read OTP memory byte

*) I = Input
O = Output

Table 10-1
Pin Definitions and Functions of the C513AO-2E in Programming Mode (cont'd)

Symbol	Pin Number			I/O *)	Function
	P-DIP-40	P-LCC-44	P-MQFP-44		
XTAL2	18	20	14	O	**XTAL2** Output of the inverting oscillator amplifier.
V_{SS}	20	22	16	–	**Ground**
	–	1	39	–	**Ground, Optional** 1)
V_{DD}	40	23	17	–	**Power Supply** (+ 5 V)
	–	44	38	–	**Power Supply, Optional** 1)
P2.0-7	21-28	24-31	18-25	I	**Address Lines** P2.0-7 are used as multiplexed address input lines A0-A7 and A8-A13. A8-A13 must be latched with PALE.
$\overline{\text{PSEN}}$	29	32	26	I	**Program Store Enable** This input must be at static "0" level throughout Programming Mode.
$\overline{\text{PROG}}$	30	33	27	I	**Programming Mode Write Strobe** This input is used in Programming Mode as a write strobe for OTP memory program, and lock bit write operations During basic Programming Mode selection, a low level must be applied to $\overline{\text{PROG}}$.
$\overline{\text{EA}}/V_{PP}$	31	35	29	–	**Programming Voltage** This pin must be at 11.5 V (V_{PP}) voltage level during programming of an OTP memory byte or lock bit. During an OTP memory read operation, this pin must be at V_{IH} high level. This pin is also used for basic Programming Mode selection. At basic Programming Mode selection, a low level must be applied to $\overline{\text{EA}}/V_{PP}$.
P0.7-0	32-39	36-43	30-37	I/O	**Data Lines 0-7** In Programming Mode, data bytes are transferred via the bi-directional Port 0 data lines.
N.C.	1-8, 15-17,	2-9, 12, 17-19	1-3, 6, 11-13, 28, 40-44	–	**Not Connected** These pins should not be connected in Programming Mode.

*) I = Input
O = Output

1) These pins can be left unconnected to be compatible to the C504 OTP programming mode.

10.4 OTP Programming Mode Selection

Selection of OTP Programming Mode consists of two aspects:

- Basic Programming Mode selection
- Access Mode selection

Basic Programming Mode selection enables access of the device's OTP memory through the programming interface logic. After selection of the basic Programming Mode, the OTP memory accesses are executed according to the selected Access Mode. Access Modes include: OTP memory byte program/read, signature byte read, and program/read lock byte operations.

10.4.1 Basic Programming Mode Selection

The basic Programming Mode selection scheme is shown in **Figure 10-5**.

