SIEMENS

SAB-C502

8-Bit Single-Chip Microcontroller

User's Manual 08.94

http://www.siemens.de/
Semiconductor/

SAB-C502 User's Manual **Revision History:**	**08.94 Original Version**
Previous Version:	11.92 Target Specification Revision 1.3
Page	**Subjects (major changes since last revision)**
1-8	P2 pin numbers corrected
1-9	N.C. pin numbers corrected
3-17	IE bits corrected
5-4	Figure 5-2: RESET signal corrected
6-5	Additional text about port structures added
6-7	Text modified (6.1.2)
6-32	Figure 6-14 modified
6-35	Formula for baud rate corrected
6-39, 6-42	Text of 3rd paragraph modified ("Transmission starts...")
8-5	Figure 8-2 modified
9-2	Table 9-1: 3rd row modified
10-2	I_{TL} also valid for port 1; $I_{\mathrm{CC\,max}}$ values added
10-3	Note 3 modified
10-4, 10-5	Note 7: formula for $I_{\mathrm{CC\,max}}$ added
10-4, 10-7	t_{LLAX}, t_{LLAX2} value corrected
10-8	t_{LLIV} name corrected
10-9	t_{RHDZ} value corrected; t_{LLAX2} value corrected
10-12	20 MHz clock high/low/rise/fall times changed
10-14	t_{LLAX} corrected into t_{LLAX2}
	AC testing waveforms reference point corrected

Edition 08.94

This edition was realized using the software system FrameMaker®.

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

Attention please!

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

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

Terms of delivery and rights to change design reserved.

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

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

Siemens AG is an approved CECC manufacturer.

Packing

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

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

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

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

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

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

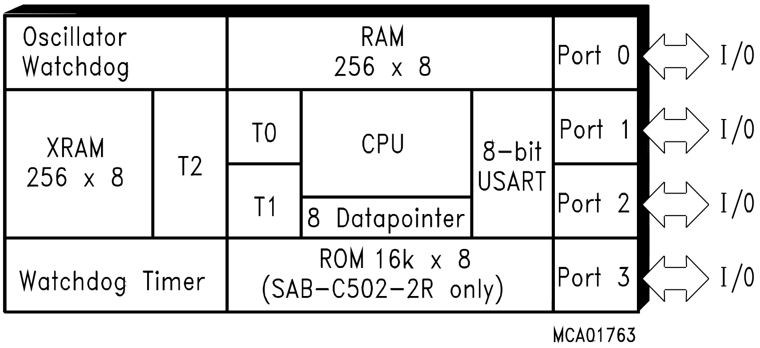
Table of Contents **Page**

Table of Contents Page

1 Introduction

The SAB-C502-L/C502-2R described in this document is compatible (also pin-compatible) with the SAB 80C52 and can be used for all present SAB 80C52 applications.

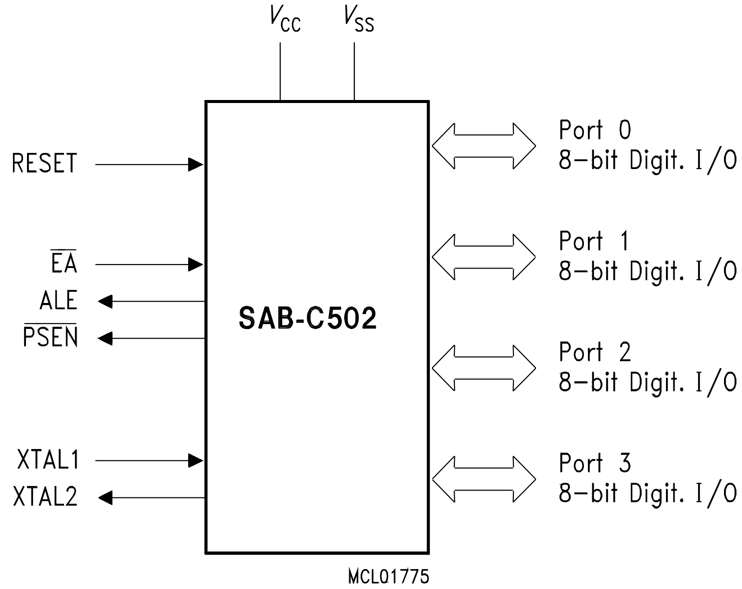
The SAB-C502-2R contains a non-volatile 16K×8 read-only program memory, a volatile 256×8 read/write data memory, four ports, three 16-bit timers/counters, a seven source, two priority level interrupt structure, a serial port and versatile fail save mechanisms. The SAB-C502-L is identical, except that it lacks the program memory on chip. The SAF-C502 is identical, except for the extended temperature range. Therefore the term SAB-C502 refers to all versions within this specification unless otherwise noted.



SAB-C502

Listed below is a summary of the main features of the SAB-C502:

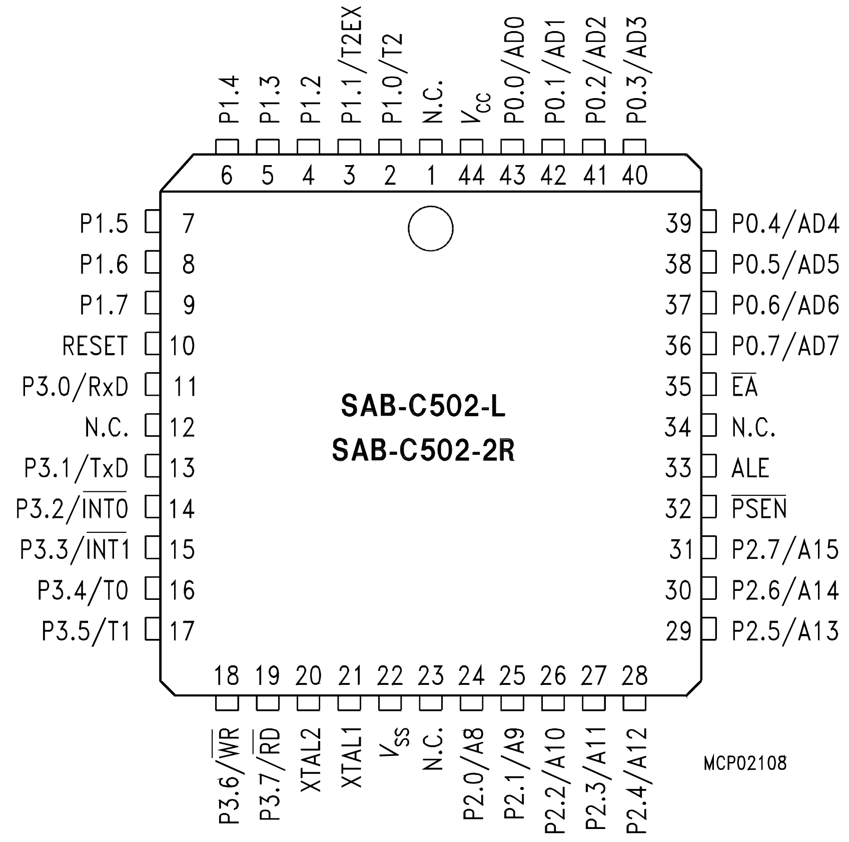
- Fully compatible to standard 8051 microcontroller
- Versions for 12 / 20 MHz operating frequency
- 16 K×8 ROM (SAB-C502-2R only)
- 256×8 RAM
- 256×8 XRAM (additional on-chip RAM)
- Eight datapointers for indirect addressing of program and external data memory (including XRAM)
- Four 8-bit ports
- Three 16-bit Timers / Counters (Timer 2 with Up/Down Counter feature)
- USART
- Six interrupt sources, two priority levels
- Programmable 15-bit Watchdog Timer
- Oscillator Watchdog
- Fast Power On Reset
- Power Saving Modes
- P-DIP-40 and P-LCC-44 packages
- Temperature ranges: SAB-C502 T_A : 0 to 70°C
  SAF-C502 T_A : – 40 to 85°C



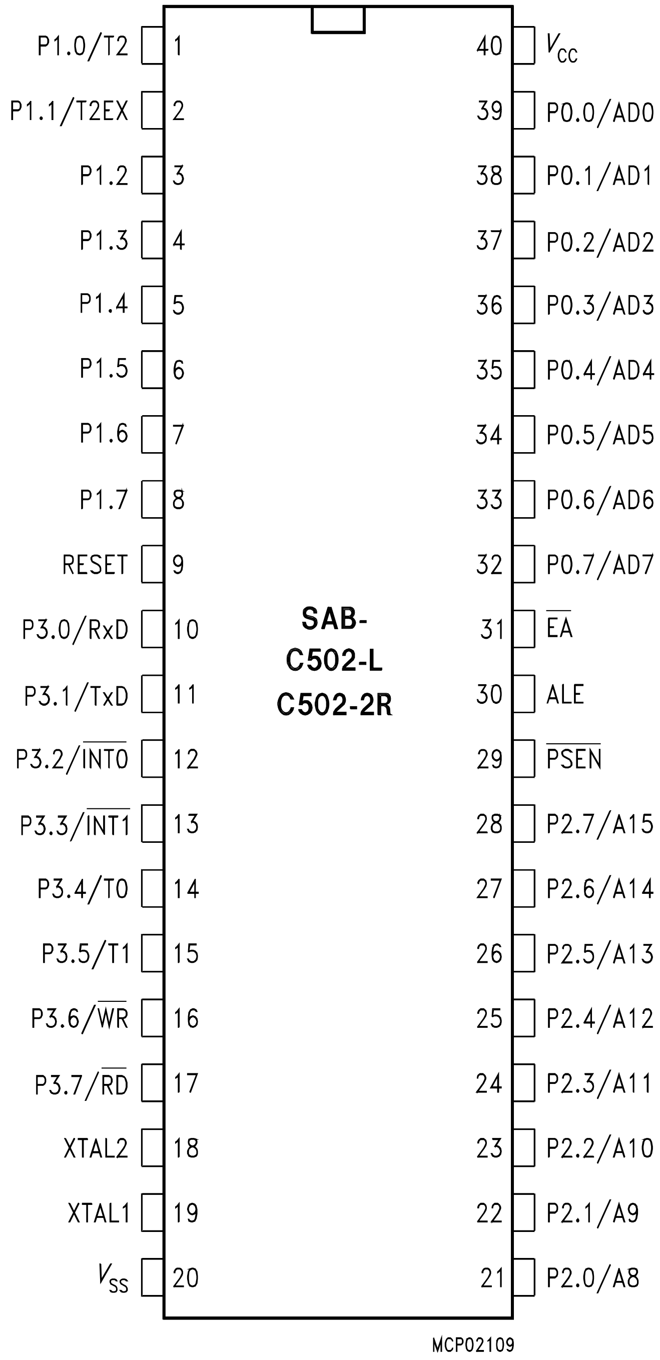
Logic Symbol

Pin Configuration

(top view)



Pin Configuration
(top view)



Pin Definitions and Functions

Symbol	Pin Number		I/O *)	Function
	P-LCC-44	P-DIP-40		
P1.7-P1.0	9-2	8-1	I	**Port 1** is a bidirectional I/O port with internal pull-up resistors. Port 1 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (IIL, in the DC characteristics) because of the internal pullup resistors. Port 1 also contains the timer 2 pins as secondary function.The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 1, as follows:
	2	1		P1.0 T2 Input to counter 2
	3	2		P1.1 T2EX Capture -Reload trigger of timer 2 Up-Down count

*) I = Input
O = Output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O *)	Function
	P-LCC-44	P-DIP-40		
P3.0-P3.7	11, 13-19	10-17	I/O	**Port 3** is a bidirectional I/O port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port 0 and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows
	11	10		P3.0 R×D receiver data input (asynchronous) or data input/output (synchronous) of serial interface 0
	13	11		P3.1 T×D: transmitter data output (asynchronous) or clock output (synchronous) of serial interface 0
	14	12		P3.2 $\overline{\text{INT0}}$ interrupt 0 input/timer 0 gate control
	15	13		P3.3 $\overline{\text{INT1}}$ interrupt 1 input/timer 1 gate control
	16	14		P3.4 T0 counter 0 input
	17	15		P3.5 T1 counter 1 input
	18	16		P3.6 $\overline{\text{WR}}$ write control signal latches the data byte from port 0 into the external data memory
	19	17		P3.7 $\overline{\text{RD}}$ the read control signal enables the external data memory to port 0
XTAL2	20	18	–	**XTAL2** Output of the inverting oscillator amplifier.

*) I = Input
O = Output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O *)	Function
	P-LCC-44	P-DIP-40		
XTAL1	21	19	–	**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
P2.0-P2.7	24-31	21-28	I/O	**Port 2** is a bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1s written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullup resistors when issuing 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register.
$\overline{\text{PSEN}}$	32	29	O	The $\overline{\textbf{Program Store Enable}}$ output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periodes except during external data memory accesses. Remains high during internal program execution.
RESET	10	9	I	**RESET** A high level on this pin for two machine cycls 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_{CC}.

*) I = Input
O = Output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number		I/O *)	Function
	P-LCC-44	P-DIP-40		
ALE	33	30	O	The **Address Latch Enable** output is used for latching the low-byte of the address into external memory during normal operation. It is activated every six oscillator periodes except during an external data memory access.
$\overline{\text{EA}}$	35	31	I	$\overline{\textbf{External Access Enable}}$ When held at high level, instructions are fetched from the internal ROM (SAB-C502-2R only) when the PC is less than 4000_H. When held at low level, the SAB-C502 fetches all instructions from external program memory. For the SAB-C502-L this pin must be tied low.
P0.0-P0.7	43-36	39-32	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impendance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pullup resistors when issuing 1s. Port 0 also outputs the code bytes during program verification in the SAB-C502-2R if ROM-protection was not enabled. External pullup resistors are required during program verification.
V_{SS}	22	20	–	**Circuit ground potential**
V_{CC}	44	40	–	**Supply terminal** for all operating modes
N.C.	1, 12, 23, 34	–	–	**No connection**

*) I = Input
O = Output

2 Fundamental Structure

The SAB-C502 is fully compatible to the standard 8051 microcontroller family.

It is compatible with the SAB 80C52. While maintaining all architectural and operational characteristics of the SAB 80C52 the SAB-C502 incorporates some enhancements in the Timer2 and Fail Save Mechanism Unit.

Figure 2-1 shows a block diagram of the SAB-C502.

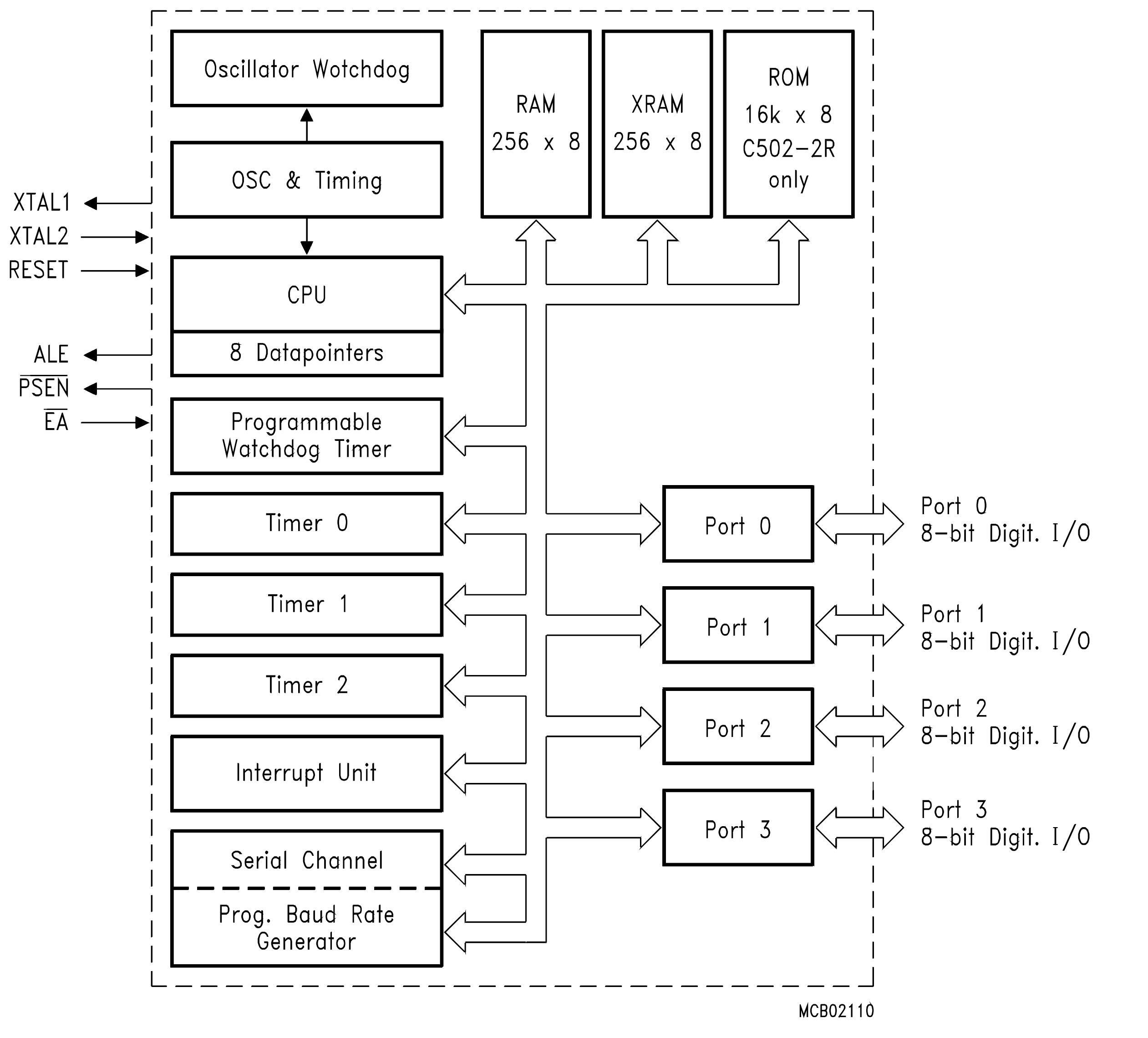


Figure 2-1
Block Diagram of the SAB-C502

2.1 CPU

The SAB-C502 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 12-MHz crystal, 58% of the instructions execute in 1.0 µs (20 MHz : 600 ns).

The CPU (Central Processing Unit) of the SAB-C502 consists of the instruction decoder, the arithmetic section and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers and control the ALU processing.

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

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

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

Accumulator

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

Program Status Word

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

Special Function Register PSW (Address D0$_H$)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
Addr. D0$_H$	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

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

Reset value of PSW is 00$_H$.

B Register

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

Stack Pointer

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

2.2 CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is devided into a phase 1 half, during which the phase 1 clock is active, and a phase 2 half, during which the phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbererd S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts for two oscillator periods. Typically, arithmetic and logically 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 signals and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

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

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

Most SAB-C502 instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed. **Figure 2-2 c)** and **d)** show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

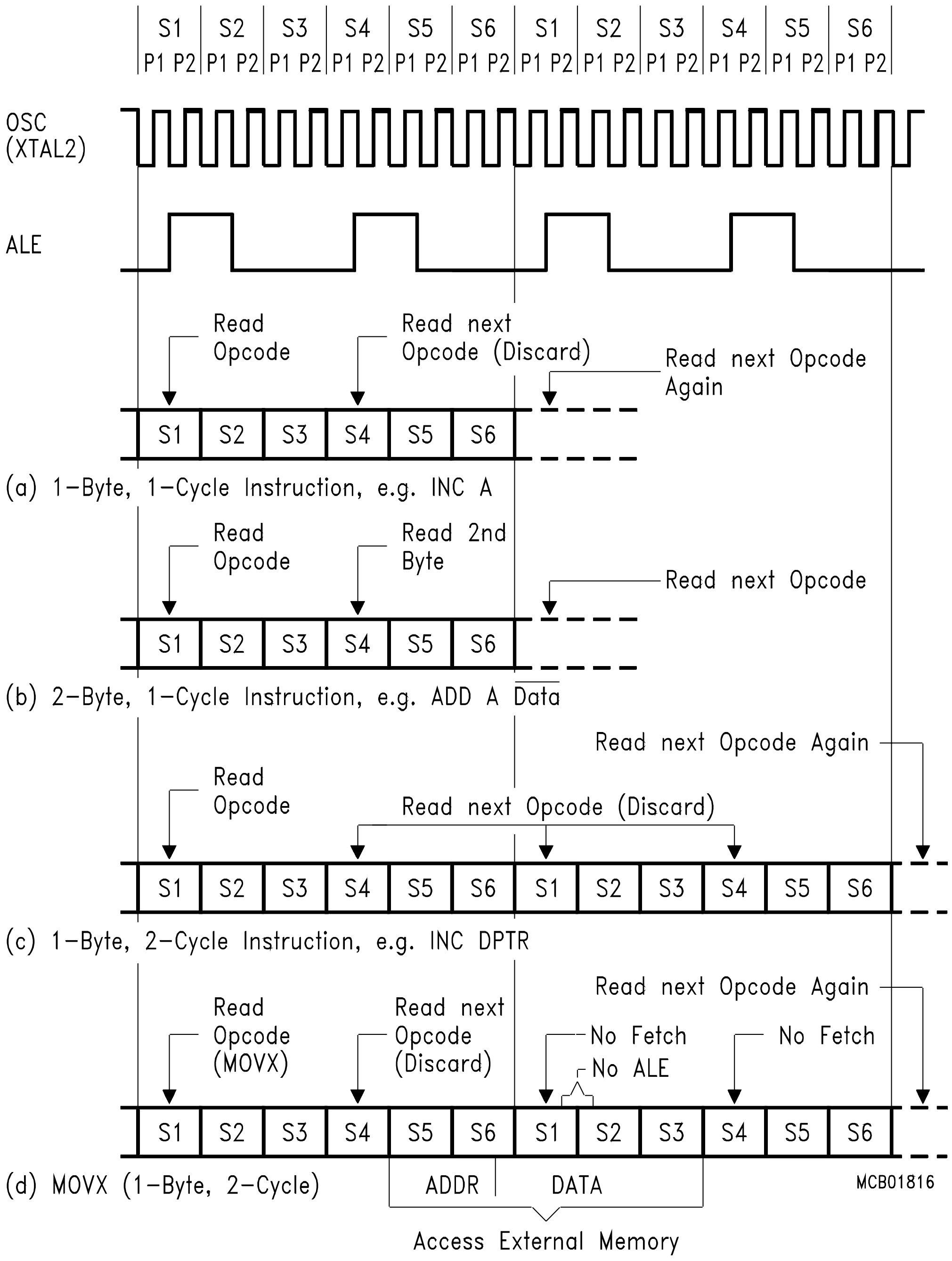

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6
P1 P2
OSC (XTAL2)
ALE
Read Opcode
Read next Opcode (Discard)
Read next Opcode Again
(a) 1–Byte, 1–Cycle Instruction, e.g. INC A
Read Opcode
Read 2nd Byte
Read next Opcode
(b) 2–Byte, 1–Cycle Instruction, e.g. ADD A Data
Read next Opcode Again
Read Opcode
Read next Opcode (Discard)
(c) 1–Byte, 2–Cycle Instruction, e.g. INC DPTR
Read next Opcode Again
Read Opcode (MOVX)
Read next Opcode (Discard)
No Fetch
No ALE
No Fetch
(d) MOVX (1–Byte, 2–Cycle)
ADDR
DATA
Access External Memory
MCB01816

Figure 2-2
Fetch Execute Sequence

3 Memory Organization

The SAB-C502 CPU manipulates operands in the following four address spaces:

- up to 64 Kbyte of external program memory
- up to 64 Kbyte of external data memory
- 256 bytes of internal data memory
- a 256 x 8 area which is accessed like external RAM (MOVX-instructions), called XRAM implemented on-chip
- a 128 byte special function register area

Figure 3-1 illustrates the memory address spaces of the SAB-C502.