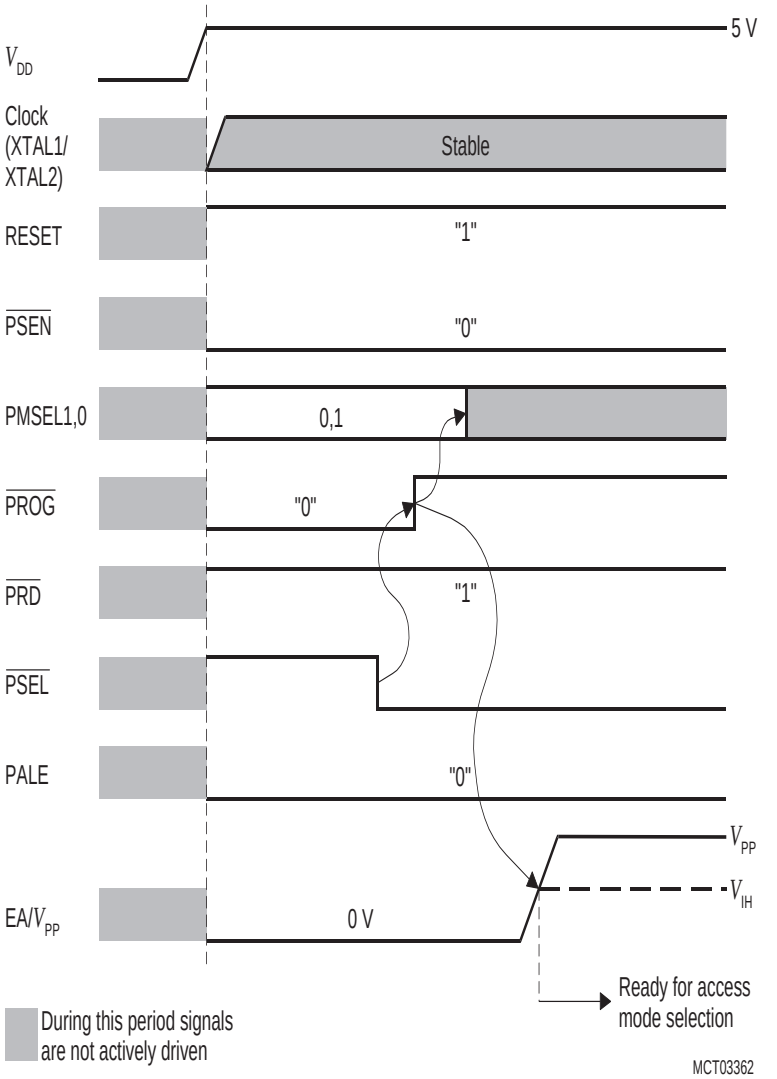


Figure 10-5
Basic Programming Mode Selection

Basic Programming Mode is selected by executing the following steps:

1. With a stable V_{DD}, a clock signal is applied to the XTAL pins; the RESET pin is set to “1” level and the PSEN pin is set to “0” level.
2. PROG, PALE, PMSEL1, and EA/V_{PP} are set to “0” level; PRD, PSEL, and PMSEL0 are set to “1” level.
3. PSEL is changed from “1” to “0” level and thereafter PROG is switched to “1” level.
4. PMSEL1,0 can now be changed; after EA/V_{PP} has been set to V_{IH} high level or to V_{PP}, the OTP memory is ready for access.

The pins RESET and PSEN must stay at “1” and “0” static signal levels respectively throughout Programming Mode. With a falling edge of PSEL, the logic states of PROG and EA/V_{PP} are internally latched. These two signals are now used as a programming write pulse signal (PROG) and as a programming voltage input pin V_{PP}. After the falling edge of PSEL, PSEL must stay at “0” state during all programming operations.

*Note: If Protection Level 1 to 3 has been programmed (see **Section 10.6**) and Programming Mode has been left, it is not possible to re-enter Programming Mode.*

10.4.2 OTP Memory Access Mode Selection

When the C513AO-2E has been put into Programming Mode using basic programming mode selection, several Access Modes of the OTP memory programming interface are available. The conditions for the different control signals of these access modes are listed in **Table 10-2**.

Table 10-2
Access Modes Selection

Access Mode	EA/ V_{PP}	PROG	PRD	PMSEL		Address (Port 2)	Data (Port 0)
				1	0		
Program OTP memory byte	V_{PP}	⊓	H	H	H	A0-7 A8-13	D0-7
Read OTP memory byte	V_{IH}	H	⊓				
Program OTP lock bits	V_{PP}	⊓	H	H	L	–	D1,D0 see **Table 10-3**
Read OTP lock bits	V_{IH}	H	⊓				
Read OTP version byte	V_{IH}	H	⊓	L	H	Byte addr. of version byte	D0-7

The Access Modes from the table above are basically selected by setting the two PMSEL1,0 lines to the required logic level. The PROG and PRD signals are the write and read strobe signals. Data is transferred via Port 0 and addresses are applied to Port 2.

The following sections describe the details of the different Access Modes.

10.5 Program/Read OTP Memory Bytes

The Program/Read OTP Memory Byte Access Mode is defined by PMSEL1,0 = 1,1. It is initiated when the PMSEL1,0 = 1,1 is valid at the rising edge of PALE. With the falling edge of PALE, the upper addresses A8-A13 of the 14-bit OTP memory address are latched. After A8-A13 has been latched, A0-A7 is put on the address bus (Port 2). A0-A7 must be stable when $\overline{PROG}$ is low or $\overline{PRD}$ is low. If subsequent OTP address locations are accessed with constant address information at the high address lines A8-13, A8-A13 must be latched only once (page address mechanism).

Figure 10-6 shows a typical OTP memory programming cycle followed by an OTP memory read operation. In this example, A0-A13 of the read operation are identical to A8-A13 of the proceeding programming operation.

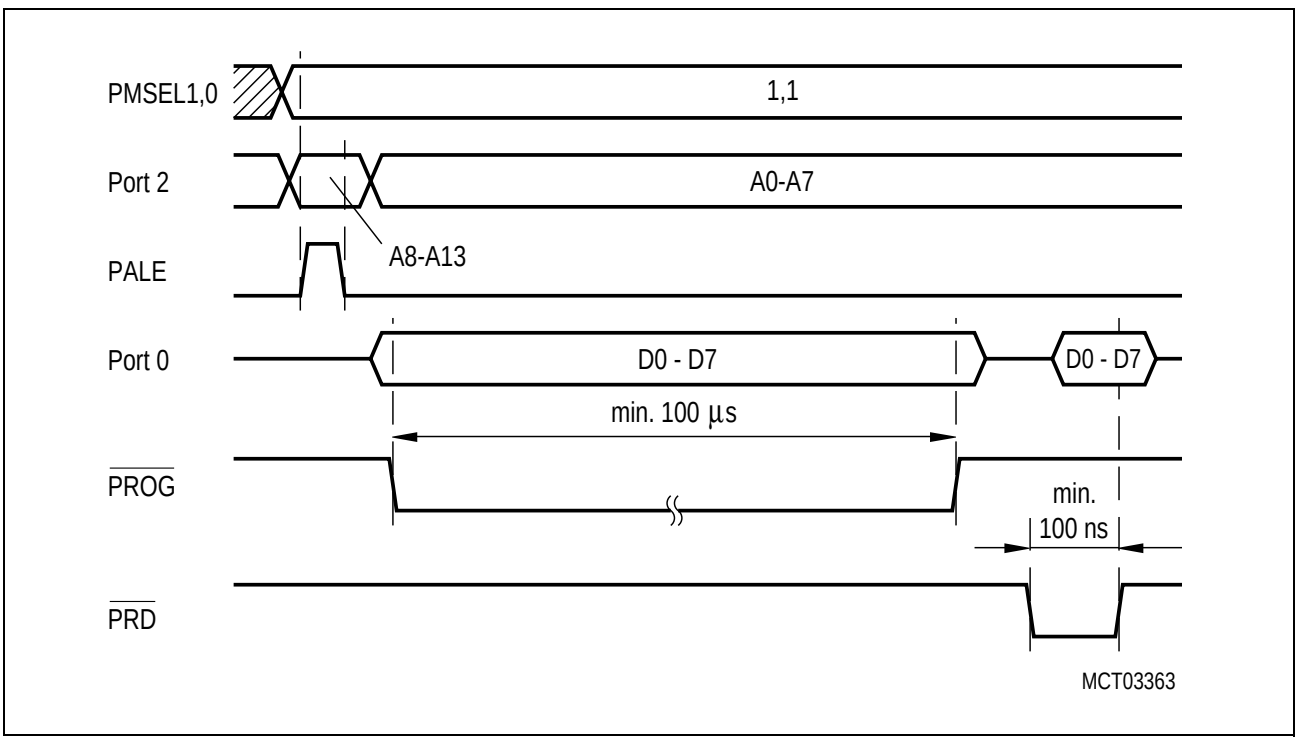


Figure 10-6
Programming/Verify OTP Memory Access Waveform

If the address lines A8-A13 must be updated, PALE must be activated for the latching of the new A8-A13 value. Control, address, and data information must be switched only when the $\overline{PROG}$ and $\overline{PRD}$ signals are at high level. The PALE high pulse must always be executed if a different Access Mode has been used prior to the current Access Mode.

For multiple OTP memory read operations, PALE must be activated only for latching a new A8-A13 address value. Control and address information must be switched only when the $\overline{PRD}$ or $\overline{PROG}$ signals are at high level.

Figure 10-7 shows a waveform example of the Program/Read Access Mode for several OTP memory bytes. In this example, OTP memory locations $3FD_H$ to 400_H are programmed. Thereafter, OTP memory locations 400_H and $3FD_H$ are read.