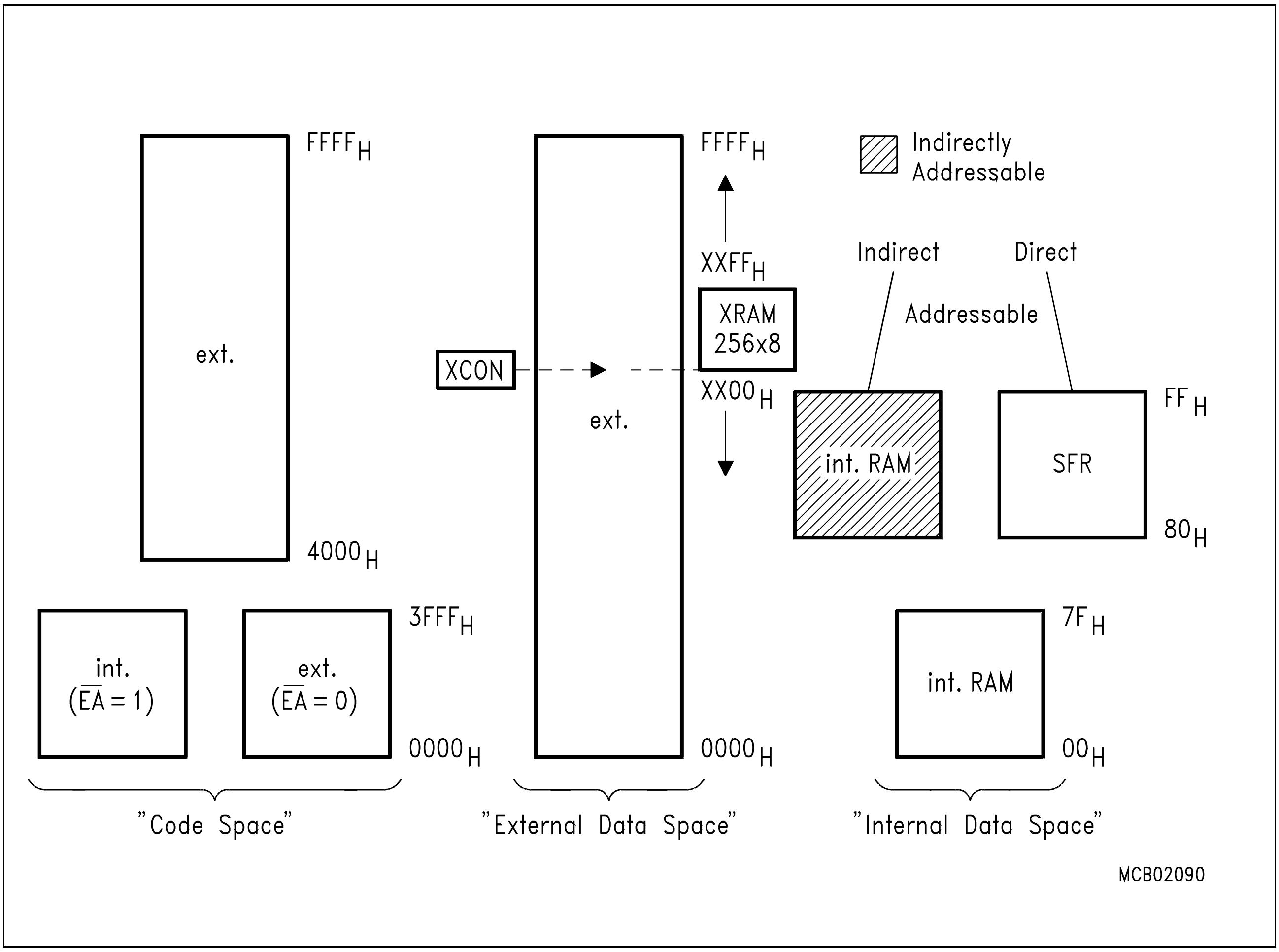


Figure 3-1
Memory Map

3.1 Program Memory, "Code Space"

The SAB-C502-2R has 16 Kbytes of read-only program memory, while the SAB-C502-L has no internal program memory. The program memory can be externally expanded up to 64 Kbytes. If the $\overline{EA}$ pin is held high, the SAB-C502 executes out of internal ROM unless the address exceeds $3FFF_H$. Locations 4000_H through $FFFF_H$ are then fetched from the external program memory. If the $\overline{EA}$ pin is held low, the SAB-C502 fetches all instructions from the external program memory.

3.2 Data Memory, "Data Space"

The data memory address space consists of an internal and an external memory space. The SAB-C502 contains another 256 Byte of On-Chip RAM additional to the 256 byte internal RAM. This RAM is called XRAM (‘e**X**tended **RAM**’) in this document.

– **Internal Data Memory**
  The internal data memory is divided into three physically separate and distinct blocks:
  - the lower 128 byte of RAM including four register banks containing eight registers each
  - the upper 128 byte of RAM
  - the 128 byte special function register area
  - a 256 byte area which is accessed like external RAM (MOVX-instructions), implemented on-chip. Special Function Register SYSCON controls whether data is read from or written to XRAM or external RAM.

– **External Data Memory**
  Up to 64 Kbyte external data memory can be addressed by instructions that use 8-bit or 16-bit indirect addressing. For 8-bit addressing MOVX instructions in combination with registers R0 and R1 can be used. A 16-bit external memory addressing is supported by a 16-bit datapointer. Registers XPAGE, XCON, and SYSCON control whether data fetches at addresses $XX00_H$ to $XXFF_H$ (high byte address is defined by the content of SFR XCON) are done from internal XRAM or from external data memory.

  Eight Datapointers for Faster External Bus Access
  For complex applications with numerous external peripherals or extended data storage capacity only one datapointer would be a ‘bottle-neck’ for communication to the external world. Especially programming in high level languages (PLM51, ‘C’, PASCAL51) requires extended RAM capacity and at the same time a fast access to this additional RAM because of the reduced code efficiency of these languages.
  The SAB-C502 contains a set of eight 16-bit datapointers from which the actual datapointer can be selected. This means that the user’s program may keep up to eight 16-bit addresses resident in these registers, but only one register at a time is selected to be the datapointer. Thus the datapointer in turn is accessed (or selected) via indirect addressing.
  This indirect addressing is done through a special function register called DPSEL (data pointer select register). All instructions of the SAB-C502 which can handle the datapointer therefore affect only one of the eight pointers which is addressed by DPSEL at the very moment.

Special Function Register DPSEL (Address 92_H)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
92_H	–	–	–	–	–	.2	.1	.0	DPSEL

Bit	Function
–	Not implemented, reserved for future use
DPSEL.2-0	Datapointer select biits 0 0 0 Datapointer 0 selected 0 0 1 Datapointer 1 selected 0 1 0 Datapointer 2 selected 0 1 1 Datapointer 3 selected 1 0 0 Datapointer 4 selected 1 0 1 Datapointer 5 selected 1 1 0 Datapointer 6 selected 1 1 1 Datapointer 7 selected

Reset value of DPSEL is XXXX $X000_B$.

The advantage of using multiple datapointers is a faster execution of external accesses and less code. Whenever the content of the datapointer must be altered between eight 16-bit addresses, one single instruction, which selects one of the other datapointers, does this job. If the program uses just one datapoinrter, the n it has to save the old value (with two 8-bit instructions) and load the new address, byte by byte. This not only takes more time, it also requires additional space in the internal RAM.

3.3 Architecture of the XRAM

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

The additional on-chip XRAM is logically located in the “external data memory” range. The location is under control of SFR XCON (default location after reset is $F800_H$ - $F8FF_H$). It is possible to enable and disable (only by reset) the XRAM. If it is disabled, the device shows the same behaviour as the parts without XRAM, i.e. all MOVX accesses use the external bus to physically address external data memory.

Special Function Register XCON (Address 94_H)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
94_H	.7	.6	.5	.4	.3	.2	.1	.0	XCON

Bit	Function
XCON.7-0	XRAM start address (high byte) These bits are used to program the high byte address of the XRAM start address to determine the location in external data memory space.

Reset value of XCON is $F8_H$.

3.3.1 Accesses to XRAM

Because the XRAM is used in the same way as external data memory the same instruction types must be used for accessing the XRAM.

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

Reset detection at cycle 1 : The new value will not be written to XRAM. The old value is not affected.

Reset detection at cycle 2 : The old value in XRAM is overwritten by the new value.

3.3.2 Accesses to XRAM using the DPTR

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

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

Normally the use of these instructions would use a physically external memory. However, in the SAB-C502 the XRAM is accessed if it is enabled and if the DPTR points to the XRAM address space.

3.3.3 Accesses to XRAM using the Registers R0/R1

The 8051 architecture provides also instructions for accesses to external data memory range which use only an 8-bit address (indirect addressing with registers R0 or R1). These instructions are :

– MOVX	A, @Ri	(Read)
– Movx	@Ri, A	(Write)

In application systems, either a real 8-bit bus (with 8-bit address) is used or port 2 serves as page register which selects pages of 256 bytes. However, the distinction, whether port 2 is used as general purpose I/O or as “page address” is made by the external system design. From the device’s point of view it cannot be decided whether the port 2 data is used externally as address or as I/O data.

Hence, a special page register is implemented into the SAB-C502 to provide the possibility of accessing the XRAM also with the MOVX @Ri instructions, i.e. XPAGE serves the same function for the XRAM as port 2 for external data memory.

Special Function Register XPAGE (Address 91_H)

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
91_H	.7	.6	.5	.4	.3	.2	.1	.0	XPAGE

Reset value of XPAGE is 00_H.

XPAGE can be set and read by software.

Figure 3-2 to **3-4** show the dependencies off XPAGE- and port 2 - addressing in order to explain the differences in accessing XRAM, external RAM or what is to do when port 2 is used as an I/O port.

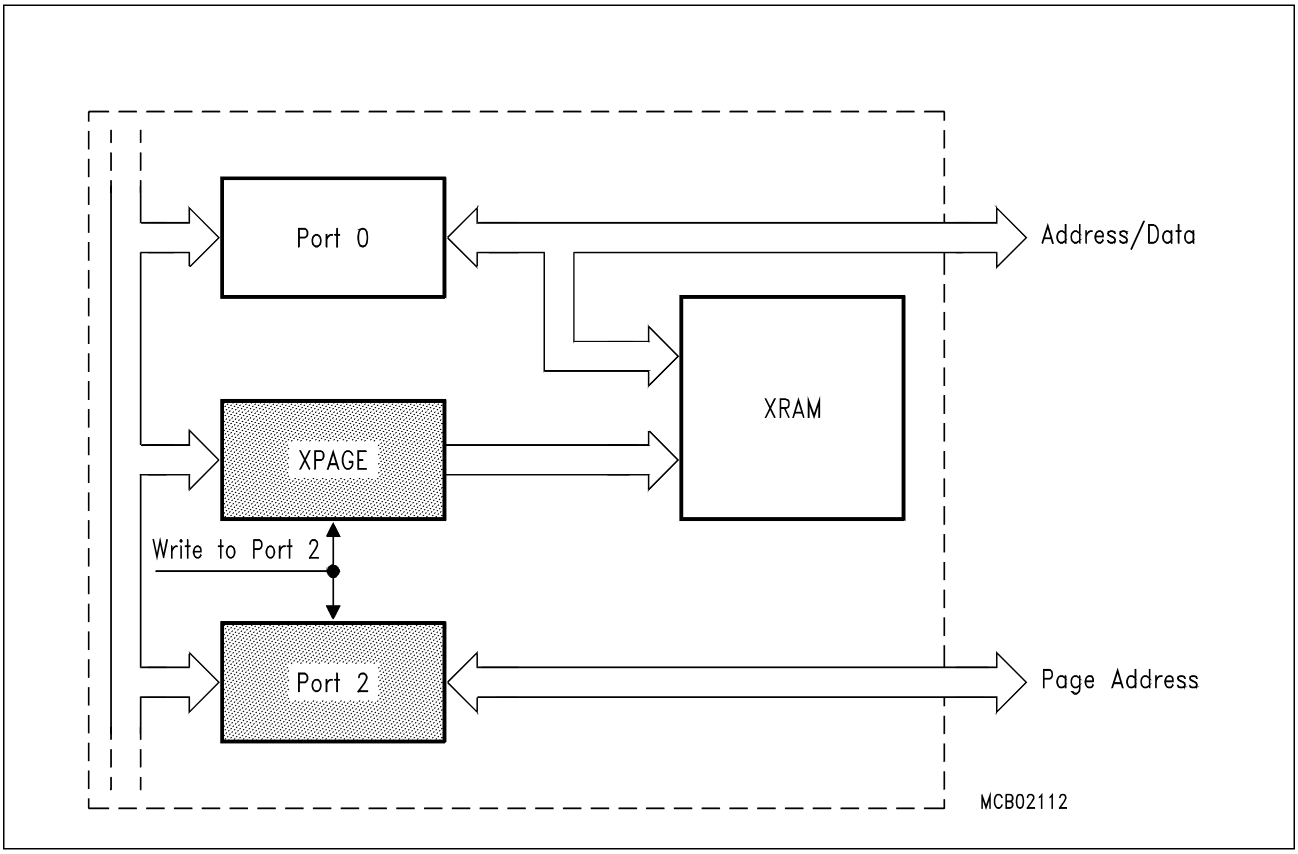


Figure 3-2
Write Page Address to Port 2

MOV P2, pageaddress will write the page address to port 2 and XPAGE register.

When external RAM located in the XRAM address area is to be accessed, XRAM has to be disabled. When external RAM outside the XRAM area is to be addressed, XRAM may remain enabled. In this case there is no need to override XPAGE by a second move instruction.

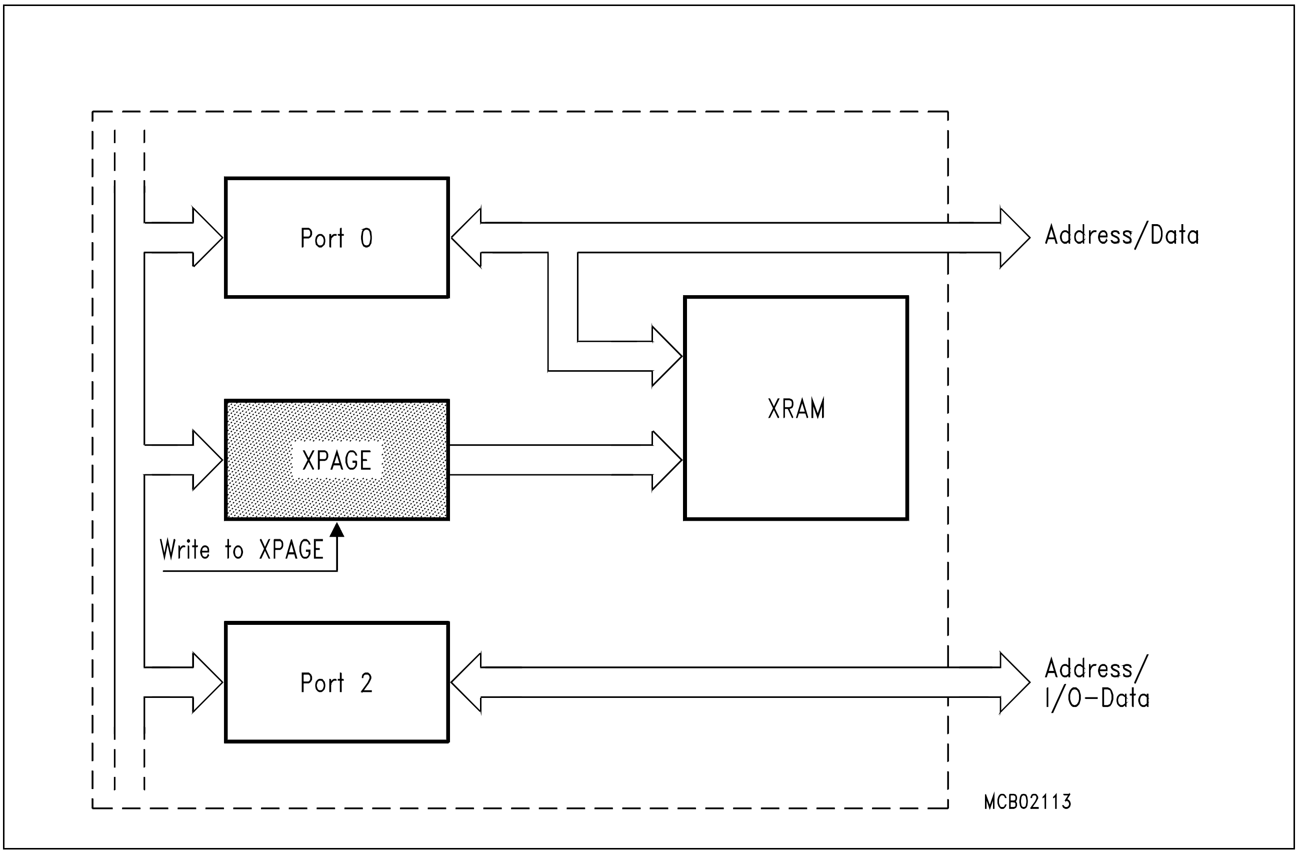


Figure 3-3
Write Page Address to XPAGE

Tthe page address is only written to XPAGE register. Port 2 is available for addresses or I/O data. See **figure 3-4** to see what happens when port 2 is used as I/O port.

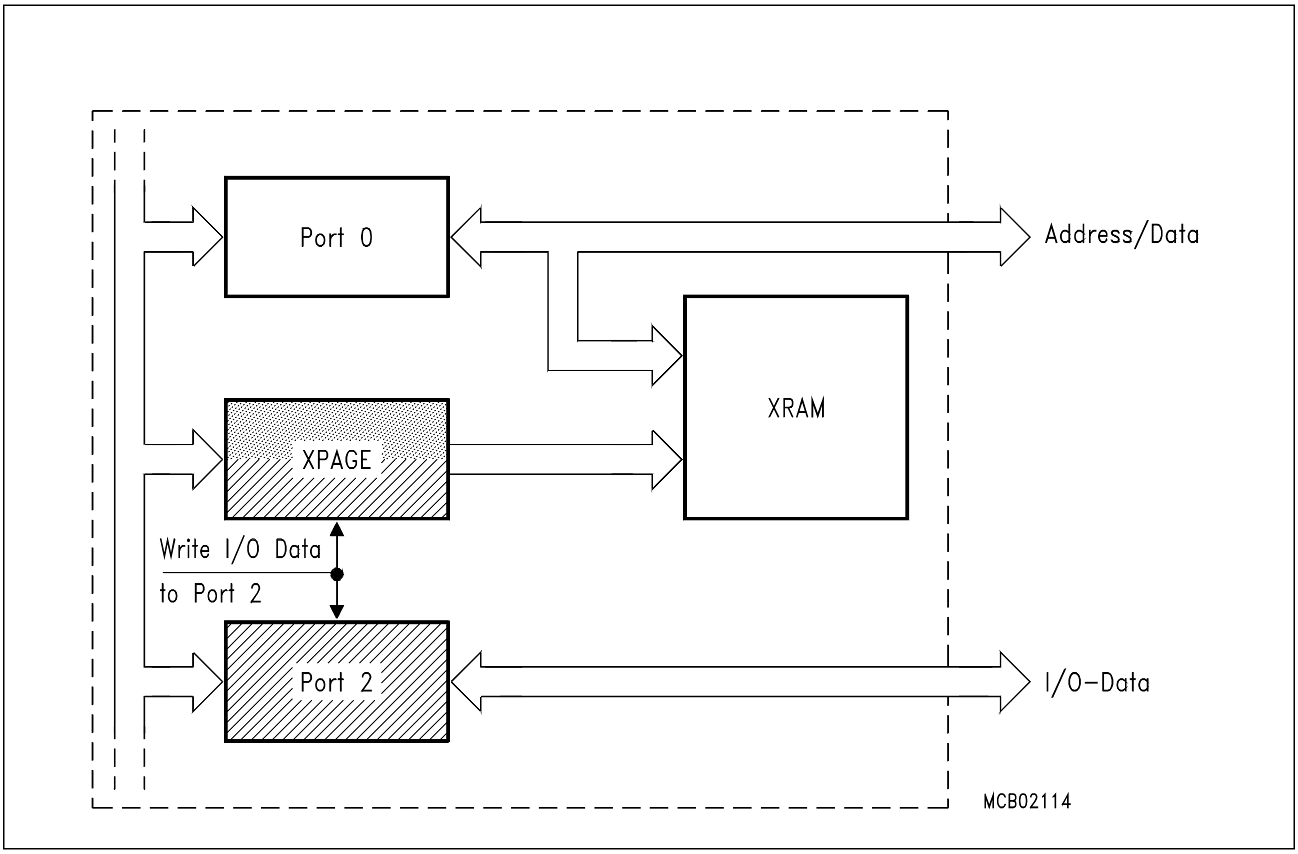


Figure 3-4
Use of Port 2 as I/O Port

At a write to port 2, XRAM address in XPAGE register will be overwritten because of the concurrent write to port 2 and XPAGE register. So whenever XRAM is used and the XRAM address differs from the byte written to port 2 latch it is absolutely necessary to rewrite XPAGE with page address.

Example :

I/O data at port 2 shall be 0AAH. A byte shall be fetched from XRAM at address 0F830H.

```
MOV     R0, #30H           ;
MOV     P2, #0AAH          ; P2 shows 0AAH
MOV     XPAGE, #0F8H       ; P2 still shows 0AAH but XRAM is addressed
MOVX    A, @R0             ; the contents of XRAM at 0F830H is moved to accumulator
```

The register XPAGE provides the upper address byte for accesses to XRAM with MOVX @Ri instructions. If the address formed by XPAGE and Ri points outside the XRAM address range (XPAGE ≠ XCON), then an external access is performed. For the SAB-C502 the contents of XPAGE must be equal to the content of SFR XCON in order to use the XRAM. Of course, the XRAM must be enabled if it shall be used with MOVX @Ri instructions.

Thus, the register XPAGE is used for addressing of the XRAM; additionally its contents are used for generating the internal XRAM select. If the contents of XPAGE is higher or less than the XRAM address range (XCON) then an external bus access is performed where the upper address byte is provided by P2 and not by XPAGE.

Therefore, the software has to distinguish two cases, if the MOVX @Ri instructions with paging shall be used :

a) Access to XRAM : The upper address byte must be written to XPAGE or P2 (content of XPAGE has to be equal to the content of XCON); both writes select the XRAM address range.

b) Access to external memory : The upper address byte must be written to P2; XPAGE will be loaded with the same address in order to deselect the XRAM.

3.3.4 Control of XRAM

There are two control bits in register STSCON which control the use and the bus operation during accesses to the additional on-chip RAM (XRAM).

Special Function Register SYSCON (Address 0B1$_H$)

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
0B1$_H$	–	–	–	–	–	–	XMAP1	XMAP0	SYSCON

Bit	Function
XMAP1	Control bit for $\overline{RD}$/$\overline{WR}$ signals during accesses to XRAM; this bit has no effect if XRAM is disabled (XMAP0 = 1) or if addresses exceeding the XRAM address range are used for MOVX accesses. XMAP1 = 0 : The signals $\overline{RD}$ and $\overline{WR}$ are not activated during accesses to XRAM. XMAP1 = 1 : The signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM.
XMAP0	Global enable/disable bit for XRAM memory. XMAP0 = 0 : The access to XRAM (= on-chip XDATA memory) is enabled. XMAP0 = 1 : The access to XRAM is disabled. All MOVX accesses are performed by the external bus (reset state).

Reset value of SYSCON is XXXX XX01$_B$.

The control bit XMAP0 is a global enable/disable bit for the additional on-chip RAM (XRAM). If this bit is set, the XRAM is disabled, all MOVX accesses use external memory via the external bus. In this case the SAB-C502 does not use the additional on-chip RAM and is compatible with the types without XRAM.

XMAP0 is hardware protected by an unsymetric latch. An unintentional disabling of XRAM could be dangerous since indeterminate values would be read from external bus. To avoid this the XMAP1 bit is forced to ‘1’ only by reset. Additionally, during reset an internal capacitor is loaded. So after reset state XRAM is disabled. Because of the load time of the capacitor XMAP0 bit once written to ‘0’ (that is, discharging capacitor) cannot be set to ‘1’ again by software. On the other hand any distortion (software hang-up, noise, ...) is not able to load this capacitor, too. That is, the stable status is XRAM enabled. The only way to disable XRAM after it was enabled is a reset.

The clear instruction for XMAP0 should be integrated in the program initialization routine before XRAM is used. In extremely noisy systems the user may have redundant clear instructions.

The control bit XMAP1 is relevant only if the XRAM is accessed. In this case the external $\overline{WR}$ and $\overline{RD}$ signals at P3.6 and P3.7 are not activated during the access, if XMAP1 is cleared. For debug purposes it might be useful to have these signals and the addresses at port 0 and 2 available. This is performed if XMAP1 is set.

The behaviour of P0 and P2 during a MOVX access depends on the control bits in register SYSCON and on the state of pin $\overline{\text{EA}}$. The table 1 lists the various operating conditions. It shows the following characteristics :

a) Use of P0 and P2 pins during the MOVX access.

Bus : The pins work as external address/data bus. If (internal) XRAM is accessed, the data written to the XRAM can be seen on the bus in debug mode.

I/O : The pins work as input/output lines under control of their latch.

b) Activation of the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pin during the access.

c) Use of internal or external XDATA memory.

The standard areas describe the standard operation as each 80C51 device without on-chip XRAM behaves.