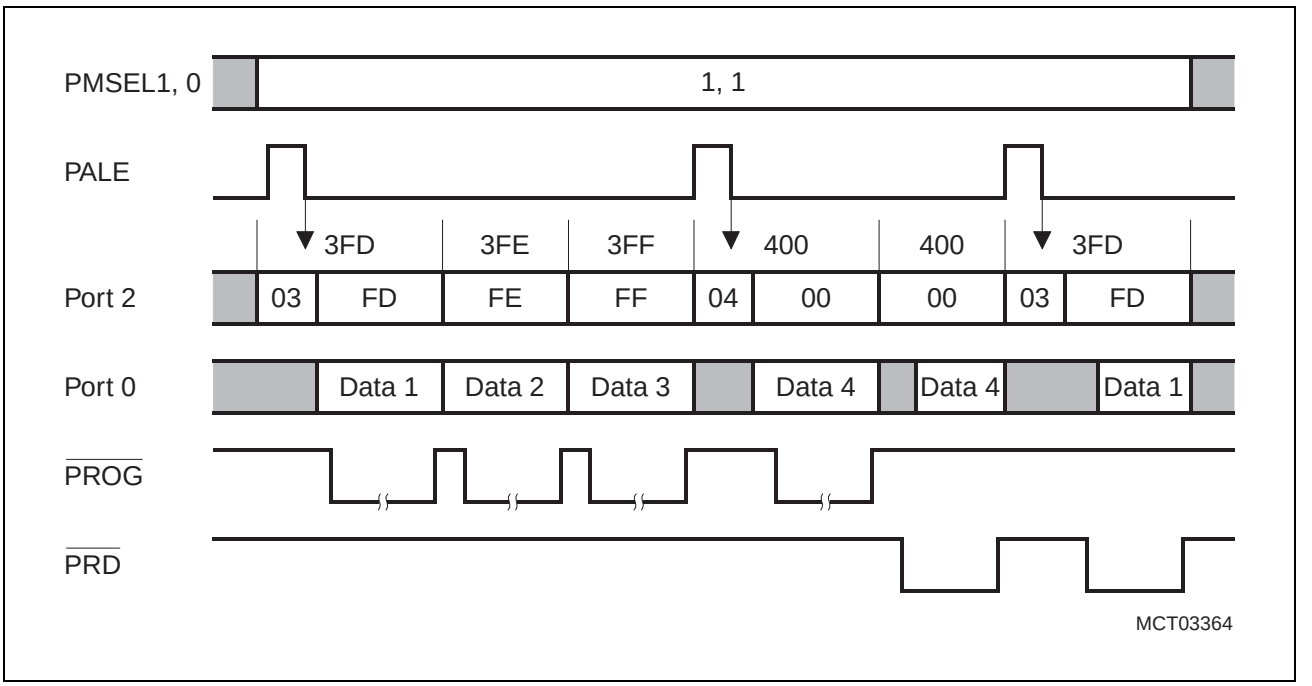


Figure 10-7
Typical OTP Memory Programming/Verify Access Waveform

10.6 Lock Bits Programming / Read

The C513AO-2E has two programmable lock bits which provide four levels of protection for the on-chip OTP code memory, when programmed according **Table 10-3**.

Table 10-3
Lock Bit Protection Types

Lock Bits at D1,D0		Protection Level	Protection Type
D1	D0		
1	1	Level 0	The OTP lock feature is disabled. During normal operation of the C513AO-2E, the state of the $\overline{EA}$ pin is not latched on reset.
1	0	Level 1	During normal operation of the C513AO-2E, MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset. An OTP memory read operation is possible only according to ROM/OTP Verification Mode 2. Further programming of the OTP memory is disabled (reprogramming security).
0	1	Level 2	Same as Level 1 except that OTP memory read operation using OTP verification mode is disabled also.
0	0	Level 3	Same as Level 2 except that external code execution by setting $\overline{EA}$ = low during normal operation of the C513AO-2E is no longer possible. External code execution, which is initiated by an internal program (for example: by an internal jump instruction above the ROM boundary), is still possible.

Note: A “1” means that the lock bit is not programmed. A “0” means that lock bit is programmed.

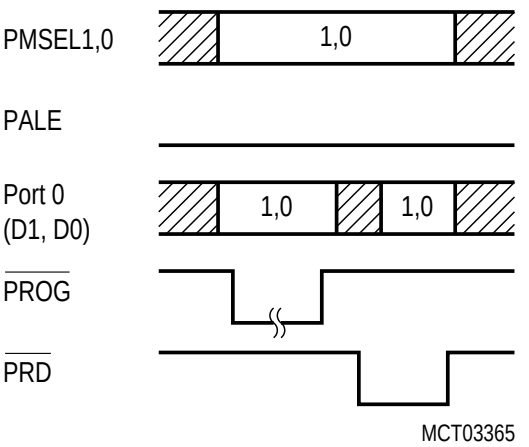
For an OTP verify operation at Protection Level 1, the C513AO-2E must be put into the OTP Verification Mode.