Table 1, Behaviour of P0/P2 and $\overline{RD}$/$\overline{WR}$ during MOVX accesses

		$\overline{EA}$ = 0			$\overline{EA}$ = 1		
		XMAP1, XMAP0			**XMAP1, XMAP0**		
		00	**10**	**X1**	**00**	**10**	**X1**
MOVX @DPTR	DPTR outside XRAM address range (DPH ≠ XCON)	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used
	DPTR within XRAM address range (DPH = XCON)	a) P0/P2 ➔ Bus ($\overline{WR}$-Data only) b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0/P2 ➔ Bus ($\overline{WR}$-Data only) b) $\overline{RD}$/$\overline{WR}$ active c) XRAM is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0/P2 ➔ Bus ($\overline{WR}$-Data only) b) $\overline{RD}$/$\overline{WR}$ active c) XRAM is used	a) P0/P2 ➔ Bus b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used
MOVX @Ri	XPAGE outside XRAM addr. page range (XPAGE ≠ XCON)	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used
	XPAGE within XRAM addr. page range (XPAGE = XCON)	a) P0 ➔ Bus ($\overline{WR}$-Data only) P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0 ➔ Bus ($\overline{WR}$-Data only) P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) XRAM is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used	a) P0/P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0 ➔ Bus ($\overline{WR}$-Data only) P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) XRAM is used	a) P0 ➔ Bus P2 ➔ I/O b) $\overline{RD}$/$\overline{WR}$ active c) ext. memory is used

▨ modes compatible to the standard 8051-family

3.4 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general purpose registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word, PSW.3 and PSW.4, select the active register bank (see description of the PSW). This allows fast context switching, which is useful when entering subroutines or interrupt service routines.

The 8 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 register to address internal or external memory (e.g. MOV @R0).

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

3.5 Special Function Registers

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

The 36 special function register (SFR) include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. There are also 128 directly addressable bits within the SFR area.

All SFRs are listed in **table 3-1**, **table 3-2** and **table 3-3**.

In **table 3-1** they are organized in numeric order of their addresses. In **table 3-2** they are organized in groups which refer to the functional blocks of the SAB-C502. **Table 3-3** illustrates the contents of the SFRs.

Table 3-1
Special Function Registers in Numeric Order of their Addresses

Address	Register	Contents after Reset	Address	Register	Contents after Reset
80_H	**P0** [1]	**FF_H**	**98_H**	**SCON** [1]	**00_H**
81_H	SP	07_H	99_H	SBUF	XX_H [2]
82_H	DPL	00_H	$9A_H$	reserved	XX_H [2]
83_H	DPH	00_H	$9B_H$	reserved	XX_H [2]
84_H	reserved	XX_H	$9C_H$	reserved	XX_H [2]
85_H	reserved	XX_H	$9D_H$	reserved	XX_H [2]
86_H	WDTREL	00_H	$9E_H$	reserved	XX_H [2]
87_H	PCON	$000X0000_B$ [2]	$9F_H$	reserved	XX_H [2]
88_H	**TCON** [1]	**00_H**	**$A0_H$**	**P2** [1]	**$0FF_H$**
89_H	TMOD	00_H	$A1_H$	reserved	XX_H [2]
$8A_H$	TL0	00_H	$A2_H$	reserved	XX_H [2]
$8B_H$	TL1	00_H	$A3_H$	reserved	XX_H [2]
$8C_H$	TH0	00_H	$A4_H$	reserved	XX_H [2]
$8D_H$	TH1	00_H	$A5_H$	reserved	XX_H [2]
$8E_H$	reserved	XX_H [2]	$A6_H$	reserved	XX_H [2]
$8F_H$	reserved	XX_H [2]	$A7_H$	reserved	XX_H [2]
90_H	**P1** [1]	**FF_H**	**$A8_H$**	**IE** [1]	**$0X000000_B$** [2]
91_H	XPAGE	00_H	$A9_H$	reserved	XX_H [2]
92_H	DPSEL	$XXXXX000_B$ [2]	AA_H	SRELL	$D9_H$ [2]
93_H	reserved	XX_H [2]	AB_H	reserved	XX_H [2]
94_H	XCON	$F8_H$ [2]	AC_H	reserved	XX_H [2]
95_H	reserved	XX_H [2]	AD_H	reserved	XX_H [2]
96_H	reserved	XX_H [2]	AE_H	reserved	XX_H [2]
97_H	reserved	XX_H [2]	AF_H	reserved	XX_H [2]

1) : Bit-addressable Special Function Register
2) : X means that the value is undefined and the location is reserved

Table 3-1, Special Function Registers in Numeric Order of their Addresses (cont'd)

Address	Register	Contents after Reset	Address	Register	Contents after Reset
B0$_H$	**P3** [1]	**FF$_H$**	**D8$_H$**	**BAUD** [1]	**0XXXXXXXB**
B1$_H$	SYSCON	XXXXXX01$_B$ [2]	D9$_H$	reserved	XX$_H$
B2$_H$	reserved	XX$_H$ [2]	DA$_H$	reserved	XX$_H$
B3$_H$	reserved	XX$_H$ [2]	DB$_H$	reserved	XX$_H$ [2]
B4$_H$	reserved	XX$_H$ [2]	DC$_H$	reserved	XX$_H$ [2]
B5$_H$	reserved	XX$_H$ [2]	DD$_H$	reserved	XX$_H$ [2]
B6$_H$	reserved	XX$_H$ [2]	DE$_H$	reserved	XX$_H$ [2]
B7$_H$	reserved	XX$_H$ [2]	DF$_H$	reserved	XX$_H$ [2]
B8$_H$	**IP** [1]	**XX000000$_B$** [2]	**E0$_H$**	**ACC** [1]	**00$_H$**
B9$_H$	reserved	XX$_H$ [2]	E1$_H$	reserved	XX$_H$ [2]
BA$_H$	SRELH	XXXXXX11$_B$ [2]	E2$_H$	reserved	XX$_H$ [2]
BB$_H$	reserved	XX$_H$ [2]	E3$_H$	reserved	XX$_H$ [2]
BC$_H$	reserved	XX$_H$ [2]	E4$_H$	reserved	XX$_H$ [2]
BD$_H$	reserved	XX$_H$ [2]	E5$_H$	reserved	XX$_H$ [2]
BE$_H$	reserved	XX$_H$ [2]	E6$_H$	reserved	XX$_H$ [2]
BF$_H$	reserved	XX$_H$ [2]	E7$_H$	reserved	XX$_H$ [2]
C0$_H$	**WDCON** [1]	**XXXX0000$_B$** [2]	**E8$_H$**	reserved	**XX$_H$** [2]
C1$_H$	reserved	XX$_H$ [2]	E9$_H$	reserved	XX$_H$ [2]
C2$_H$	reserved	XX$_H$ [2]	EA$_H$	reserved	XX$_H$ [2]
C3$_H$	reserved	XX$_H$ [2]	EB$_H$	reserved	XX$_H$ [2]
C4$_H$	reserved	XX$_H$ [2]	EC$_H$	reserved	XX$_H$ [2]
C4$_H$	reserved	XX$_H$ [2]	ED$_H$	reserved	XX$_H$ [2]
C6$_H$	reserved	XX$_H$ [2]	EE$_H$	reserved	XX$_H$ [2]
C7$_H$	reserved	XX$_H$ [2]	EF$_H$	reserved	XX$_H$ [2]
C8$_H$	**T2CON** [1]	**00$_H$**	**F0$_H$**	**B** [1]	**00$_H$**
C9$_H$	T2MOD	XXXXXXX0$_B$ [2]	F1$_H$	reserved	XX$_H$ [2]
CA$_H$	RC2L	00$_H$	F2$_H$	reserved	XX$_H$ [2]
CB$_H$	RC2H	00$_H$	F3$_H$	reserved	XX$_H$ [2]
CC$_H$	TL2	00$_H$	F4$_H$	reserved	XX$_H$ [2]
CD$_H$	TH2	00$_H$	F5$_H$	reserved	XX$_H$ [2]
CE$_H$	reserved	XX$_H$ [2]	F6$_H$	reserved	XX$_H$ [2]
CF$_H$	reserved	XX$_H$ [2]	F7$_H$	reserved	XX$_H$ [2]
D0$_H$	**PSW** [1]	**00$_H$**	**F8$_H$**	reserved	**XX$_H$** [2]
D1$_H$	reserved	XX$_H$ [2]	F9$_H$	reserved	XX$_H$ [2]
D2$_H$	reserved	XX$_H$ [2]	FA$_H$	reserved	XX$_H$ [2]
D3$_H$	reserved	XX$_H$ [2]	FB$_H$	reserved	XX$_H$ [2]
D4$_H$	reserved	XX$_H$ [2]	FC$_H$	reserved	XX$_H$ [2]
D5$_H$	reserved	XX$_H$ [2]	FD$_H$	reserved	XX$_H$ [2]
D6$_H$	reserved	XX$_H$ [2]	FE$_H$	reserved	XX$_H$ [2]
D7$_H$	reserved	XX$_H$ [2]	FF$_H$	reserved	XX$_H$ [2]

[1] : Bit-addressable Special Function Register

[2] : X means that the value is undefined and the location is reserved

Table 3-2, 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$_H$
	DPSEL	Data Pointer Select Register	92$_H$	XXXXX000$_B$ [3]
	PSW	Program Status Word Register	**D0$_H$** [1]	00$_H$
	SP	Stack Pointer	81$_H$	07$_H$
Interrupt System	IE	Interrupt Enable Register	**A8$_H$** [1]	0X000000$_B$ [3]
	IP	Interrupt Priority Register	**B8$_H$** [1]	XX000000$_B$ [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$_H$
XRAM	XPAGE	Page Address Register for XRAM	91$_H$	00$_H$
	XCON	XRAM StartAddress (High Byte)	94$_H$	F8$_H$
	SYSCON	XRAM Control Register	B1$_H$	XXXXXX01$_B$ [3]
Serial Channels	PCON [2]	Power Control Register	87$_H$	000X 0000$_B$ [3]
	SBUF	Serial Channel Buffer Register	99$_H$	XX$_H$ [3]
	SCON	Serial Channel 0 Control Register	**98$_H$** [1]	00$_H$
	SRELL	Baudrate Generator reload value, low Byte	AA$_H$	D9$_H$
	SRELH	Baudrate Generator reload value, high Byte	BA$_H$	XXXXXX11$_B$ [3]
	BAUD	Baudrate Generator Enable Bit	**D8$_H$** [1]	0XXXXXXX$_B$ [3]
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$_H$
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$_H$	000X0000$_B$ [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

Table 3-3
Contents of SFRs, SFRs in Numeric Order

Address	Register	Bit 7	6	5	4	3	2	1	0
80_H	P0								
81_H	SP								
82_H	DPL								
83_H	DPH								
86_H	WDTREL								
87_H	PCON	SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE
88_H	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89_H	TMOD	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
$8A_H$	TL0								
$8B_H$	TL1								
$8C_H$	TH0								
$8D_H$	TH1								
90_H	P1								
91_H	XPAGE								
92_H	DPSEL	–	–	–	–	–	.2	.1	.0
94_H	XCON								
98_H	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99_H	SBUF								
$A0_H$	P2								
$A8_H$	IE	EA	–	ET2	ES	ET1	EX1	ET0	EX0
AA_H	SRELL								

SFR bit and byte addressable

SFR not bit addressable

▨ must not be used

– : = bit location is reserved

Table 3-3
Contents of SFRs, SFRs in Numeric Order (cont'd)

Address	Register	Bit 7	6	5	4	3	2	1	0
$B0_H$	P3								
$B1_H$	SYSCON	–	–	–	–	–	–	XMAP1	XMAP0
$B8_H$	IP	–	PADC	PT2	PS	PT1	PX1	PT0	PX0
BA_H	SRELH								
$C0_H$	WDCON	–	–	–	–	0WDS	WDTS	WDT	SWDT
$C8_H$	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$
$C9_H$	T2MOD	–	–	–	–	–	–	–	DCEN
CA_H	RC2L								
CB_H	RC2H								
CC_H	TL2								
CD_H	TH2								
$D0_H$	PSW	CY	AC	F0	RS1	RS0	OV	F1	P
$D8_H$	BAUD	BD	–	–	–	–	–	–	–
$E0_H$	ACC								
$F0_H$	B								

□□□□□□□□ SFR bit and byte addressable

⬚⬚⬚⬚⬚⬚⬚⬚ SFR not bit addressable

▨ must not be used

– : = bit location is reserved

4 External Bus Interface

The SAB-C502 allows for external memory expansion. To accomplish this, the external bus interface common to most 8051-based controllers is employed.

4.1 Accessing External Memory

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

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

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 $0FF_H$ 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 have to contain 1s, and the contents of the port 2 SFR are not modified.

If an 8-bit address is used (MOVX @Ri), the contents of the port 2 SFR remain at the port 2 pins throughout the external memory cycle. This will facilitate paging. It should be noted that, if a port 2 pin outputs an address bit that is a 1, strong pullups will be used for the entire read/write cycle and not only for two oscillator periods.

Timing

The timing of the external bus interface, in particular the relationship between the control signals ALE, $\overline{\text{PSEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$ and information on port 0 and port 2, is illustated in **figure 4-1 a)** and **b)**.

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

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

External Program Memory Access

The external program memory is accessed under two conditions:

– whenever signal $\overline{\text{EA}}$ is active: or
– whenever the program counter (PC) contains a number that is larger than $3FFF_H$.

This requires the ROM-less version SAB-C502-L to have $\overline{\text{EA}}$ wired low to allow the lower 16K 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 is not affected. During external program memory fetches port 2 lines output the high byte of the PC, and during accesses to external data memory they output either DPH or the port 2 SFR (depending on whether the external data memory access is a MOVX @DPTR or a MOVX @Ri).

Since the SAB-C502-L has no internal program memory, accesses to program memory are always external, and port 2 is at all times dedicated to output the high-order address byte. This means that port 0 and port 2 of the SAB-C502-L can never be used as general-purpose I/O. This also applies to the SAB-C502-2R when it is operated with only an external program memory.

4.2 $\overline{\text{PSEN}}$, Program Store Enable

The read strobe for external fetches is $\overline{\text{PSEN}}$. $\overline{\text{PSEN}}$ is not activated for internal fetches. When the CPU is accessing external program memory, $\overline{\text{PSEN}}$ is activated twice every cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{\text{PSEN}}$ is activated its timing is not the same as for $\overline{\text{RD}}$. A complete $\overline{\text{RD}}$ cycle, including activation and deactivation of ALE and $\overline{\text{RD}}$, takes 12 oscillator periods. A complete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$ takes 6 oscillator periods. The execution sequence for these two types of read cycles is shown in **figure 4-1 a)** and **b)**.

4.3 ALE, Address Latch Enable

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 into an external latch during fetches from external memory. The address byte is valid at the negative transition of ALE. For that purpose, ALE is activated twice every machine cycle. This activation takes place even if the cycle involves no external fetch. The only time no ALE pulse comes out is during an access to external data memory when $\overline{\text{RD}}$/$\overline{\text{WR}}$ 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.

4.4 Overlapping External Data and Program Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is used for storing data. In the SAB-C502 the external program and data memory spaces can be combined by AND-ing $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive logic AND of these two signals produces an active low read strobe that can be used for the combined physical memory. Since the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory needs to be fast enough to adapt to the $\overline{\text{PSEN}}$ cycle.

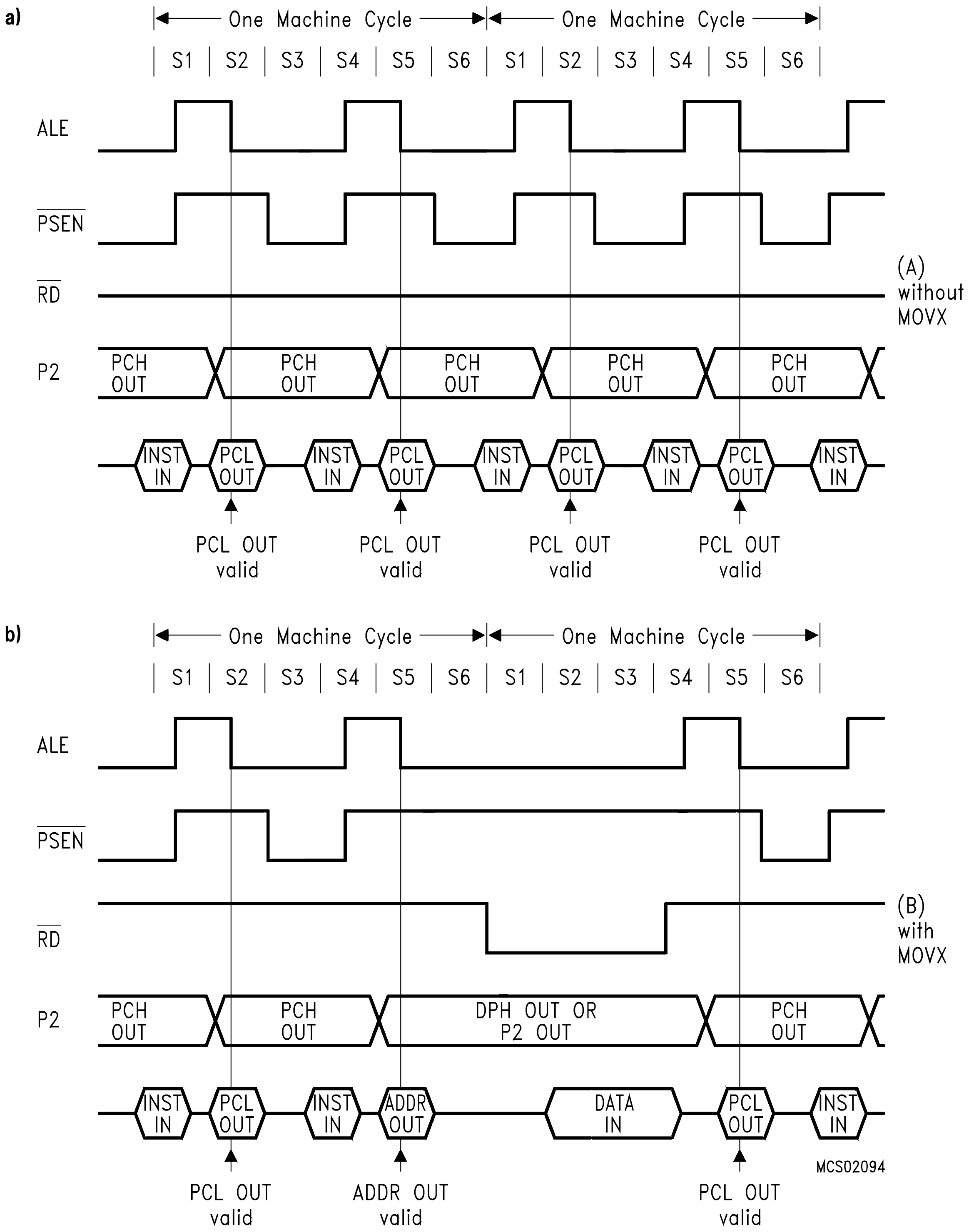


Figure 4-1 a) and b)
External Program Memory Execution

5 System Reset

5.1 Hardware Reset

The hardware reset function incorporated in the SAB-C502 allows for an easy automatic start-up at a minimum of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation in order to restart the device. This is particularly done when the power-down mode is to be terminated.

Additionally to the hardware reset, which is applied externally to the SAB-C502, there are two internal reset sources, the watchdog timer and the oscillator watchdog. The chapter at hand only deals with the external hardware reset.

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 one machine cycle (12 oscillator periods) while the oscillator is running. With the oscillator running the internal reset is executed during the second machine cycle and is repeated every cycle until RESET goes low again.

During reset, 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 test modes which are reserved for test purposes. This in turn may cause unpredictable output operations at several port pins).

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

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

The contents of the internal RAM of the SAB-C502 is not affected by a reset. After power-up the contents is undefined, while it remains unchanged during a reset if the power supply is not turned off.

5.2 Fast Internal Reset after Power-On

The SAB-C502 uses the oscillator watchdog unit for a fast internal reset procedure after power-on.

Figure 5-1 shows the power-on sequence under control of the oscillator watchdog.

Normally the devices of the 8051 family (e.g. SAB 80C515) 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 in order 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. 1 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 SAB-C502 the oscillator watchdog unit avoids this situation. In this case, after power-on the oscillator watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 microseconds). In the following 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 also all ports to the defined state (**see figure 5-1**). The time period from power-on until reaching the reset state at the ports derives from the following terms:

–	RC oscillator start-up	< 2 µs
–	synchronization of the RC oscillators divider-by-5	< 6 T
–	synchronization of the state and cycle counters	< 6 T
–	reset procedure till correct port states are reached	< 12 T

Delay between power-on and correct reset state:

Typ.: 18 µs

Max.: 34 µs

After the on-chip oscillator finally has started, the oscillator watchdog detects the correct function; then the watchdog still holds the reset active for a time period of 768 cycles of the RC oscillator in order to allow the oscillation of the on-chip oscillator to stabilize (**figure 5-1, II**). Subsequently the clock is supplied by the on-chip oscillator and the oscillator watchdog's reset request is released (**figure 5-1, III**). However, an externally applied reset still remains active (**figure 5-1, IV**) and the device does not start program execution (**figure 5-1, 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. The reasons are as follows:

- Termination of Software Power-Down Mode
- Reset of the status flag OWDS that is set by the oscillator watchdog during the power up sequence.

The external reset signal must be hold active at least until the on-chip oscillator has started and the internal watchdog reset phase is completed. An external reset time of more than 50 µs should be sufficient in typical applications. If only a capacitor at pin RESET is used a value of less than 100 nF provides the desired reset time.

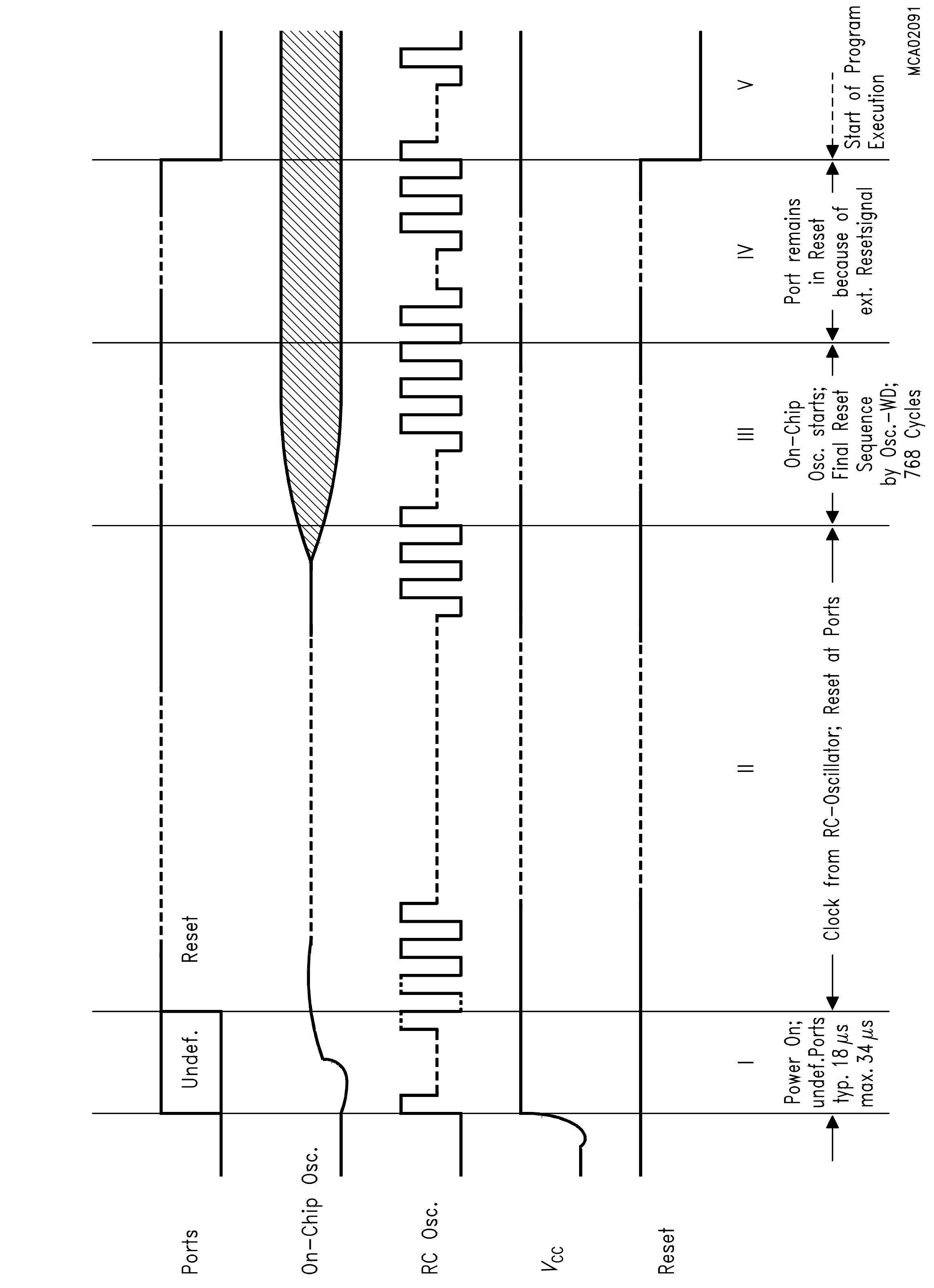

Ports
On−Chip Osc.
RC Osc.
V_{CC}
Reset
Undef.
Reset
I
II
III
IV
V
Power On; undef.Ports typ. 18 µs max. 34 µs
Clock from RC−Oscillator; Reset at Ports
On−Chip Osc. starts; Final Reset Sequence by Osc.−WD; 768 Cycles
Port remains in Reset because of ext. Resetsignal
Start of Program Execution
MCA02091

Figure 5-1
Power-On of the SAB-C502

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 happens in state 5 phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (high level) the internal reset procedure is started. It needs two complete machine cycles to put the complete device to its correct reset state, i.e. all special function registers contain their default values, the port latches contain 1's etc. 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 one machine cycle; after this time the SAB-C502 remains in its reset state as long as the signal is active. When the signal goes inactive this transition is recognized in the following state 5 phase 2 of the machine cycle. Then the processor starts its address output (when configured for external ROM) in the following state 5 phase 1. One phase later (state 5 phase 2) the first falling edge at pin ALE occurs.

Figure 5-2 shows this timing for a configuration with $\overline{EA}$ = 0 (external program memory). Thus, between the release of the RESET signal and the first falling edge at ALE there is a time period of at least one machine cycle but less than two machine cycles.

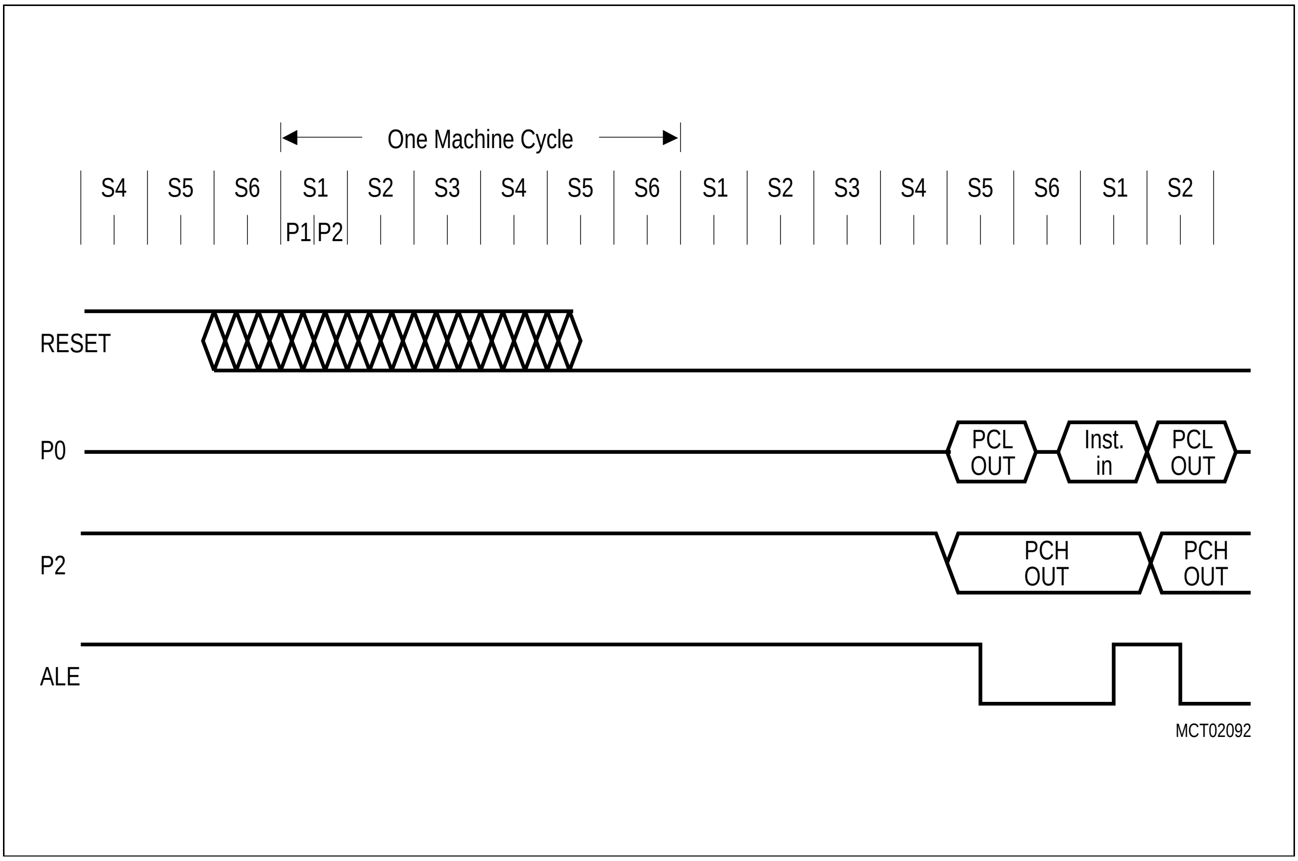


Figure 5-2
CPU Timing after Reset

6 On-Chip Peripheral Components

I/O Ports

The SAB-C502 has three 8-bit I/O ports and one 8-bit input port. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 3 are quasi-bidirectional I/O ports with internal pullup resistors. That means, when configured as inputs, ports 1 to 3 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

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

6.1 Parallel I/O

6.1.1 Port Structures

Digital I/O

The SAB-C502 allows for digital I/O on 32 lines grouped into 4 bidirectional 8-bit ports. Each port bit consists of a latch, an output driver and an input buffer. Read and write accesses to the I/O ports P0, P1, P2 and P3 are performed via their corresponding special function registers.

Digital I/O Port Circuitry

Figure 6-1 shows a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the 4 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. The Q output of the flip-flop is placed on the internal bus in response to a "read-latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read-pin" signal from the CPU. Some instructions that read from a port (i.e. from the corresponding port SFR P0 to P3) activate the "read-latch" signal, while others activate the "read-pin" signal.

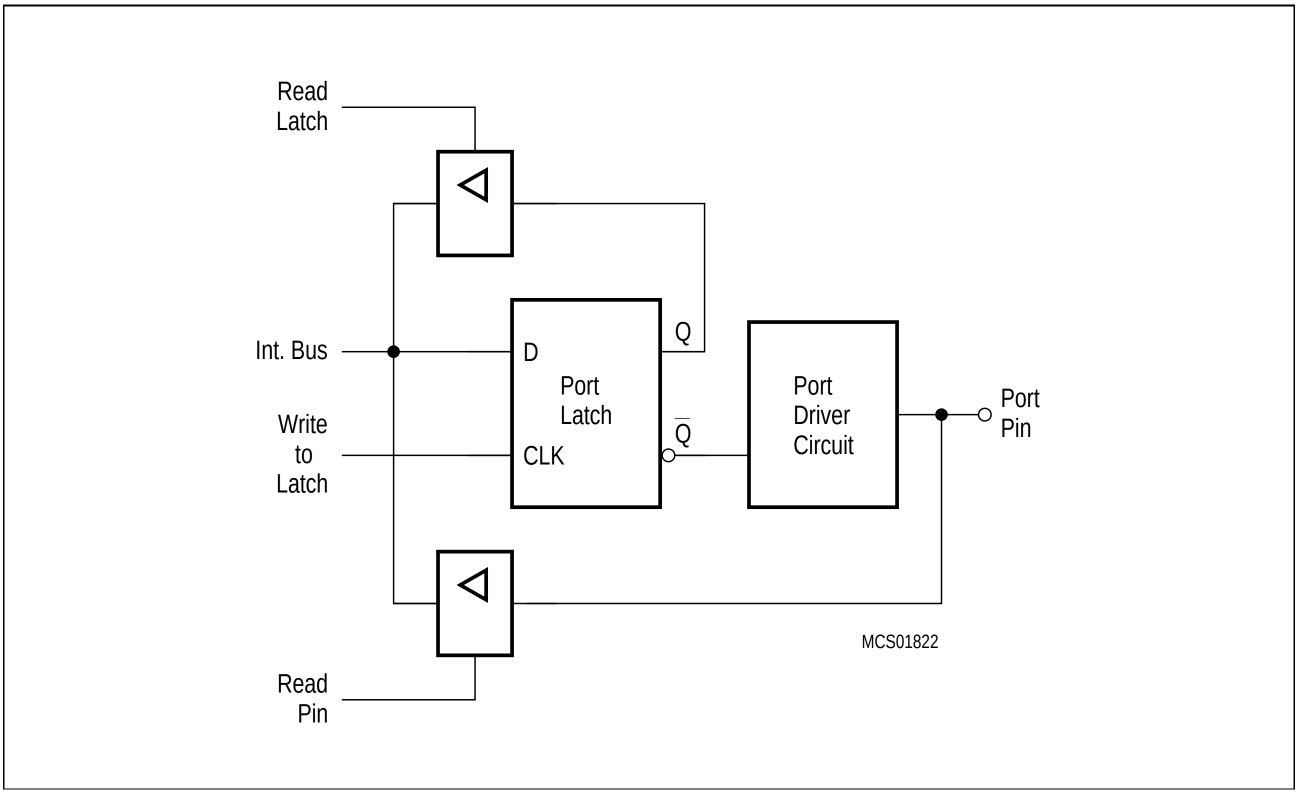


Figure 6-1
Basic Structure of a Port Circuitry

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

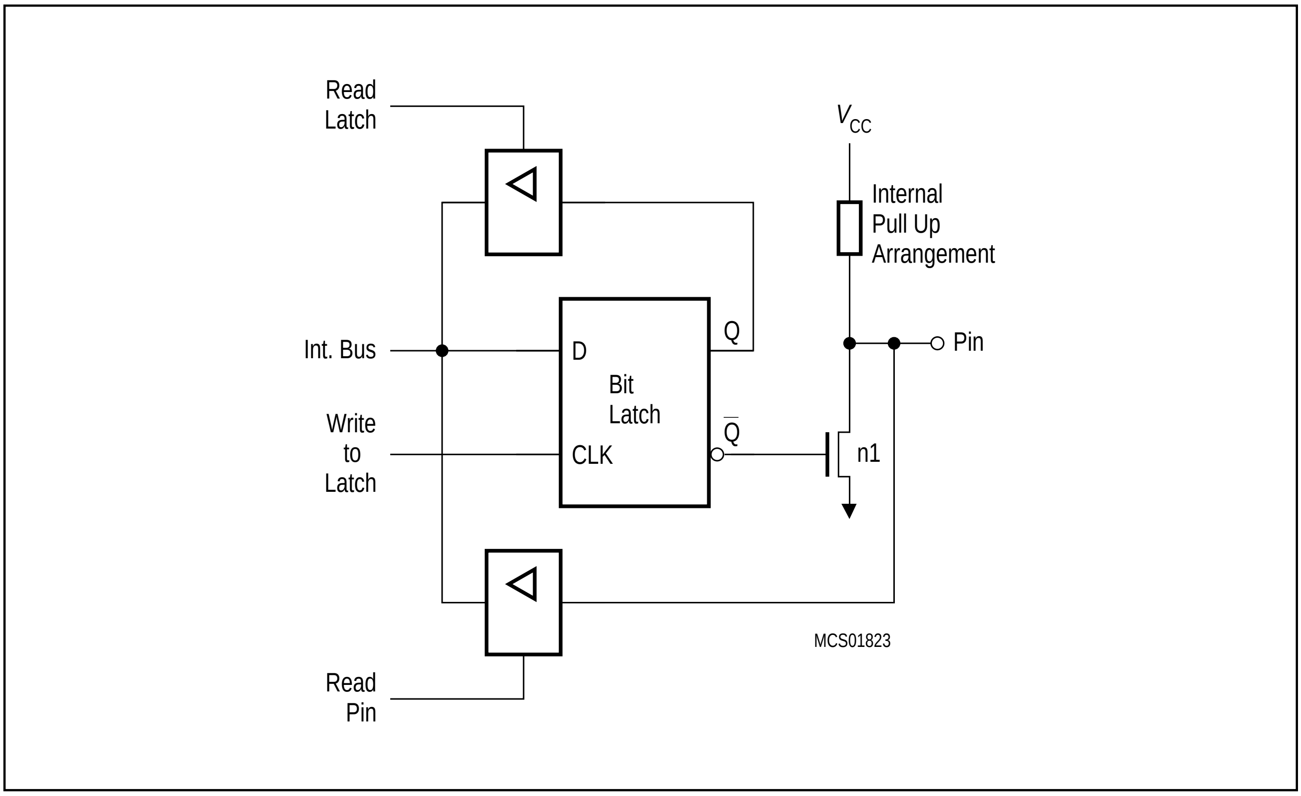


Figure 6-2
Basic Output Driver Circuit of Ports 1, 2 and 3

In fact, the pullups mentioned before and included in **figure 6-2** are pullup arrangements shown in **figure 6-3**. One n-channel pulldown FET and three pullup FETs are used:

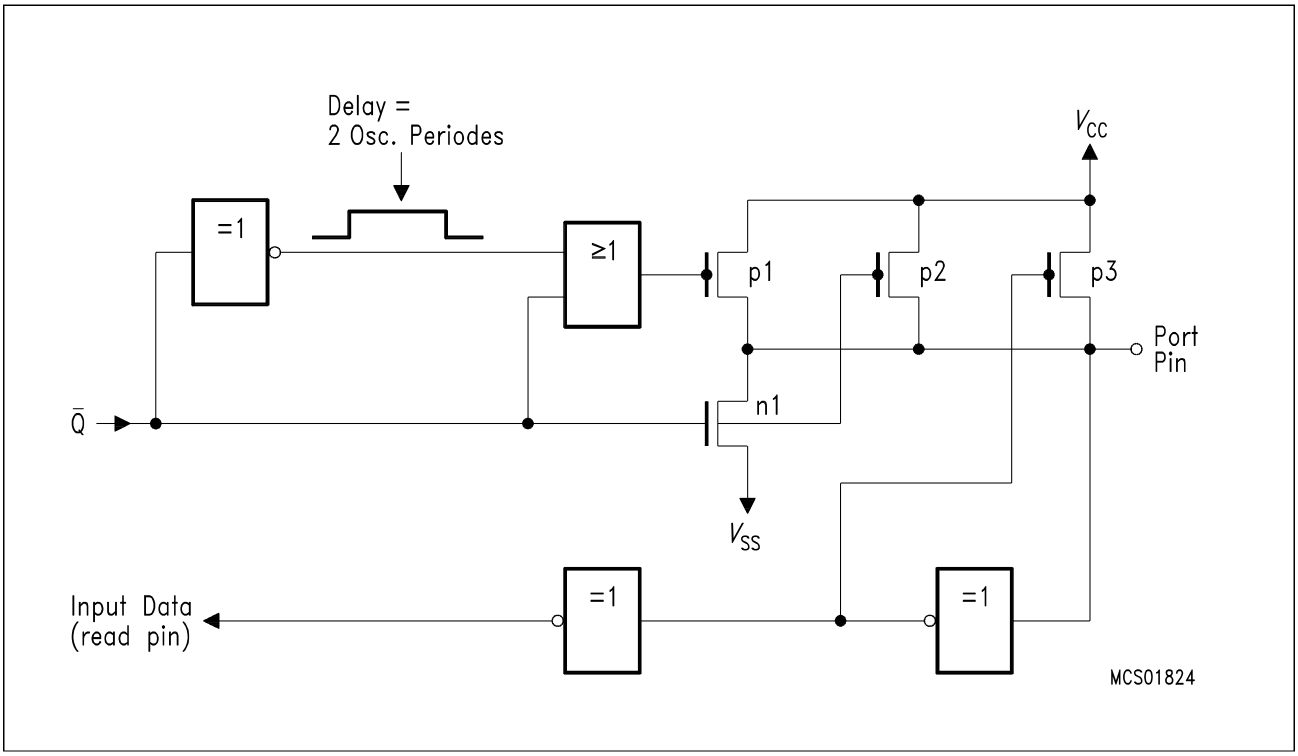


Figure 6-3
Output Driver Circuit of Ports 1, 2 and 3

- The **pulldown FET n1** is of n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a "0" is programmed to the port pin. A short circuit to V_{CC} must be avoided if the transistor is turned on, since the high current might destroy the FET.
- The **pullup FET p1** is of p-channel type. It is activated for two oscillator periods (S1P1 and S1P2) if a 0-to-1 transition is programmed to the port pin, i.e. a "1" is programmed to the port latch which contained a "0". The extra pullup can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.
- The **pullup FET p2** is of p-channel type. It is always activated when a "1" is in the port latch, thus providing the logic high output level. This pullup FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.
- The **pullup FET p3** is of p-channel type. It is only activated if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pullup current if a logic high level shall be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level, e.g when used as input. In this configuration only the weak pullup FET p2 is active, which sources the current I_{IL} . If, in addition, the pullup FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, an additional power consumption can be avoided if port pins are used as inputs

with a low level applied. However, the driving capability is stronger if a logic high level is output.

The described activating and deactivating of the four different transistors translates into four states the pins can be:

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

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

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

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

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

The same is true if a pin is used as bidirectional line and the external circuitry is switched from outpout 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.

Port 0, in contrast to ports 1, 2 and 3, is considered as "true" bidirectional, because the port 0 pins float when configured as inputs. Thus, this port differs in not having internal pullups. The pullup FET in the P0 output driver (**see figure 6-4a**) is used only when the port is emitting 1 s during the external memory accesses. Otherwise, the pullup is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a "1" to the port latch leaves both output FETs off and the pin floats. In that condition it can be used as high-impedance input. If port 0 is configured as general I/O port and has to emit logic high-level (1), external pullups are required.

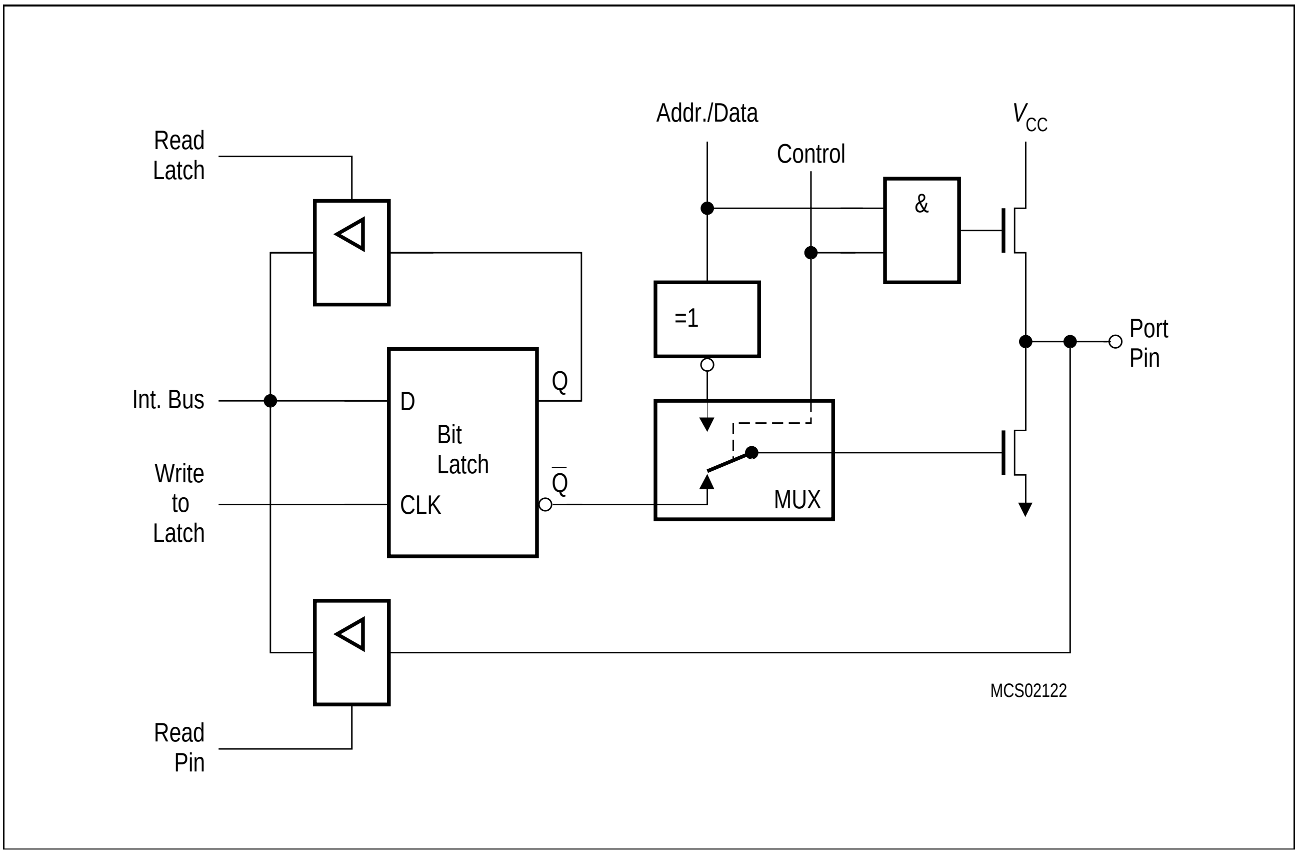


Figure 6-4a
Port 0 Circuitry

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

As shown in **figure 6-4a** and below in **figure 6-4b**, the output drivers of ports 0 and 2 can be switched to an internal address or address/data bus for use in external memory accesses. In this application they cannot be used as general purpose I/O, even if not all address lines are used externally. The switching is done by an internal control signal dependent on the input level at the $\overline{\text{EA}}$ 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 P0/P2 SFR remains unchanged. Being an address/data bus, port 0 uses a pullup FET as shown in **figure 6-4a**. 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.

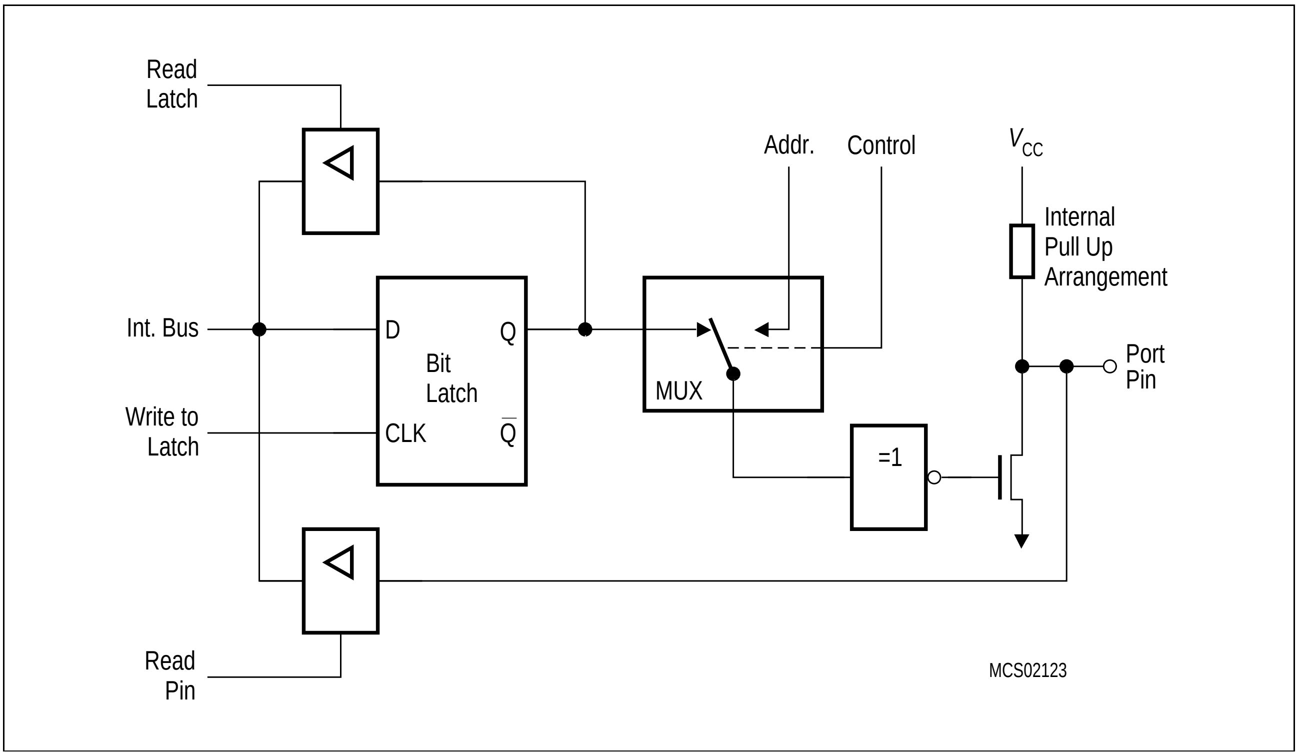


Figure 6-4b
Port 2 Circuitry

6.1.3 Alternate Functions

The pins of ports 1 and 3 are multifunctional. They are port pins and also serve to implement special features as listed in **table 6-1**.