If a device is programmed with Protection Level 2 or 3, it is not possible to verify the OTP content of a customer rejected (FAR) OTP device.

When a protection level has been activated by programming the lock bits, basic Programming Mode must be left for activation of the protection mechanisms. This means that, after the activation of a protection level, further OTP program/verify operations are still possible if basic Programming Mode is maintained.

The state of the lock bits can always be read if Protection Level 0 is selected. If any of the Protection Levels 1 to 3 has been programmed and Programming Mode has been left, it is not possible to enter Programming Mode again. Additionally, in this case, the lock bits can no longer be read.

Figure 10-8 shows the waveform of a lock bit write/read access. To simplify the illustration, the $\overline{\text{PROG}}$ pulse has been shortened. In reality, for Lock Bit programming, a 100 µs $\overline{\text{PROG}}$ low pulse must be applied.



The example shows the programming and reading of a protection level 1.

Figure 10-8
Write/Read Lock Bit Waveform

10.7 Access of Version Bytes

The C513AO-2E provides three version bytes at address locations FC_H, FD_H, and FE_H. The information stored in the version bytes is defined by the mask of each microcontroller step, Therefore, the version bytes can be read, but not written. The three version bytes hold information such as manufacturer code, device type, and stepping code.

To read the version bytes, the control lines must be used in accordance with **Table 10-2** and **Figure 10-9**. The address of the version byte must be applied at the Port 1 address lines. PALE must not be activated.

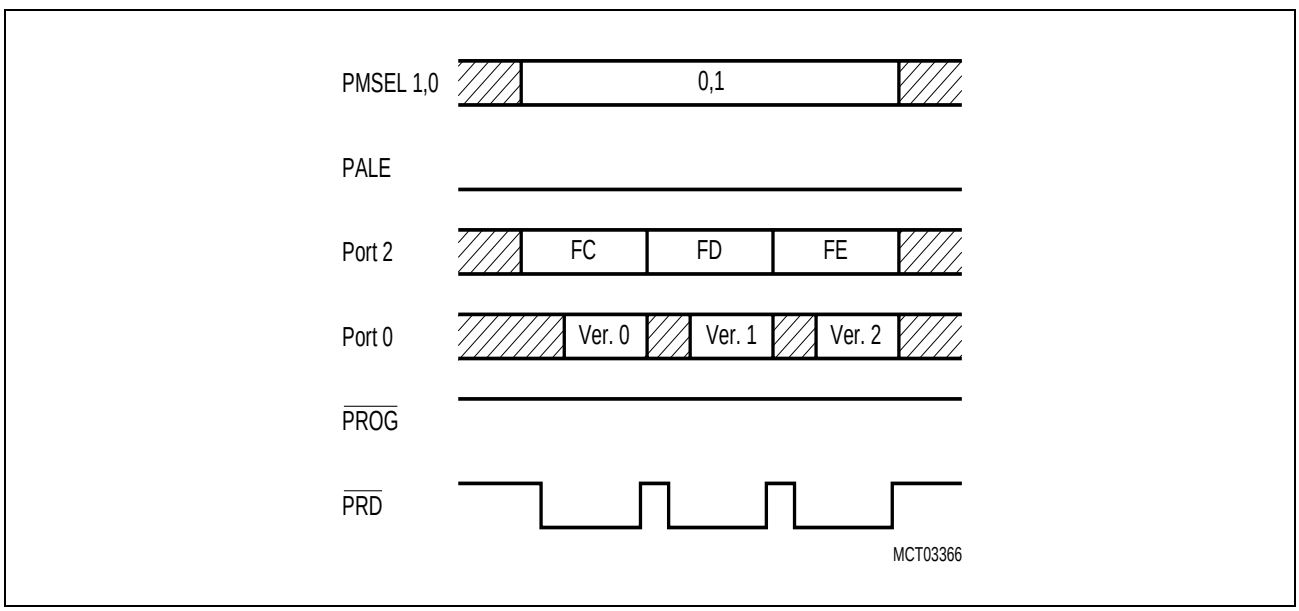


Figure 10-9
Read Version Byte(s) Waveform

Version bytes are typically used by programming systems for adapting the programming firmware to specific device characteristics such as OTP size etc.