Figure 6-5 shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise the pulldown FET is on and the port pin is stuck at 0. After reset all port latches contain ones (1).

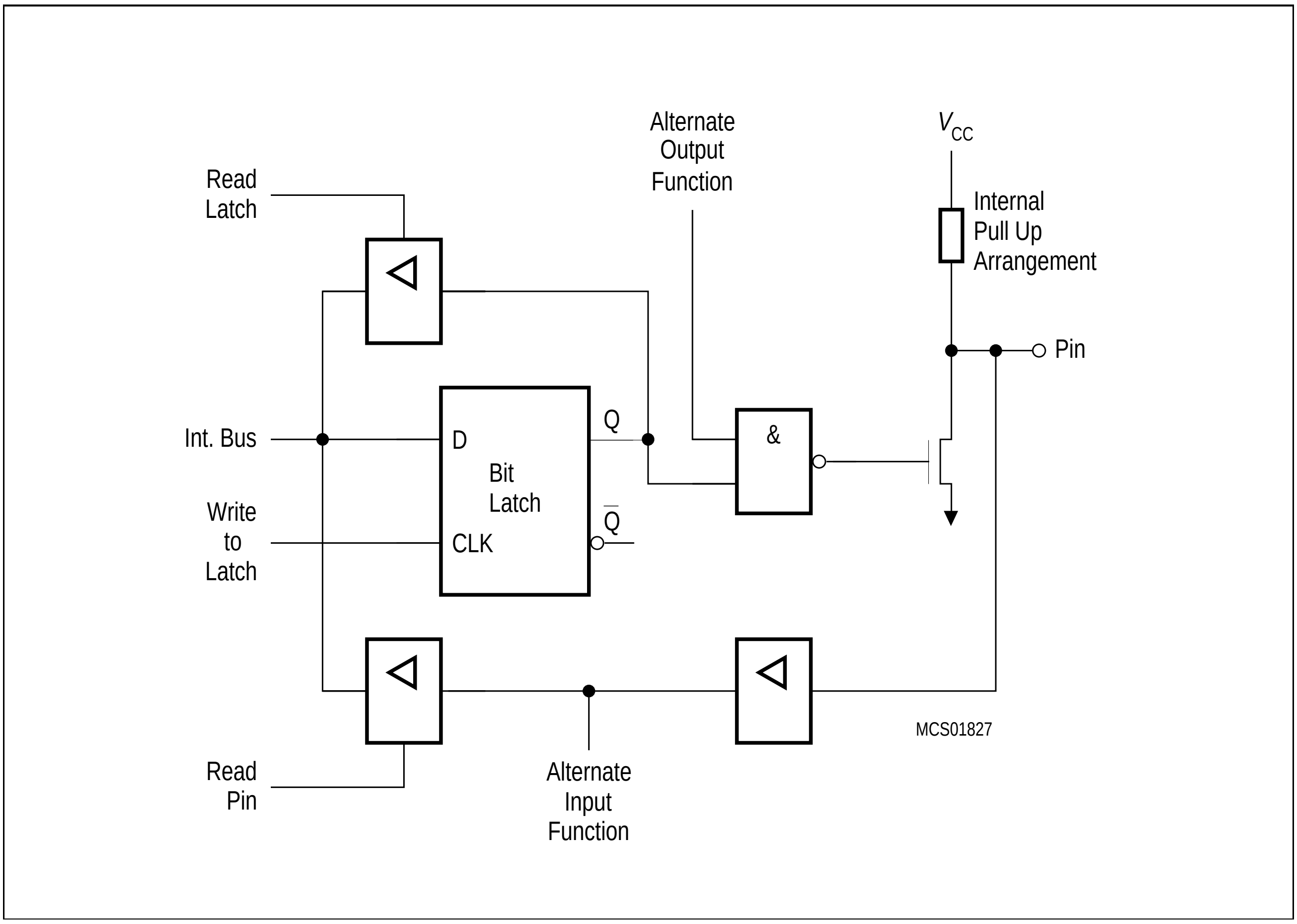


Figure 6-5
Ports 1 and 3

Ports 1 and 3 are provided for several alternate functions, as listed in **table 6-1**:

Table 6-1
Alternate Functions of Port 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
P3.0	RxD	Serial port’s receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TxD	Serial port’s transmitter data output (asynchronous) or data clock output (synchronous)
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 momory read strobe

6.1.4 Port Handling

6.1.4.1 Port Timing

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

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

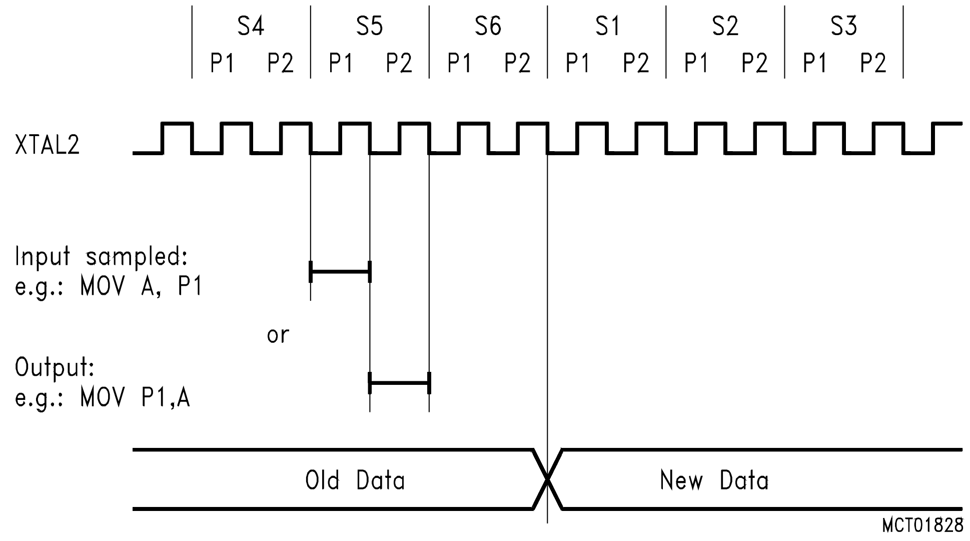


Figure 6-6
Port Timing

6.1.4.2 Port Loading and Interfacing

The output buffers of ports 1, 2 and 3 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can be be looked up in the DC characteristics in the Data Sheet of the SAB-C502. The corresponding parameters are V_{OL} and V_{OH}.

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

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

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

Some port-reading instructions read the latch and others read the pin (**see figure 6-1**). The instructions reading the latch rather than the pin read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write"- instructions, which are listed in **table 6-2**. 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. In any case, reading from latch or pin, resp., is performed by reading the SFR P0, P1, P2 and P3; for example, "MOV A, P3" reads the value from port 3 pins, while "ANL P3, #0AA$_H$" reads from the latch, modifies the value and writes it back to the latch.

It is not obvious that the last three instructions in **table 6-2** are read-modify-write instructions, but they are. The reason is that they read the port byte, all 8 bits, modify the addressed bit, then write the complete byte back to the latch.

Table 6-2
"Read-Modify-Write"-Instructions

Instruction	Function
ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic exclusive OR; e.g. XRL P3, A
JBC	Jump if bit is set and clear bit; e.g. JBC P1.1, LABEL
CPL	Complement bit; e.g. CPL P3.0
INC	Increment byte; e.g. INC P4
DEC	Decrement byte; e.g. DEC P5
DJNZ	Decrement and jump if not zero; e.g. 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

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

6.2 Timers/Counters

The SAB-C502 contains three 16-bit timers/counters which are useful in many applications for timing and counting.

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

In "counter" function, the register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 and P3.5, resp.). In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the 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. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

6.2.1 Timer/Counter 0 and 1

Timer / counter 0 and 1 of the SAB-C502 are fully compatible with timer / counter 0 and 1 of the SAB 8051 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; Timer/counter 1 in this mode holds its count. The effect is the same as setting TR1 = 0.

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

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

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

Special Function Register TCON (Address 88_H)

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

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

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

Special Function Register TMOD (Address 89_H)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
89_H	Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0	TMOD
	Timer 1				Timer 0				

Timer/counter 0/1 mode control register.

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	
0	0	8-bit timer/counter. "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler.
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops

6.2.1.1 Mode 0

Putting either timer/counter 0,1 into mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. **Figure 6-7** shows the mode 0 operation.

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

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

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

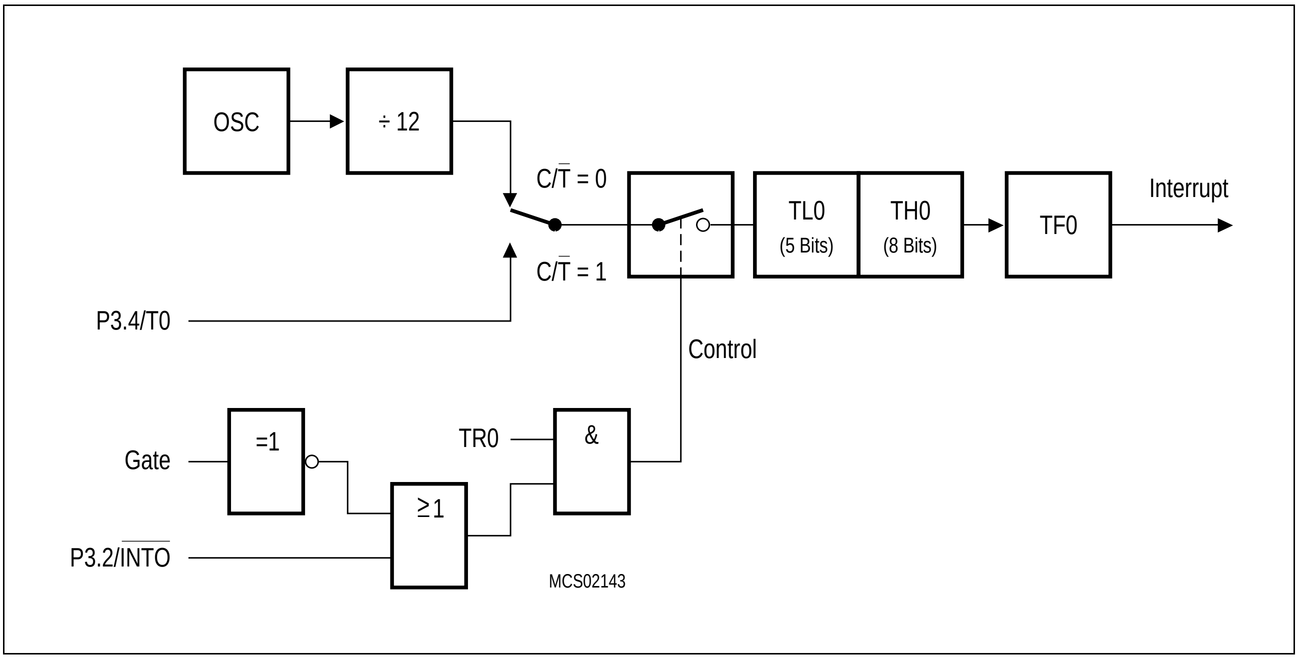

OSC
÷ 12
C/$\overline{\text{T}}$ = 0
C/$\overline{\text{T}}$ = 1
TL0
(5 Bits)
TH0
(8 Bits)
TF0
Interrupt
P3.4/T0
Control
Gate
=1
TR0
&
≥1
P3.2/$\overline{\text{INT0}}$
MCS02143

Figure 6-7
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

6.2.1.2 Mode 1

Mode 1 is the same as mode 0, except that the timer register is running with all 16 bits.

Mode 1 is shown in **figure 6-8**.

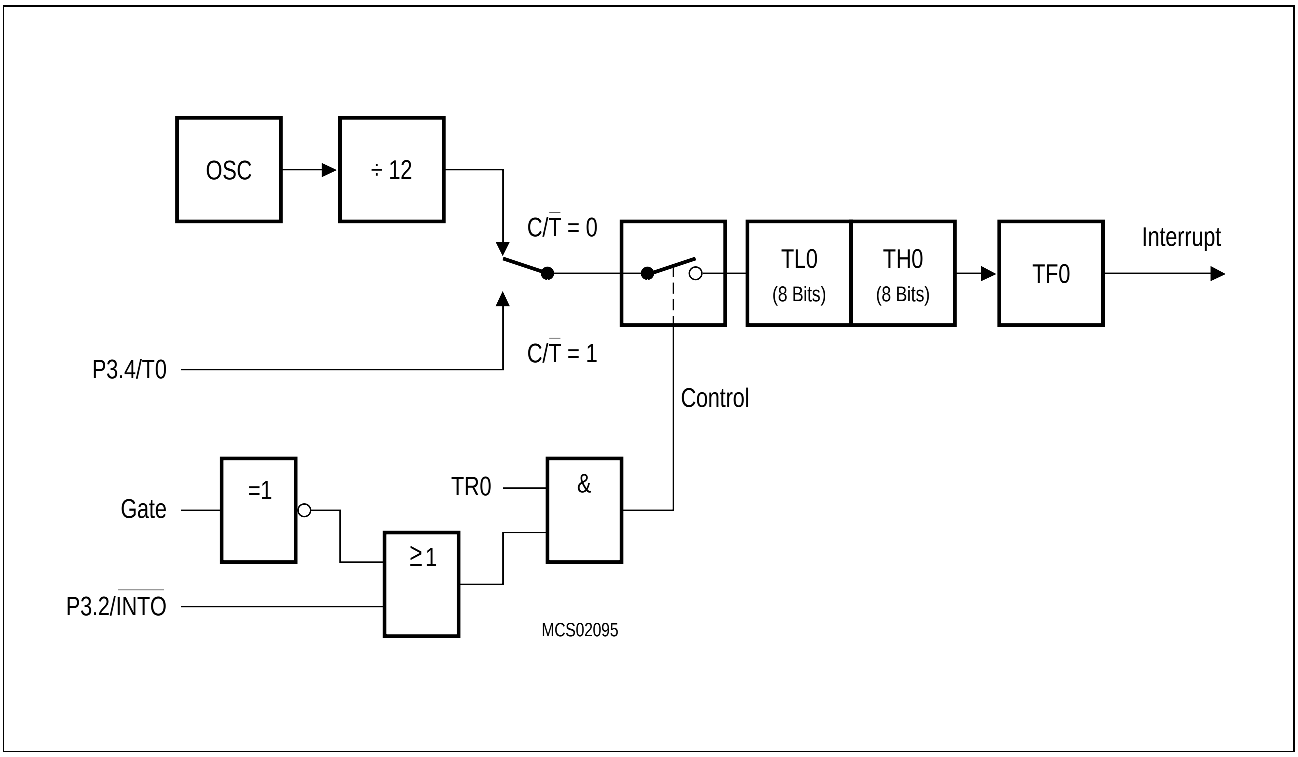


Figure 6-8
Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

6.2.1.3 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in **figure 6-9**. 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.

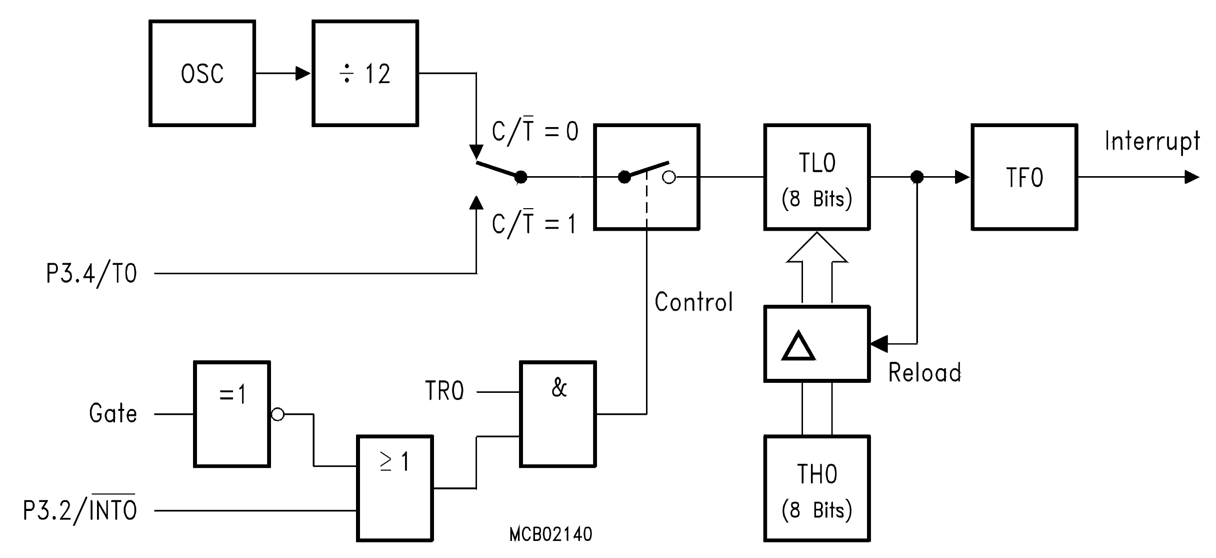


Figure 6-9
Timer/Counter 0,1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

6.2.1.4 Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1=0. Timer 0 in mode 3 establishes TL0 and TH0 as two seperate counters. The logic for mode 3 on timer 0 is shown in **figure 6-10**. TL0 uses the timer 0 control bits: $C/\overline{T}$, Gate, TR0, $\overline{INT0}$ and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

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

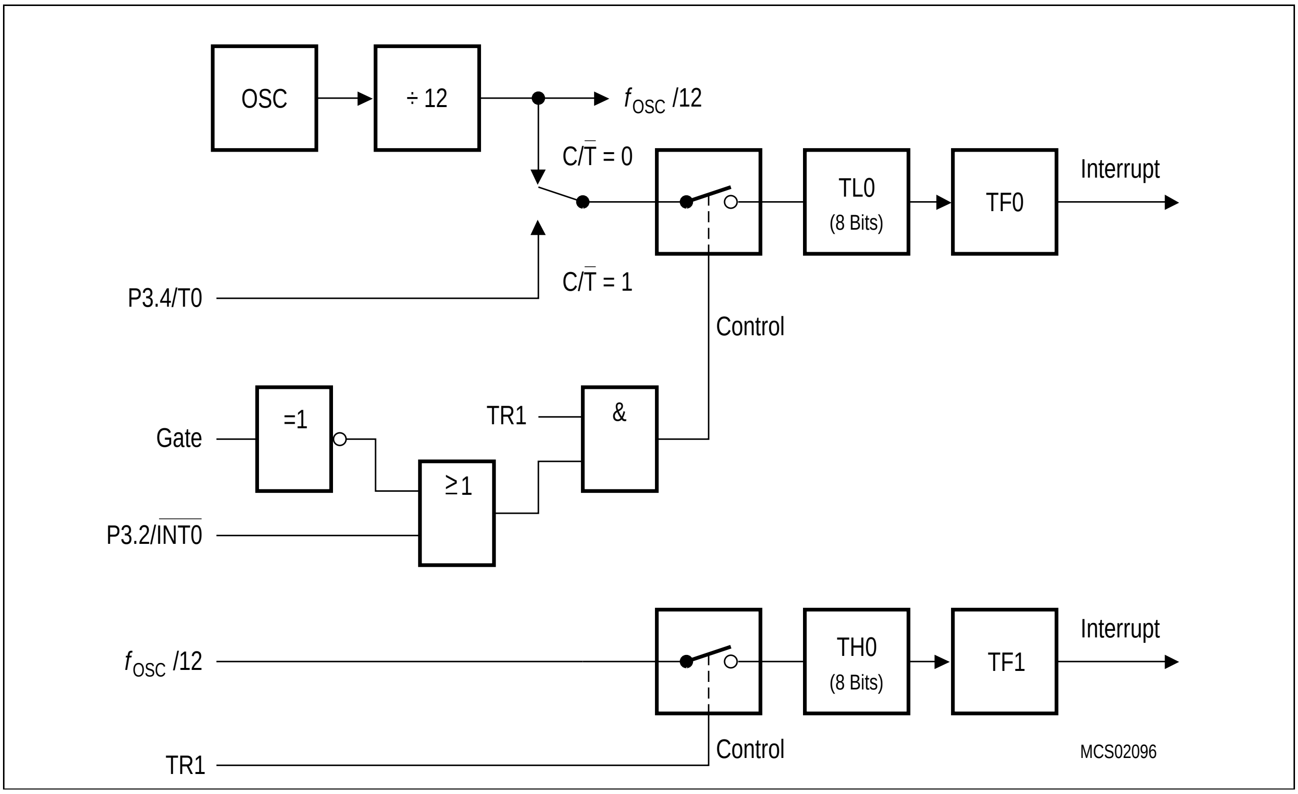


Figure 6-10
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

6.2.2 Timer/Counter 2

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

– 16-bit auto-reload mode (up or down counting)
– 16-bit capture mode
– Baudrate generator (see 6.3.2.2 Serial Interface)

The modes are selected by bits in the SFR T2CON ($C8_H$) as shown in **table 6-3**:

Table 6-3
Timer/Counter 2 - Operating Modes

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

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

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

Special Function Register T2CON (Address $C8_H$)

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

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

The reset value of T2CON is 00_H.

6.2.2.1 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, C9$_H$). When DCEN is set, timer 2 can count up or down depending on the value of pin T2EX (P1.1).

Special Function Register T2MOD (Address C9$_H$)

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.

Reset value of T2MOD is XXXX XXX0$_B$.

Figure 6-11 shows timer 2 automatically counting up when DCEN = 0. In this mode there are two options selectable by bit EXEN2 in SFR T2CON.

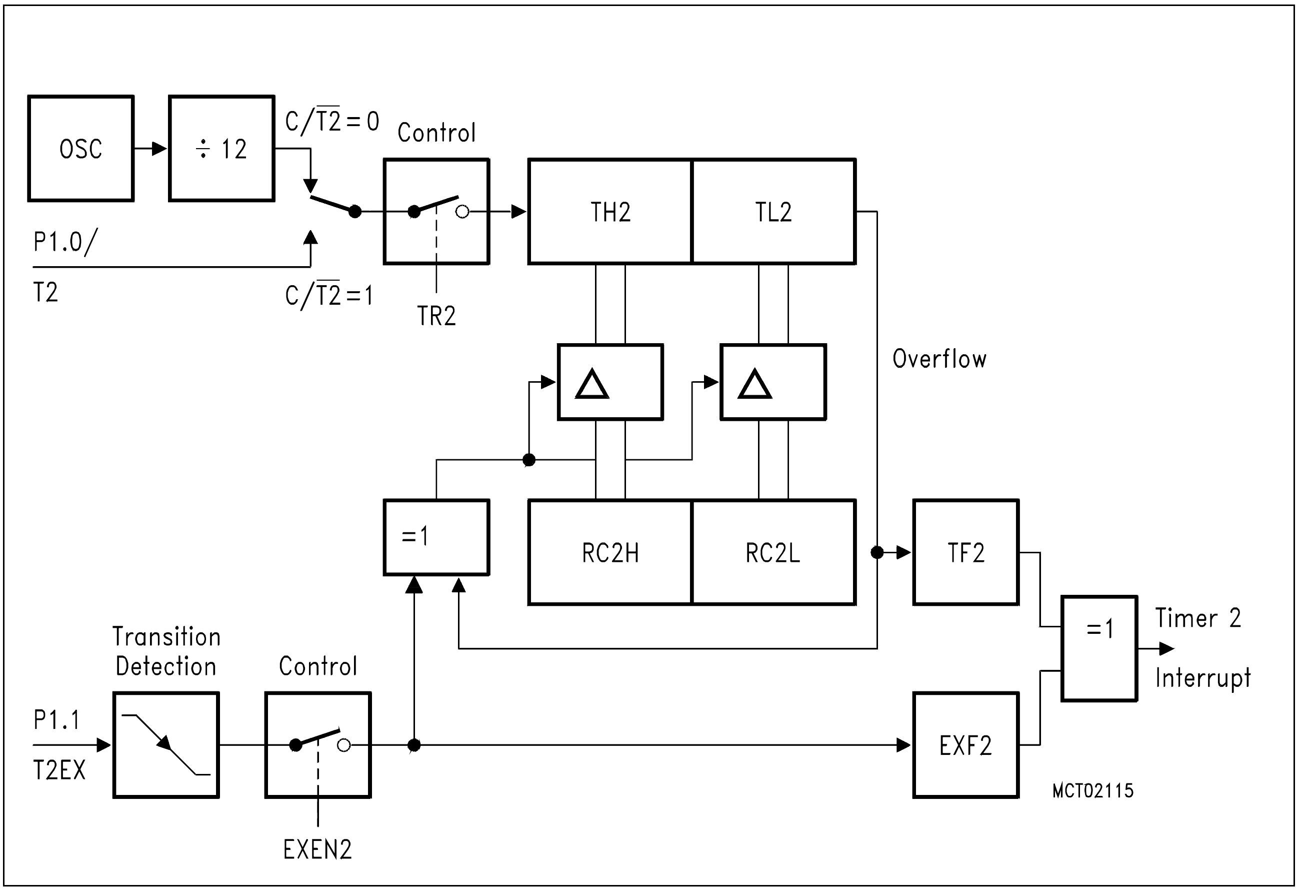


Figure 6-11
Timer 2 Auto-Reload Mode (DCEN = 0)

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

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

Setting the DCEN bit enables timer 2 to count up or down as shown in **figure 6-12**. In this mode the T2EX pin controls the direction of count.

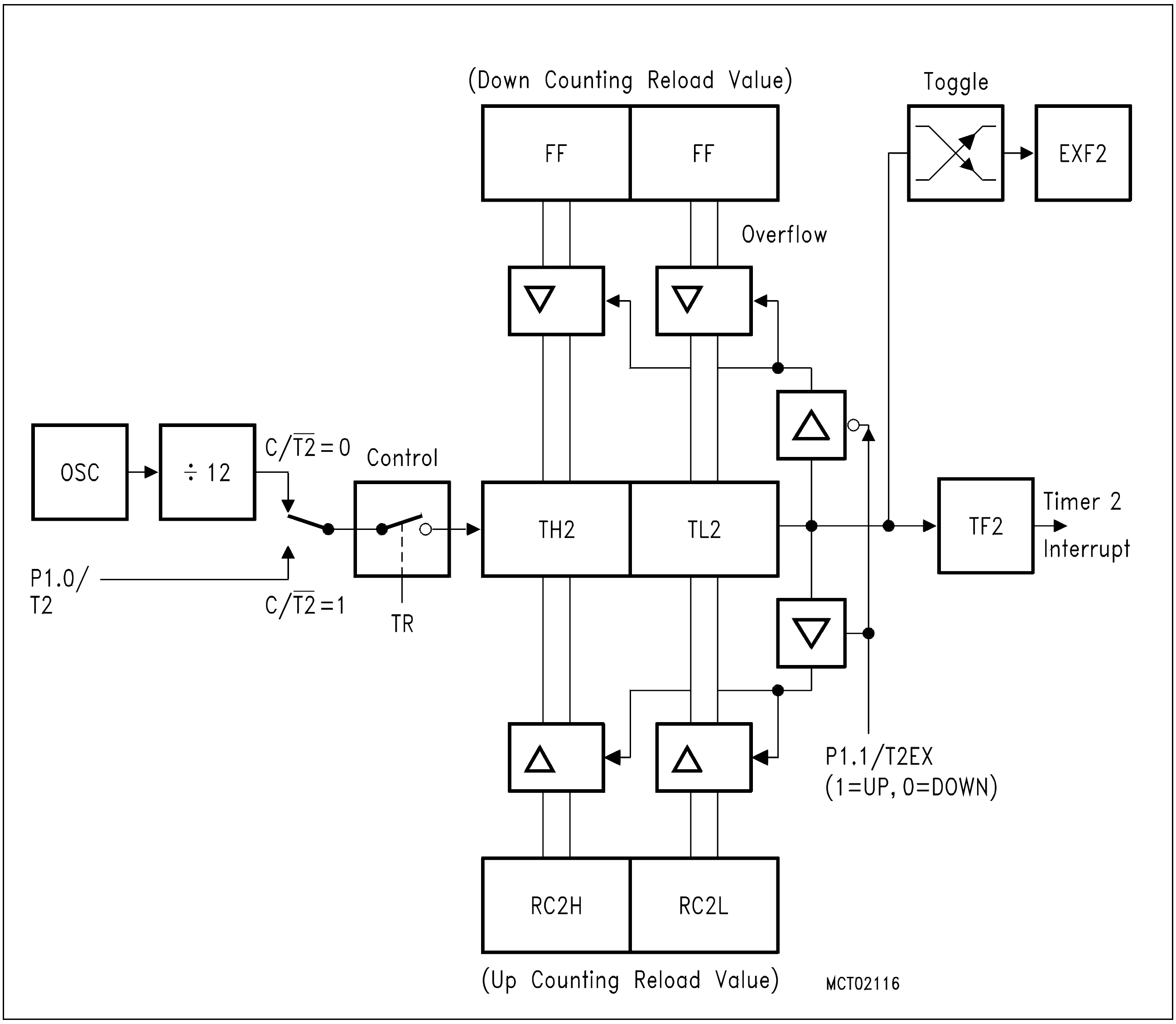


Figure 6-12
Timer 2 Auto-Reload Mode (DCEN = 1)

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

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

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

6.2.2.2 Capture

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

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

If EXEN2 = 1, timer 2 still does the above, but with added feature that a 1-to-0 transition at external input T2EX causes the current value in TH2 and TL2 to be captured into RC2H and RC2L, respectively. In addition, the transition at T2EX causes bit EXF2 in SFR T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in **figure 6-13**.

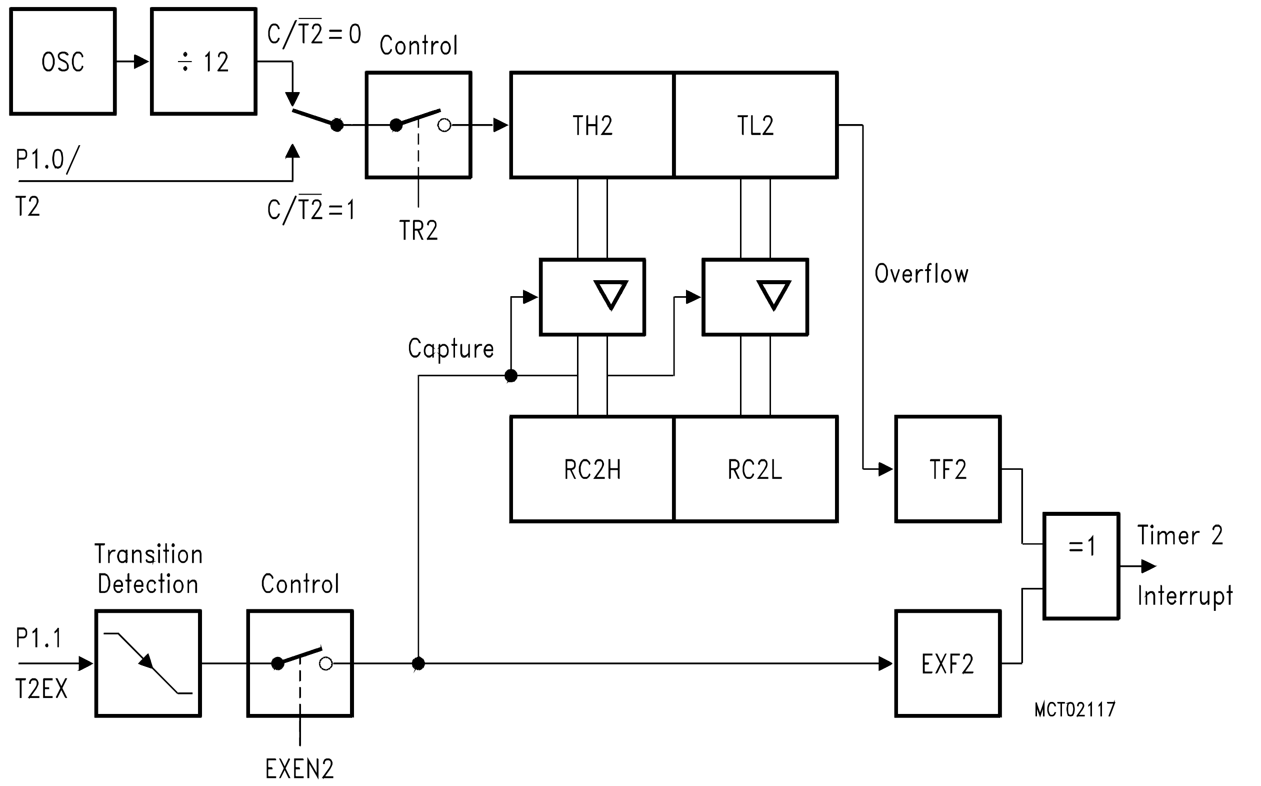


Figure 6-13
Timer 2 in Capture Mode

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

6.3 Serial Interface (USART)

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

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

Mode 0, Shift Register (Synchronous) Mode:

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 data bits are transmitted/ received: (LSB first). The baud rate is fixed at $^1/_{12}$ of the oscillator frequency. (See section 6.3.3 for more detailed information)

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

10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in special function register SCON. The baud rate is variable. (See section 6.3.4 for more detailed information)

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

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. The baud rate is programmable to either $^1/_{32}$ or $^1/_{64}$ of the oscillator frequency. (See section 6.3.5 for more detailed information)

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

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable. (See section 6.3.5 for more detailed information)

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

6.3.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th 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. A way to use this feature in multiprocessor systems is as follows.

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

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

Serial Port Control Register

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

Special Function Register SCON (Address 98_H)

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

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

6.3.2 Baud Rates

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

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

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

However, all formulas given in the following section are already include the factor and calculate the final baud rate.

Mode 0

The baud rate in mode 0 is fixed:

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

Mode 2

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

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

Modes 1 and 3

The baud rates in mode1 and 3 are determined by the timer overflow rate. These baud rates can be determined by timer 1 or by timer 2 or by both (one for transmit and the other for receive).

Also an internal baud rate generator can be used.

6.3.2.1 Using Timer 1 to Generate Baud Rates

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

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

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

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

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

Table 6-4 lists commonly used baud rates and how they can be obtained from timer 1.

Table 6-4
Timer 1 Generated Commonly Used Baud Rates

Baud Rate	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	$FEEB_H$

6.3.2.2 Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts timer 2 into its baud rate generator mode, as shown in **figure 6-14**.

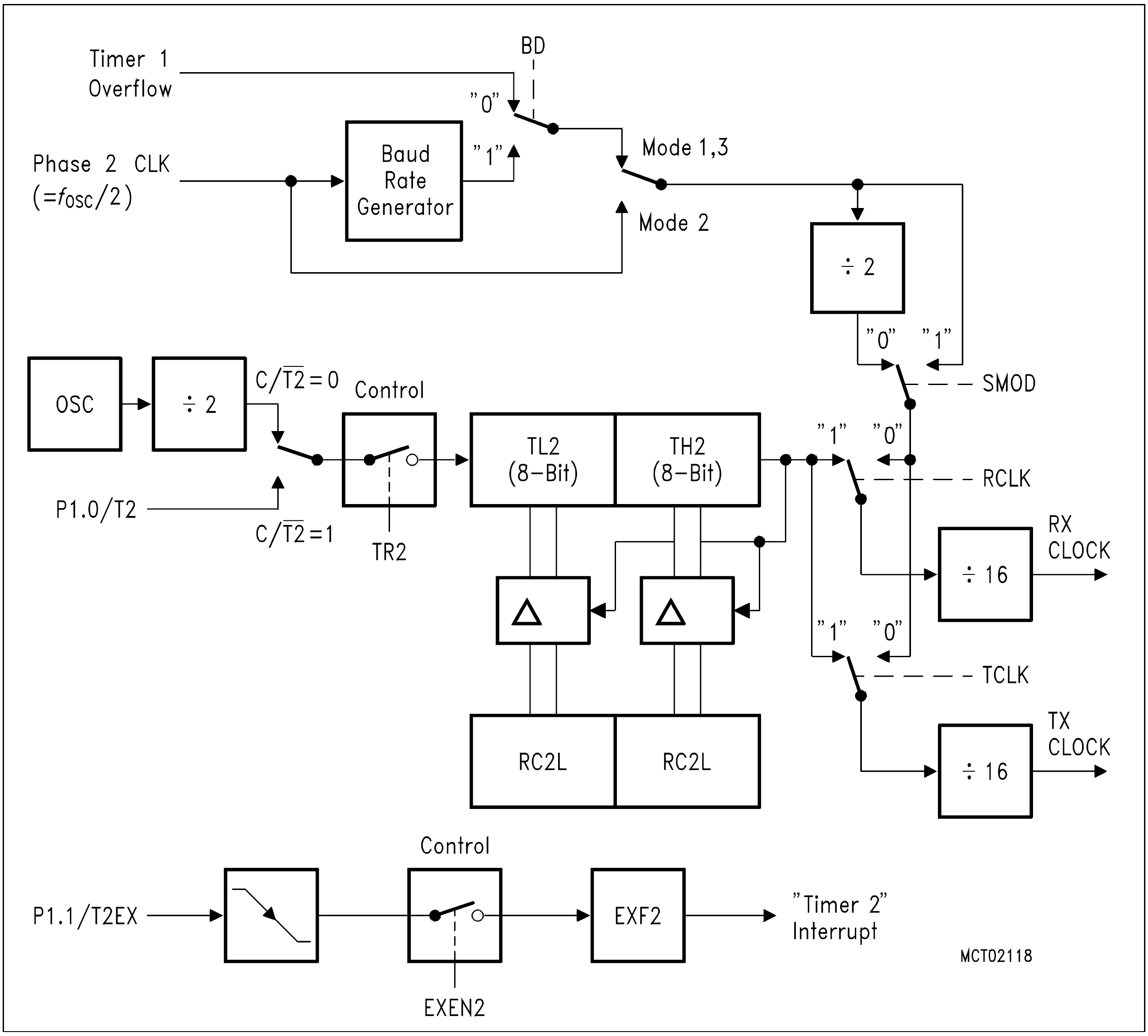


Figure 6-14
Timer 2 in Baud Rate Generator Mode

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

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

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

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

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

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

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

6.3.2.3 Using the internal Baudrate Generator

The serial channel has a baud rate generator which provides great flexibility and high resolution. To enable this feature in mode 1 and 3, bit BD (SFR BAUD) must be set.

The block diagram below shows the baud rate generator for the serial channel.

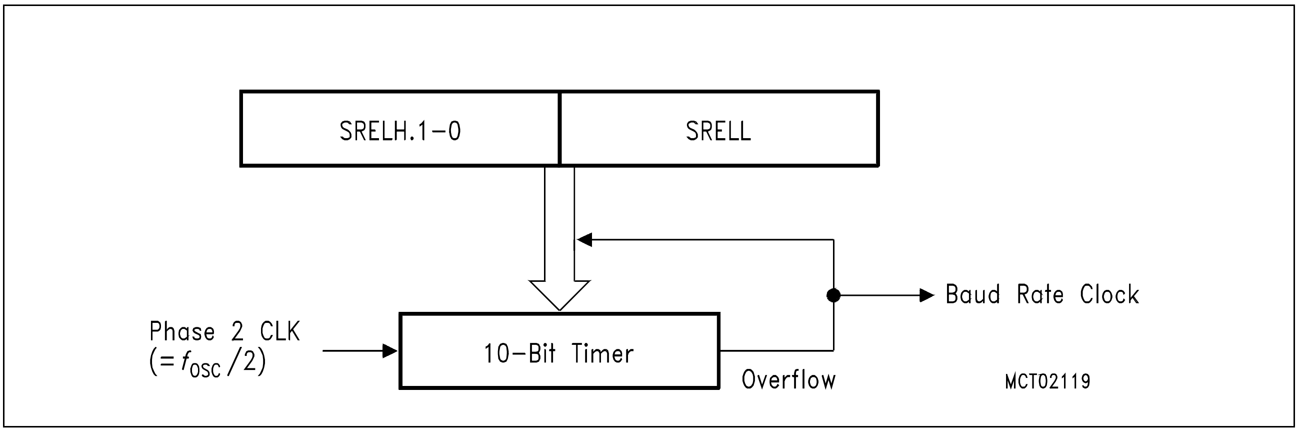


Figure 6-15
Baud Rate Generator for the Serial Interface

It consists of a free runnibg 10-bit timer with f_{OSC}/2 input frequency. On overflow of this timer there is an automatic reload from the registers SRELL (address AA$_H$) and SRELH (address BA$_H$). The lower 8 bits of the timer are reloaded from SRELL, while the upper two bits are reloaded from bit 0 and 1 of register SRELH. The baud rate timer is reloaded by writing to SRELL.

Special Function Register SRELH, SRELL (Addresses BA$_H$, AA$_H$)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
BA$_H$	▨	▨	▨	▨	▨	▨	MSB		SRELH
	7	6	5	4	3	2	1	0	
AA$_H$								LSB	SRELH

▨ Shaded areas are not used for programming of the baud rate time.

Bit	Function
SRELH.1-0	Reload value. Upper two bits of the ten bit timer reload value.
SRELL.7-0	Reload value. Lower 8 bit of the ten bit timer reload value.

Reset value of SRELL is D9$_H$, SRELH contains XXXX XX11$_B$.

Special Function Register BAUD (Address D8$_H$)

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
D8$_H$	BD	–	–	–	–	–	–	–	BAUD

Bit	Function
–	Not implemented, reserved for future use
BD	Baud Rate Generator Enable Bit. When set, the baud rate in mode 1 and 3 of the serial interface is taken from the internal baud rate generator.

Reset value of BAUD is 0XXX XXXX$_B$.

Mode 1, 3 baud rate = $(2^{SMOD} \times f_{OSC}) / (64 \times (2^{10} - SREL))$

6.3.3 Details about Mode 0

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

Figure 6-16a shows a simplyfied functional diagram of the serial port in mode 0. The associated timing is illustrated in **figure 6-16b**.

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

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

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

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

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

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

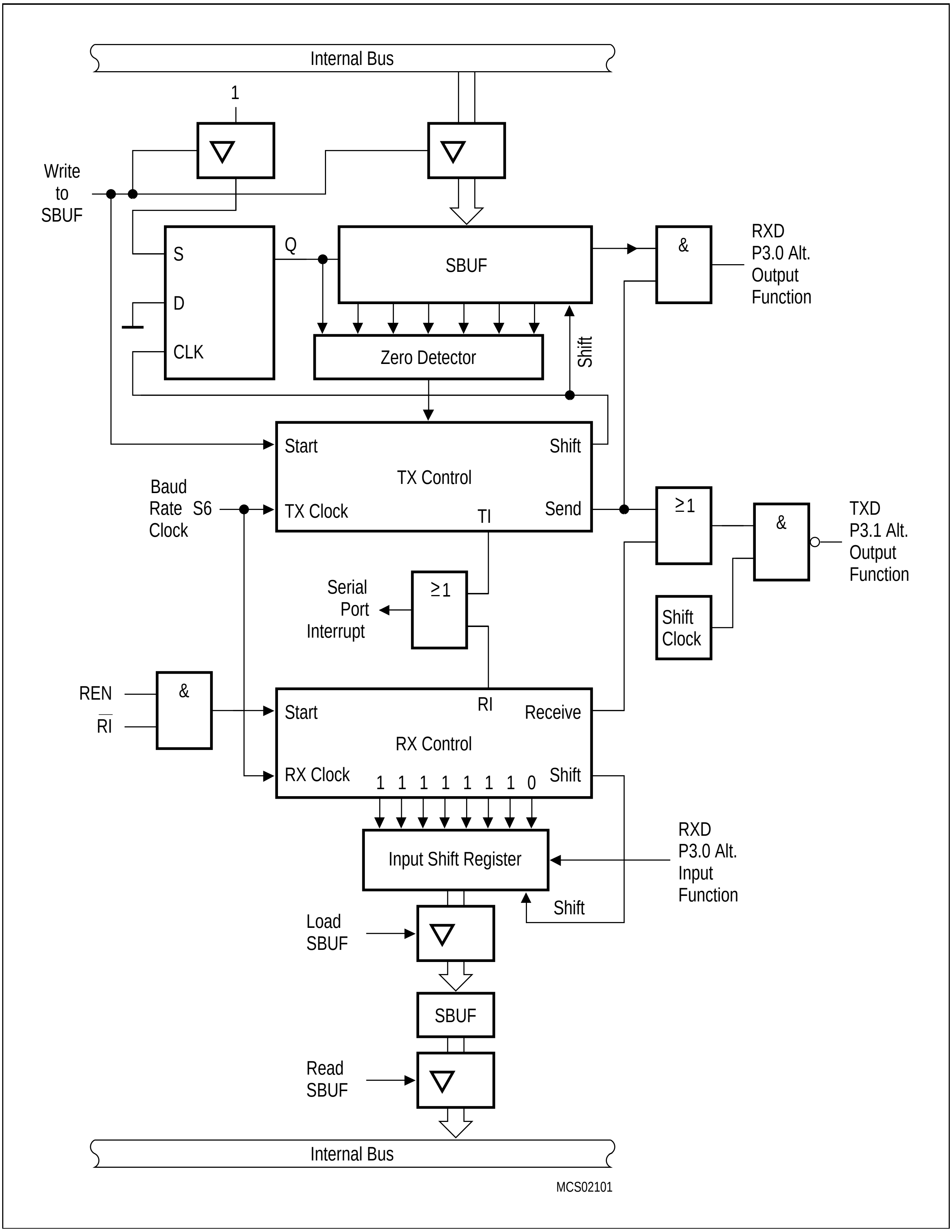


Figure 6-16a
Serial Interface, Mode 0, Functional Diagram

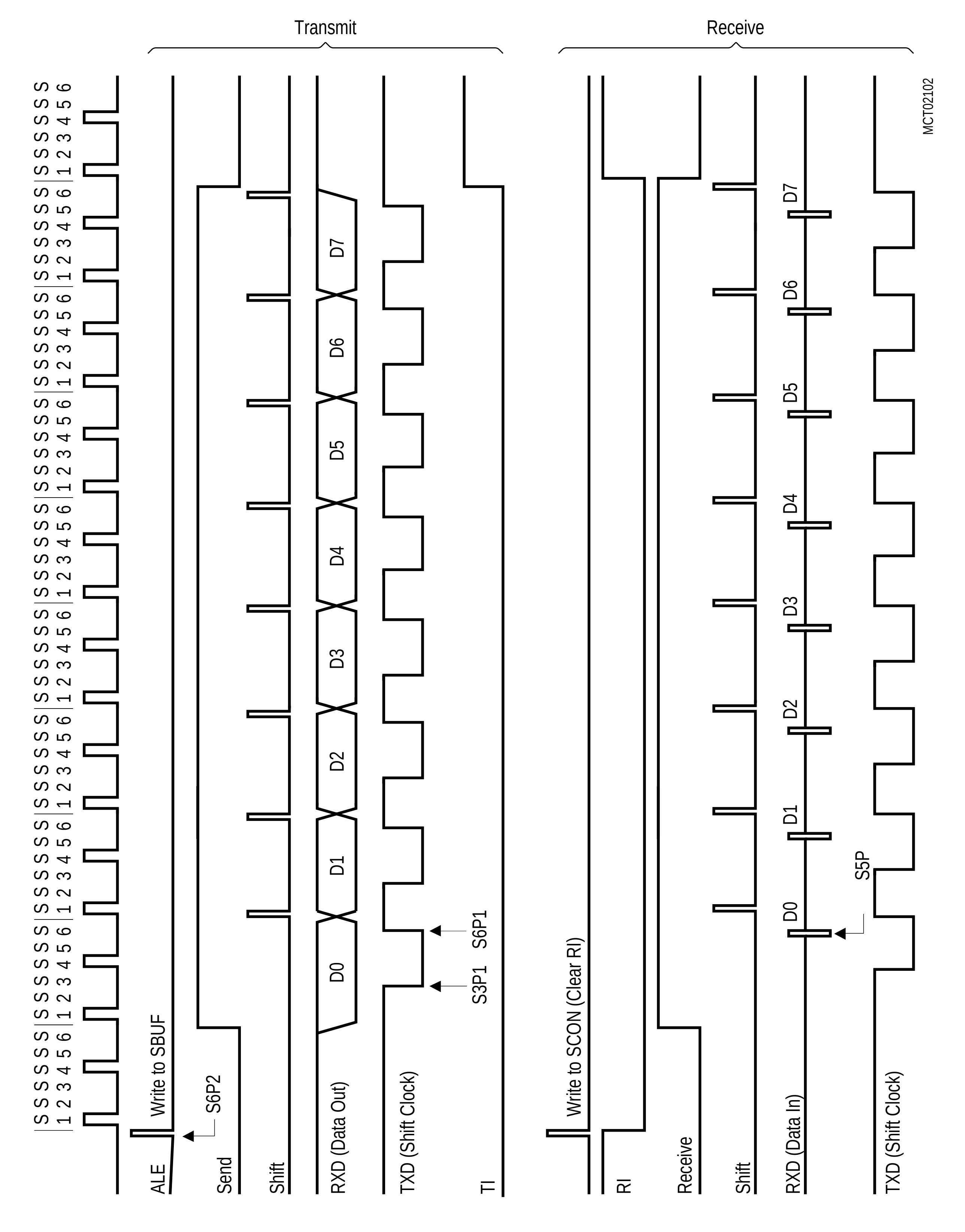


Figure 6-16b
Serial Interface, Mode 0, Timing Diagram

6.3.4 Details about Mode 1

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

Figure 6-17a shows a simplified functional diagram of the serial port in mode 1. The assiociated timings for transmit receive are illustrated in **figure 6-17b**.