Note: The three version bytes are implemented in such a way that they also can be read during Normal Program Execution Mode as a mapped register with bit RMAP in SFR SYSCON set. The addresses of the version bytes in Normal Mode are identical to the addresses which are used in Programming Mode. Therefore, in normal operating mode of the C513AO-2E, the SFR locations which hold the version bytes are referenced also as version registers.

11 Index

A

AC 3-9
ACC 3-6, 3-9
ALE signal 4-4

B

B 3-6, 3-9
Basic CPU timing 2-4
Block diagram 2-2, 10-1
BRS0 6-53
BRS1 6-53
BRS2 6-53

C

C/T 3-8
C/T2 3-9
CP/RL2 . . . . . 3-9, 6-23
CPHA 6-53
CPOL 6-53
CPU
  Accumulator 2-2
  B register 2-3
  Basic timing 2-4
  Program status word 2-2
  Stack pointer 2-3
CY 3-9

D

Data memory 3-2
DCEN 3-9
DPH 3-6, 3-8
DPL 3-6, 3-8

E

EA 3-8
EALE 1-10, 3-8
Emulation concept 4-5
ES 3-8
ET0 3-8
ET1 3-8
ET2 3-8
EWPD 3-8
EX0 3-8
EX1 3-8
Execution of instructions 2-5
EXEN2 3-9
EXF2 3-9
External bus interface 4-1–4-4
  ALE signal 4-4
  ALE switch-off control 4-4
  Overlapping of data/program memory 4-3
  Program memory access 4-3
  Program/data memory timing 4-2
  PSEN signal 4-3
  Role of P0 and P2 4-1

F

F0 3-9
F1 3-9
Fail save mechanisms 8-1–8-7
Fast power-on reset 5-3
Features 1-2
Functional units 1-1
Fundamental structure 2-1

G

GATE 3-8
General purpose registers 3-2
GF0 3-8
GF1 3-8

H

Hardware reset 5-1–5-5

I

I/O ports 6-1–6-14
IDLE 3-8
IE 3-6, 3-8
IE0 3-8
IE1 3-8
INT0 3-8
INT1 3-8
Interrupt system 7-1–7-14
  Block diagrams 7-2
  Entry sequence timing 7-12
  External Interrupts 7-13
    Edge/level triggering 7-14
    Interrupt detection 7-14
  General structure 7-3
  Handling procedure 7-12
  Priority within level structure 7-11
  Registers 7-4–7-11
    Enable registers 7-4
    Priority registers 7-10, 7-11
  Response time 7-14
Interrupts
  Request flags 7-5–7-9
IP 3-6, 3-8

IT0 3-8
IT1 3-8

L

Logic symbol 1-3
LOOPB 6-55
LSBSM 6-55

M

M0 3-8
M1 3-8
Memory map 3-1
Memory organization 3-1
Data memory 3-2
General purpose registers 3-2
Memory map 3-1
Program memory 3-2
MSTR 6-52

O

Oscillator operation 5-6–5-7
External clock source 5-7
On-chip oscillator circuitry 5-7
Recommended oscillator circuit 5-6
Oscillator watchdog 8-5–8-7
Block diagram 8-6
Fast power-on reset 8-7
OTP memory of the C504-2E . .10-1–10-13
Basic mode selection 10-7
Pin configuration 10-2
Pin definitions and functions . .10-5, 10-7
Programming mode 10-1
OTP Programming Interface
Access modes 10-8–10-13
Lock bits access 10-11–10-12
Program/Read Bytes 10-9–10-10
Programming mode selection .10-7–10-8
Version Byte access 10-13
OV 3-9
OWDS 3-8

P

P 3-9
P0 . . . . . .3-6, 3-8
P1 . . . . . .3-6, 3-8
P2 . . . . . .3-6, 3-8
P3 . . . . . .3-6, 3-8
Parallel I/O 6-1–6-14
PCON 3-6, 3-7, 3-8, 6-33
PCON1 3-8
PCON14 3-7
PDE 3-8
Pin configuration 1-4, 1-5
Pin definitions and functions 1-7–1-11
Ports 6-1–6-14
Loading and interfacing 6-13
Output/input sample timing 6-12
Read-modify-write operation 6-14
Types and structures 6-1
Power saving modes 9-1–9-8
Behaviour of external pins 9-3
Idle mode 9-3–9-4
Power down mode 9-6–9-8
Entering 9-6
External wake-up timing 9-7
Functionality 9-6
Termination 9-7
Register PCON 9-1
Register PCON1 9-2
Slow down mode 9-5
PS 3-8
PSEN signal 4-3
PSW 3-6, 3-9
PT0 3-8
PT1 3-8
PT2 3-8
PX0 3-8
PX1 3-8