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

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

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

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

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

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

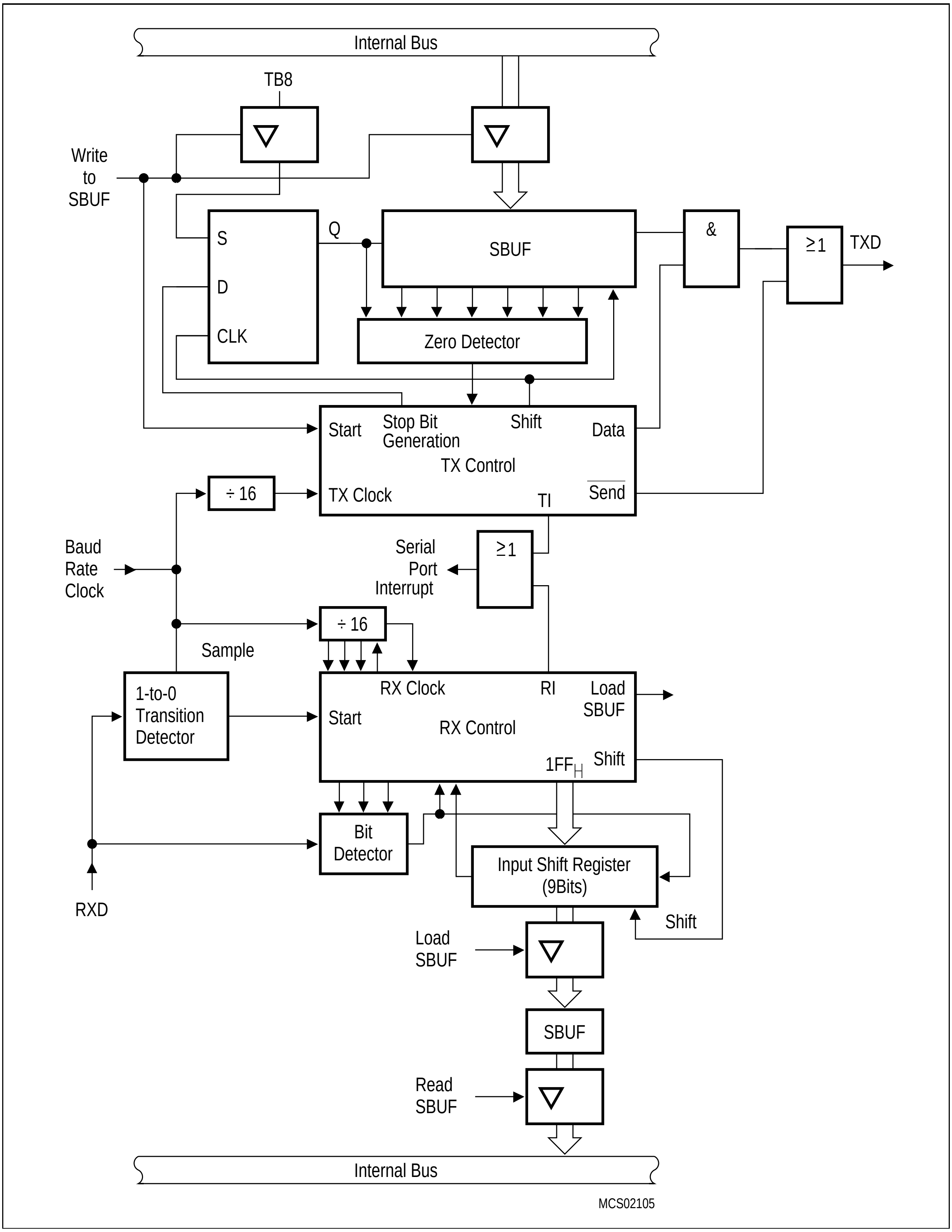


Figure 6-17a
Serial Interface, Mode 1, Functional Diagram

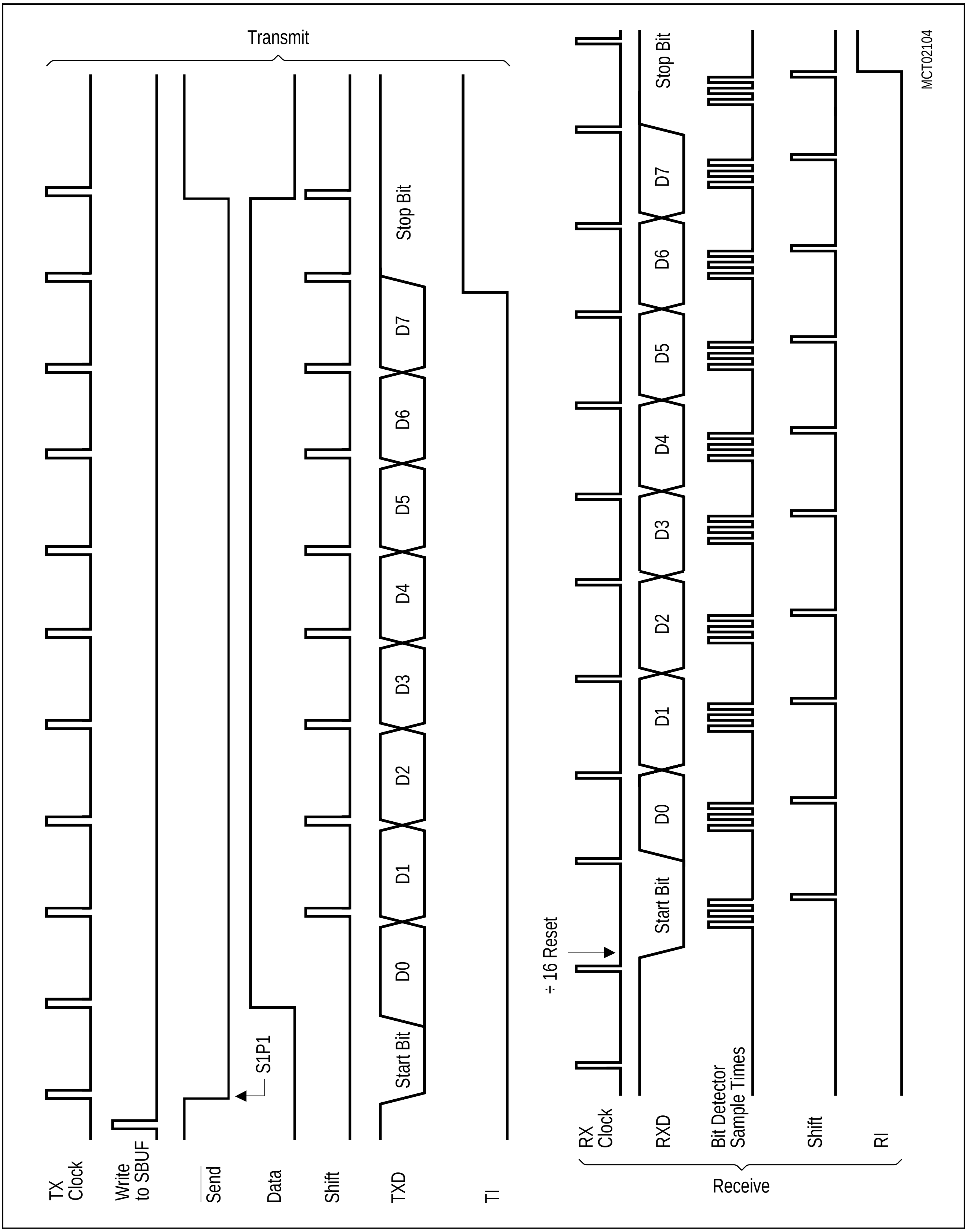


Figure 6-17b
Serial Interface, Mode 1, Timing Diagram

6.3.5 Details about Modes 2 and 3

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

Figure 6-18a shows a functional diagram of the serial port in modes 2 and 3. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register. The associated timings for transmit/receive are illustrated in **figure 6-18b**.

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.)

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

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

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

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

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

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

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

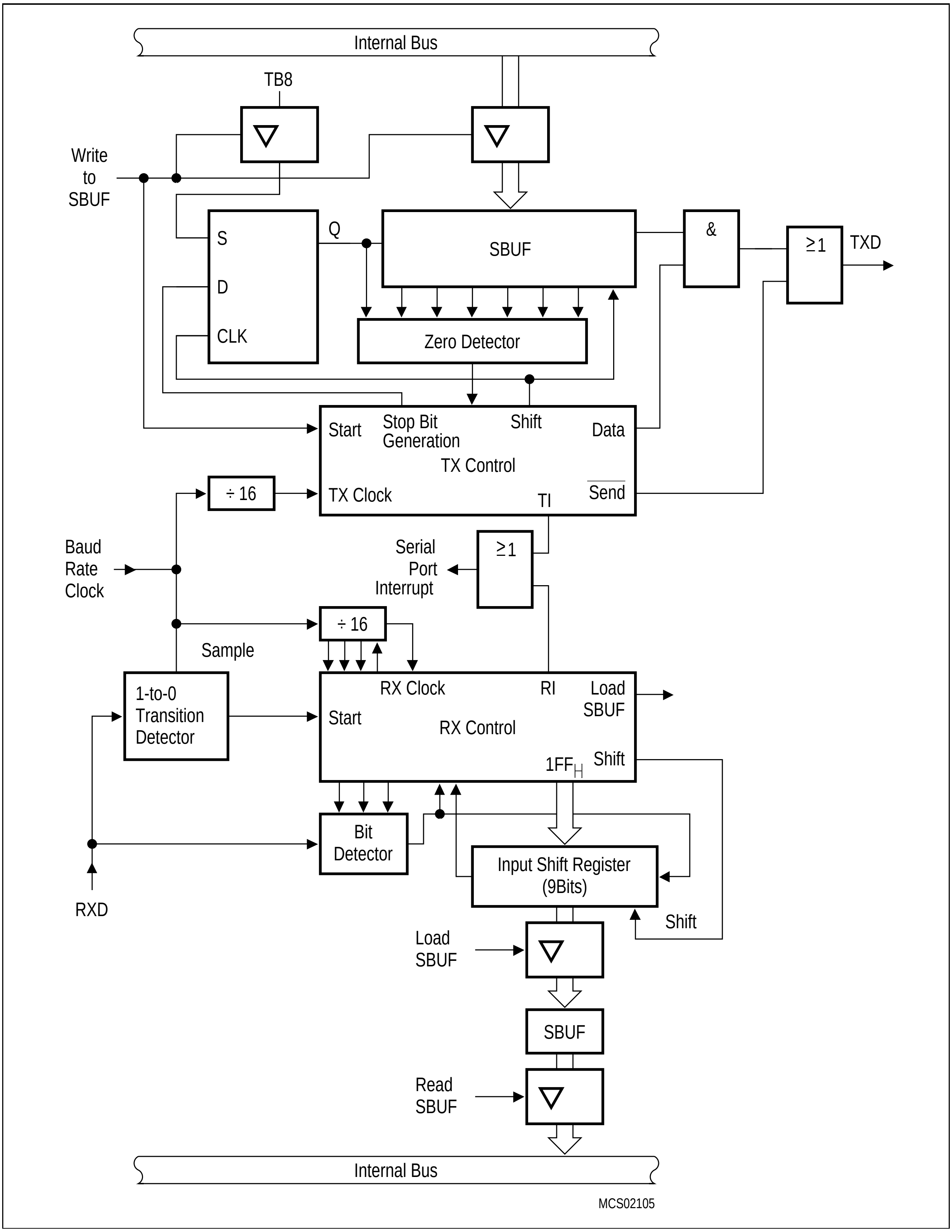


Figure 6-18a
Serial Interface, Mode 2 and 3, Functional Diagram

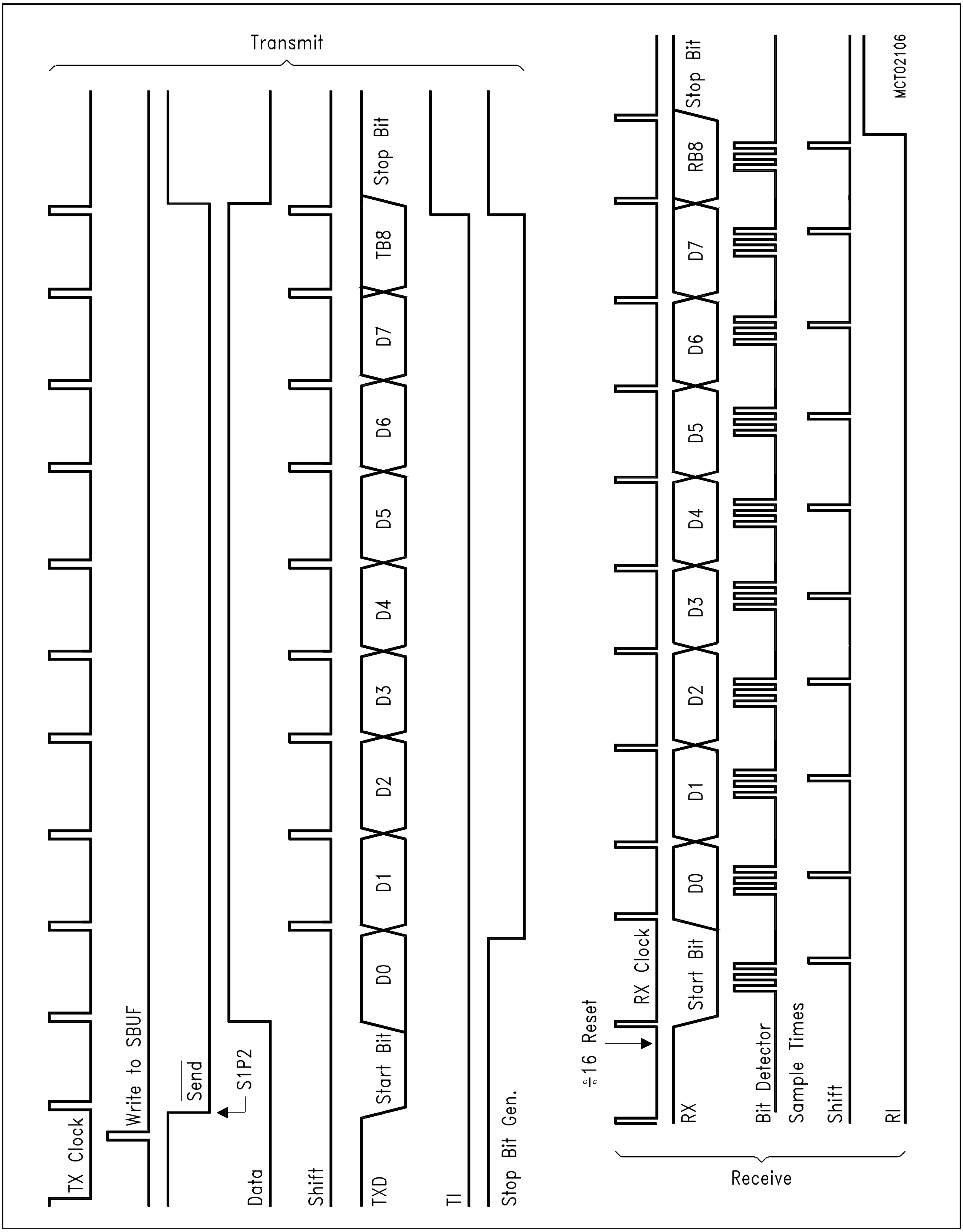


Figure 6-18b
Serial Interface, Mode 2 and 3, Timing

7 Interrupt System

The SAB-C502 provides 6 interrupt sources with two priority levels. Four interrupts can be generated by the on-chip peripherals (timer 0, timer 1, timer 2 and serial interface), and three interrupts may be triggered externally (P1.1/T2EX, P3.2/$\overline{\text{INT0}}$ and P3.3/$\overline{\text{INT1}}$).

Short Description of the Interrupt Structure for Advanced SAB-C502 Users

The interrupt structure of the SAB-C502 has been mainly adapted from the SAB 80C52. Thus, each interrupt source has its dedicated interrupt vector and can be enabled/disabled individually; there are also two priority levels available.

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

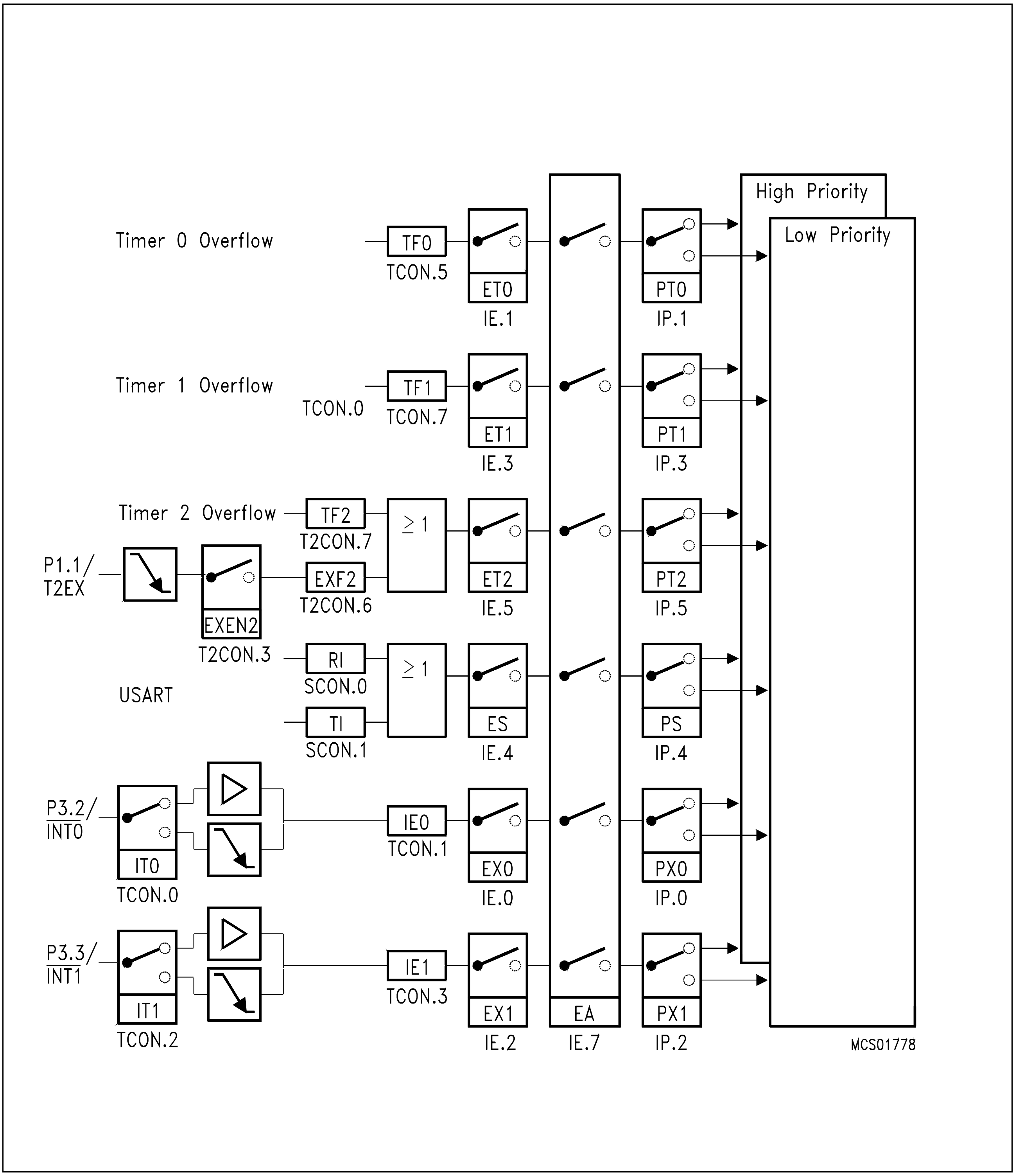


Figure 7-1
Interrupt Request Sources

7.1 Interrupt Structure

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

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

7.2 Interrupt Sources and Vectors

Source (Request Flags)	**Vector**	**Vector Address**
IE0	External interrupt 0	0003_H
TF0	Timer 0 interrupt	$000B_H$
IE1	External interrupt 1	0013_H
TF1	Timer 1 interrupt	$001B_H$
RI + TI	Serial port interrupt	0023_H
TF2 + EXF2	Timer 2 interrupt	$002B_H$

7.3 Interrupt Control Bits

7.3.1 Interrupt Enables

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

Special Function Registers IE (Address A8$_H$)

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
Addr. A8$_H$	EA	–	ET2	ES	ET1	EX1	ET0	EX0	IE

Bit	Function
–	Not implemented. Reserved for future use.
EA	Disables all Interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
ET2	Timer 2 Interrupt Enable. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	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.

Reset value of IE is 0X00 0000$_B$.

7.3.2 Interrupt Priorities

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

Special Function Registers IP (Address B8$_H$)

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
Addr. B8$_H$	–	–	PT2	PS	PT1	PX1	PT0	PX0	IP

Bit	Function
–	Not implemented. Reserved for future use.
PT2	Timer 2 Interrupt Priority Level. If PT2 = 0, the Timer 2 interrupt has a low priority.
PS	Serial Channel Interrupt Priority Level. 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.

Reset value of IP is XX00 0000$_B$.

A low-priority interrupt can itself 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.