R

RB8 3-8
RC2H 3-7, 3-9
RC2L 3-7, 3-9
RCLK 3-9
RD 3-8
Recommended oscillator circuit 5-6
REN 3-8
Reset 5-1
Fast power-on reset 5-3
Hardware reset timing 5-5
Power-on reset timing 5-4
Reset circuitries 5-2
RI 3-8, 7-7
RMAP 3-8
ROM/OTP protection 4-6
Protected ROM verification example . .4-8
Protected ROM verifiy timimg 4-7
Protected ROM/OTP mode 4-7

Unprotected ROM mode 4-6
Unprotected ROM verifiy timimg 4-6
RS0 3-9
RS1 3-9
RxD 3-8

S

SBUF 3-6, 3-8
SCEN 6-52
SCF 3-9
SCIEN 6-54
SCON 3-6, 3-8, 7-7
Serial interface (USART) 6-30–6-45
Baudrate generation 6-33–6-37
with timer 1 6-35
with timer 2 6-36
Multiprocessor communication 6-31
Operating mode 0 6-37–6-39
Operating mode 1 6-40–6-42
Operating mode 2 and 3 6-43–6-45
Registers 6-31–6-32
SM0 3-8
SM1 3-8
SM2 3-8
SMOD 3-8
SP 3-6, 3-8
Special function registers 3-5
Access with RMAP 3-5
Table - address ordered 3-8–3-9
Table - functional order. 3-6–3-7
SRB 3-9, 6-55
SSC interface 6-46–6-55
Baudrate generation 6-48
Block diagram 6-46
General operation 6-47
Master mode timing 6-50
Master/slave mode 6-49
Registers 6-52–6-55
Slave mode timing 6-51
Write collision detection 6-48
SSCCON 6-52
SSCMOD 6-55
STB 3-9, 6-55
SWDT 3-8
SYSCON 3-5, 3-6, 3-8, 4-4

T

T0 3-8
T1 3-8
T2 3-8
T2CON 3-7, 3-9, 7-8
T2EX 3-8
T2MOD 3-7, 3-9
TB8 3-8
TC 7-9
TCEN 6-54
TCLK 3-9
TCON 3-6, 3-8, 7-6
TEN 6-52
TF0 . . . . . .3-8, 7-6
TF1 . . . . . .3-8, 7-6
TF2 3-9
TH0 . . . . . .3-6, 3-8
TH1 . . . . . .3-6, 3-8
TH2 . . . . . .3-7, 3-9
TI 3-8, 7-7
Timer/counter 6-15–6-45
Timer/counter 0 and 1 6-15–6-22
Mode 0, 13-bit timer/counter 6-19
Mode 1, 16-bit timer/counter 6-20
Mode 2, 8-bit rel. timer/counter . . .6-21
Mode 3, two 8-bit timer/counter . . .6-22
Registers 6-16–6-18
Timer/counter 2 6-23–6-29
16-bit auto-reload mode . . . 6-26–6-28
16-bit capture mode 6-29
Operating modes 6-23
Registers 6-24–6-26
TL0 . . . . . .3-6, 3-8
TL1 . . . . . .3-6, 3-8
TL2 . . . . . .3-7, 3-9
TMOD 3-6, 3-8
TR0 3-8
TR1 3-8
TR2 . . . . . .3-9, 6-23
TRIO 6-55
TxD 3-8

V

VR0 3-9
VR1 3-9
VR2 3-9

W

Watchdog timer 8-1–8-4
Block diagram 8-1
Refreshing of the WDT 8-4
Registers WDCON and WDTREL 8-2

Reset operation 8-4
Starting of the WDT 8-3
Time-out periods 8-3
WCEN 6-54
WCOL 7-9
WDCON 3-7, 3-8
WDT 3-8
WDTPSEL 3-8
WDTREL 3-7, 3-8
WDTS 3-8
WR 3-8

X

XMAP 3-8
XRAM operation 3-3
Accessing twith DPTR 3-3
Accessing with R0/R1 3-4
Reset 3-3