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

Table 7-1
Priority-within-Level Structure

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

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

Table 7-2
Location of Interrupt Sources Request Flags

Interrupt Request Flag		SFR	Address	Bit-Addr.
External Interrupt 0,	IE0	TCON	88_H	88_H
Timer 0 Interrupt,	TF0	TCON	88_H	$8D_H$
External Interrupt 1,	IE1	TCON	88_H	$8B_H$
Timer 1 Interrupt,	TF1	TCON	88_H	$8F_H$
Serial Channel,	RI	SCON	98_H	98_H
Serial Channel,	TI	SCON	98_H	99_H
Timer 2 Interrupt,	TF2	T2CON	$C8_H$	CF_H
Timer 2 Interrupt,	EXF2	T2CON	$C8_H$	CE_H

7.4 How Interrupts are Handled

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

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

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

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

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

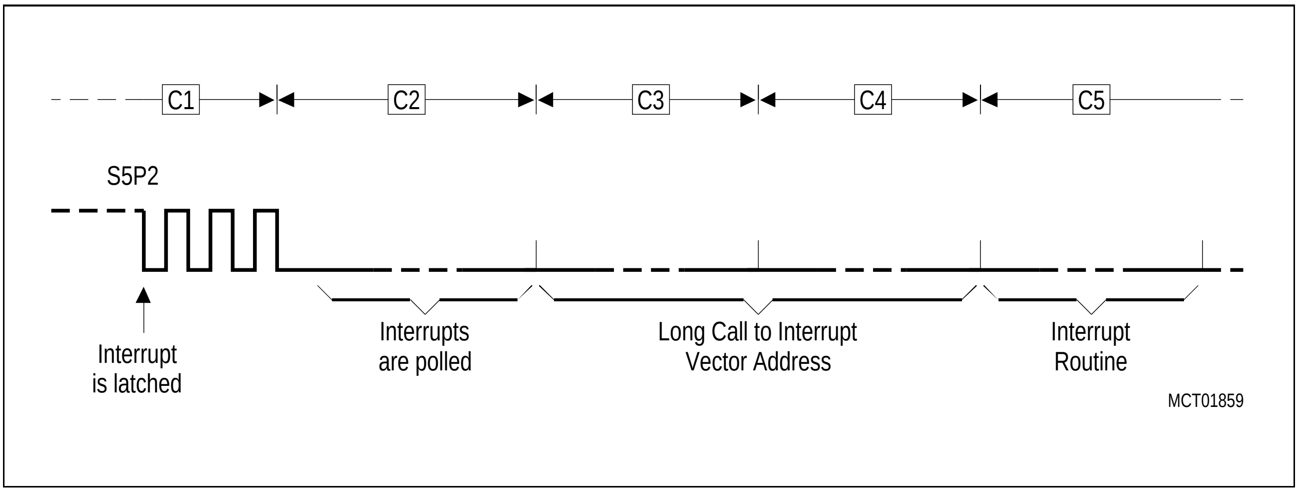


Figure 7-2
Interrupt Response Timing Diagram

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

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

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

7.5 External Interrupts

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

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

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

Since the external interrupt pins are sampled once in each machine cycle, an input low should be held for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set (**see figure 7-3**). The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

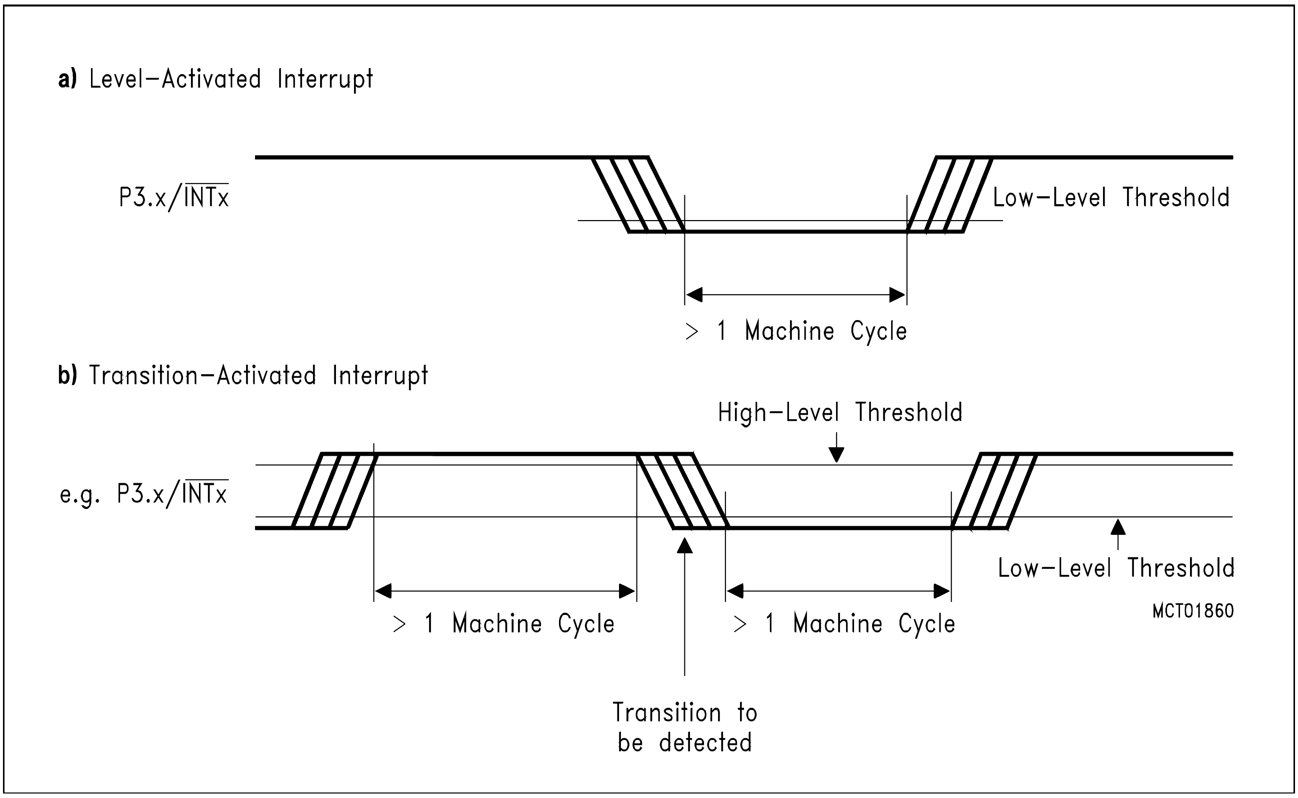

a) Level–Activated Interrupt
P3.x/$\overline{\text{INTx}}$
Low–Level Threshold
> 1 Machine Cycle
b) Transition–Activated Interrupt
High–Level Threshold
e.g. P3.x/$\overline{\text{INTx}}$
Low–Level Threshold
> 1 Machine Cycle
> 1 Machine Cycle
MCT01860
Transition to be detected

Figure 7-3
External Interrupt Detection

7.6 Response Time

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

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

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

8 Fail Safe Mechanisms

The SAB-C502 offers enhanced fail safe mechanisms, which allow an automatic recovery from software upset or hardware failure :

- a programmable watchdog timer (WDT), with variable time-out period from 512 µs up to approx. 1.1 s at 12 MHz.
- an oscillator watchdog (OWD) which monitors the on-chip oscillator and forces the microcontroller into reset state in case 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 upset, the user's program has to clear the watchdog within a previously programmed time period. If the software fails to do this periodical 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 hardware-related problems.

The Watchdog Timer in the SAB-C502 is a 15-bit timer, which is incremented by a count rate of either $f_{\text{CYCLE}}/2$ or $f_{\text{CYCLE}}/32$ ($f_{\text{CYCLE}} = f_{\text{OSC}}/12$). That is, the machine clock is divided by a series of arrangement of two prescalers, a divide-by-two and a divide-by-16 prescaler. The latter is enabled by setting bit WDTREL.7.

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

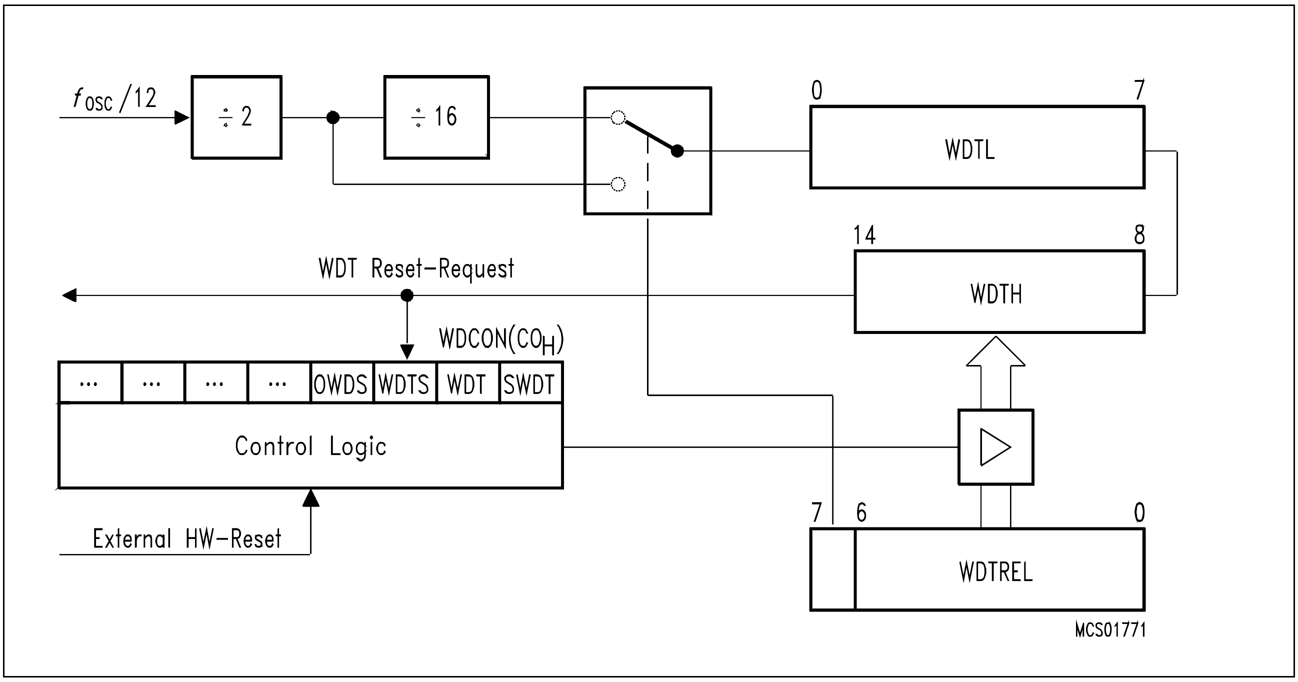


Figure 8-1
Block Diagram of the Programmable Watchdog Timer

Special Function Register WDTREL (Address 086$_H$)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
86$_H$	Watchdog Timer Reload Register								WDTREL

Bit	Function
WDTREL.7	Prescaler select bit. When set, the watchdog timer is clocked through an additional divide-by-16 prescaler .
WDTREL.6 to WDTREL.0	Seven bit reload value for the high-byte of the watchdog timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

Reset value of WDTREL is 00$_H$.

Special Function Register WDCON (Address C0$_H$)

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
C0$_H$	–	–	–	–	OWDS	WDTS	WDT	SWDT	WDCON

Bit	Function
–	Not implemented. Reserved for future use.
OWDS	Oscillator Watchdog Timer Status Flag. Set by hardware when an oscillator watchdog reset occured. Can be set and cleared by software.
WDTS	Watchdog Timer Status Flag. Set by hardware when a Watchdog Timer reset occured. 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.

Reset value of WDCON is XXXX 0000$_B$.

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

Examples (given for 12- and 20-MHz oscillator frequency):

WDTREL	Time-Out Period		Comments
	f_{osc} = 12 MHz	f_{osc} = 20 MHz	
00_H	65.535 ms	39.321 ms	This is the default value
80_H	1.1 s	0.65 s	Maximum time period
$7F_H$	512 µs	307 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 (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 the idle mode and power-down mode of the processor (see section "Power Saving Modes"). Therefore, it is possible to use the idle mode in combination with the watchdog timer function. But 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 but 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 automatically be cleared 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.

The reload register WDTREL can be written at any time, as already mentioned. Therefore, a periodical refresh of WDTREL can be added to the above mentioned 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). This internal reset differs from an external one in so far as 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 examine from which source the reset was activated. The bit WDTS can also be cleared by software.

8.2 Oscillator Watchdog Unit

The unit serves two functions:

- **Monitoring of the on-chip oscillator’s function.**
  The watchdog supervises the on-chip oscillator's frequency; if it 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 (i.e. the on-chip oscillator has a higher frequency than the RC oscillator), the part executes a final reset phase of typ. 1 ms in order to allow the oscillator to stabilize; then the oscillator watchdog reset is released and the part starts program execution again.
- **Fast internal reset after power-on.**
  The oscillator watchdog unit provides a clock supply for the reset before the on-chip oscillator has started. The oscillator watchdog unit also works identically to the monitoring function.

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 the comparison with the frequency of the on-chip oscillator.

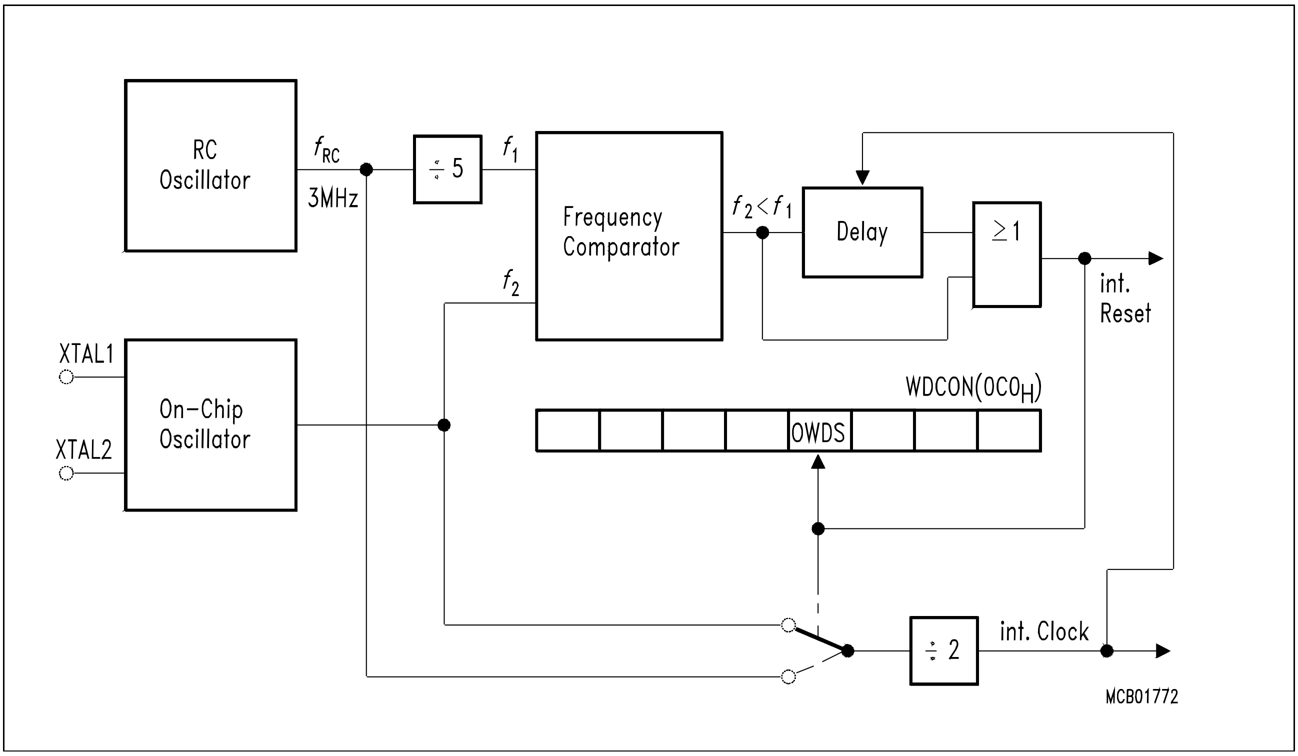


Figure 8-2
Functional Block Diagram of the Oscillator Watchdog

The frequency coming from the RC oscillator is divided by 5 and compared to the on-chip oscillator's frequency. If the frequency coming from the on-chip oscillator is found lower than the frequency derived from the RC oscillator the watchdog detects a failure condition (the 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 part 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 in order to bring the part in its defined reset state. The reset is performed because clock is available from the RC oscillator. This internal watchdog reset has the same effects as an externally applied reset signal with the following exceptions: The Watchdog Timer Status flag WDTS is not reset (the Watchdog Timer however is stopped); and bit OWDS is set. This allows the software to examine error conditions detected by the Watchdog Timer even if meanwhile an oscillator failure occured.

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 part is held in reset. This allows a reliable stabilization of the on chip oscillator. After that, the watchdog toggles 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 also the external reset request disappears.

Furthermore, the status flag OWDS is set if the oscillator watchdog was active. The status flag can be evaluated by software to detect that a reset was 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).

8.2.2 Fast Internal Reset after Power-On

The SAB-C502 can use the oscillator watchdog unit for a fast internal reset procedure after power-on.

Normally the members of the 8051 family (e. g. 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 in order 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. 1 ms). During this time period the pins have an undefined state which could have severe effects e.g. to actuators connected to port pins.

In the SAB-C502 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 microseconds). In the following 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 valid the watchdog uses the RC oscillator output as clock source for the chip. This allows correct resetting of the part and brings all ports to the defined state.

Delay time between power-on and correct reset state: max 34 µs.

9 Power Saving Modes

The ACMOS technology of the SAB-C502 allows two power saving modes of the device:

- – Idle mode
- – Power-down mode.

The Special Function Register PCON

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

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

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

Special Function Register PCON (Address 87_H)

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

Symbol	Position	Function
SMOD	PCON.7	Baud rate doubled. When set, the baud rate of the serial channel in mode 1,2,3 is doubled.
PDS	PCON.6	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode
IDLS	PCON.5	Idle start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
–	PCON.4	Not implemented. Reserved for future use.
GF1	PCON.3	General purpose flag
GF0	PCON.2	General purpose flag
PDE	PCON.1	Power-down enable bit. When set, starting of the power-down is enabled
IDLE	PCON.0	Idle mode enable bit. When set, starting of the idle mode is enabled

The reset value of PCON is 000X $0000_{B.}$

9.1 Idle Mode

In the idle mode the oscillator of the SAB-C502 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D converter, and all timers with the exception of the watchdog timer are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

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

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

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

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to 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 $\overline{\text{PSEN}}$ hold at logic high levels.

Table 9-1
Status of External Pins During Idle and Power-Down Mode

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

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

The idle mode is a useful feature which makes it possible to "freeze" the processor’s status - either for a predefined time, or until an external event reverts the controller to normal operation, as discussed below. The watchdog timer is the only peripheral which is automatically stopped during idle mode. If it were not disabled on entering idle mode, the watchdog timer would reset the controller, thus abandoning the idle mode.

The idle mode is entered by two consecutive instructions. The first instruction sets the flag bit IDLE (PCON.0) and must not set bit IDLS (PCON.5), the following instruction sets the start bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). The hardware ensures that a concurrent setting of both bits, IDLE and IDLS, does not initiate the idle mode. Bits IDLE and IDLS will automatically be cleared after being set. If one of these register bits is read the value that appears is 0. This double instruction is implemented to minimize the chance of an unintentional entering of the idle mode which would leave the watchdog timer’s task of system protection without effect.

Note:

PCON is not a bit-addressable register, so the above mentioned sequence for entering the idle mode is obtained by byte-handling instructions, as shown in the following example:

```
ORL      PCON,#00000001B      ;Set bit IDLE, bit IDLS must not be set
ORL      PCON,#00100000B      ;Set bit IDLS, bit IDLE must not be set
```

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

There are two ways to terminate the idle mode:

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

9.2 Power-Down Mode

In the power-down mode, the on-chip oscillator is stopped. Therefore all functions are stopped; only the contents of the on-chip RAM and the SFR’s are maintained. The port pins controlled by their port latches output the values that are held by their SFR’s. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode. ALE and $\overline{\text{PSEN}}$ hold at logic low level (**see table 9-1**).

The power-down mode is entered by two consecutive instructions. The first instruction has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6), the following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, does not initiate the power-down mode. Bits PDE and PDS will automatically be cleared after having been set and the value shown by reading one of these bits is always 0. This double instruction is implemented to minimize the chance of unintentionally entering the power-down mode which could possibly ”freeze” the chip's activity in an undesired status.

Note:

PCON is not a bit-addressable register, so the above mentioned sequence for entering the power-down mode is obtained by byte-handling instructions, as shown in the following example:

```
ORL       PCON,#00000010B       ;Set bit PDE, bit PDS must not be set
ORL       PCON,#01000000B       ;Set bit PDS, bit PDE must not be set
```

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

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

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

Absolute Maximum Ratings

Ambient temperature under bias (T_A)	– 40 °C to + 85 °C
Storage temperature (T_{ST})	– 65 °C to + 150 °C
Voltage on V_{CC} pins with respect to ground (V_{SS})	– 0.5 V to 6.5 V
Voltage on any pin with respect to ground (V_{SS})	– 0.5 V to V_{CC} + 0.5 V
Input current on any pin during overload condition	– 10 mA to + 10 mA
Absolute sum of all input currents during overload condition	| 100 mA |
Power dissipation	TBD

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.*

DC Characteristics

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V;	T_A = 0 to + 70 °C for the SAB-C502
T_A = – 40 to + 85 °C for the SAF-C502

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (except $\overline{\text{EA}}$, RESET)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{\text{EA}}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input low voltage (RESET)	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage (except $\overline{\text{EA}}$, RESET, XTAL1)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL1	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	
Input high voltage to RESET, $\overline{\text{EA}}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (ports 2, 3)	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA[1]
Output low voltage (port 0, ALE, $\overline{\text{PSEN}}$)	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA[1]
Output high voltage (ports 2, 3)	V_{OH}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 80 µA I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, $\overline{\text{PSEN}}$)	V_{OH1}	2.4 0.9 V_{CC}	– –	V	I_{OH} = – 800 µA[2], I_{OH} = – 80 µA[2]
Logic 0 input current (ports 1, 2, 3)	I_{IL}	– 10	– 50	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current (ports 1, 2, 3)	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, $\overline{\text{EA}}$, P1)	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 °C
Power supply current:					
Active mode, 12 MHz[7]	I_{CC}	–	23.3	mA	V_{CC} = 5 V,[4]
Idle mode, 12 MHz[7]	I_{CC}	–	7.4	mA	V_{CC} = 5 V,[5]
Active mode, 20 MHz[7]	I_{CC}	–	33.9	mA	V_{CC} = 5 V,[4]
Idle mode, 20 MHz[7]	I_{CC}	–	10.6	mA	V_{CC} = 5 V,[5]
Power Down Mode	I_{PD}	–	50	µA	V_{CC} = 2 … 5.5 V,[3]

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

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

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

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

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

[7] $I_{CC\,max}$ at other frequencies is given by:
active mode: $I_{CC\,max} = 1.32 \times f_{OSC} + 7.48$
idle mode: $I_{CC\,max} = 0.40 \times f_{OSC} + 2.62$
where f_{OSC} is the oscillator frequency in MHz. I_{CC} values are given in mA and measured at V_{CC} = 5 V.

AC Characteristics for SAB-C502-L / C502-2R

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to + 70 °C	for the SAB-C502
T_A = – 40 °C to + 85 °C	for the SAF-C502

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

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	127	–	$2t_{CLCL} – 40$	–	ns
Address setup to ALE	t_{AVLL}	43	–	$t_{CLCL} – 40$	–	ns
Address hold after ALE	t_{LLAX}	30	–	$t_{CLCL} – 53$	–	ns
ALE low to valid instr in	t_{LLIV}	–	233	–	$4t_{CLCL} – 100$	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	58	–	$t_{CLCL} – 25$	–	ns
$\overline{\text{PSEN}}$ pulse width	t_{PLPH}	215	–	$3t_{CLCL} – 35$	–	ns
$\overline{\text{PSEN}}$ to valid instr in	t_{PLIV}	–	150	–	$3t_{CLCL} – 100$	ns
Input instruction hold after $\overline{\text{PSEN}}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{\text{PSEN}}$	t_{PXIZ}*)	–	63	–	$t_{CLCL} – 20$	ns
Address valid after $\overline{\text{PSEN}}$	t_{PXAV}*)	75	–	$t_{CLCL} – 8$	–	ns
Address to valid instr in	t_{AVIV}	–	302	–	$5t_{CLCL} – 115$	ns
Address float to $\overline{\text{PSEN}}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the SAB-C502-L/C502-2R to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 Drivers.

AC Characteristics for SAB-C502-L / C502-2R

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		12 MHz Clock		Variable Clock $1/t_{\text{CLCL}}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
$\overline{\text{RD}}$ pulse width	t_{RLRH}	400	–	$6t_{\text{CLCL}} - 100$	–	ns
$\overline{\text{WR}}$ pulse width	t_{WLWH}	400	–	$6t_{\text{CLCL}} - 100$	–	ns
Address hold after ALE	t_{LLAX2}	30	–	$t_{\text{CLCL}} - 53$	–	ns
$\overline{\text{RD}}$ to valid data in	t_{RLDV}	–	252	–	$5t_{\text{CLCL}} - 165$	ns
Data hold after $\overline{\text{RD}}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{\text{RD}}$	t_{RHDZ}	–	97	–	$2t_{\text{CLCL}} - 70$	ns
ALE to valid data in	t_{LLDV}	–	517	–	$8t_{\text{CLCL}} - 150$	ns
Address to valid data in	t_{AVDV}	–	585	–	$9t_{\text{CLCL}} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	200	300	$3t_{\text{CLCL}} - 50$	$3t_{\text{CLCL}} + 50$	ns
Address valid to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	203	–	$4t_{\text{CLCL}} - 130$	–	ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ high to ALE high	t_{WHLH}	43	123	$t_{\text{CLCL}} - 40$	$t_{\text{CLCL}} + 40$	ns
Data valid to $\overline{\text{WR}}$ transition	t_{QVWX}	33	–	$t_{\text{CLCL}} - 50$	–	ns
Data setup before $\overline{\text{WR}}$	t_{QVWH}	433	–	$7t_{\text{CLCL}} - 150$	–	ns
Data hold after $\overline{\text{WR}}$	t_{WHQX}	33	–	$t_{\text{CLCL}} - 50$	–	ns
Address float after $\overline{\text{RD}}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 12 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	83.3	285.7	ns
High time	t_{CHCX}	20	$t_{\mathrm{CLCL}} - t_{\mathrm{CLCX}}$	ns
Low time	t_{CLCX}	20	$t_{\mathrm{CLCL}} - t_{\mathrm{CHCX}}$	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns

AC Characteristics for SAB-C502-L20 / C502-2R20

V_{CC} = 5 V + 10 %, – 15 %; V_{SS} = 0 V

T_A = 0 °C to + 70 °C for the SAB-C502
T_A = – 40 °C to + 85 °C for the SAF-C502

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

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		20 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	60	–	$2t_{CLCL}$ – 40	–	ns
Address setup to ALE	t_{AVLL}	20	–	t_{CLCL} – 30	–	ns
Address hold after ALE	t_{LLAX}	20	–	t_{CLCL} – 30	–	ns
ALE low to valid instr in	t_{LLIV}	–	100	–	$4t_{CLCL}$ – 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	25	–	t_{CLCL} – 25	–	ns
$\overline{\text{PSEN}}$ pulse width	t_{PLPH}	115	–	$3t_{CLCL}$ – 35	–	ns
$\overline{\text{PSEN}}$ to valid instr in	t_{PLIV}	–	75	–	$3t_{CLCL}$ – 75	ns
Input instruction hold after $\overline{\text{PSEN}}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{\text{PSEN}}$	t_{PXIZ}*)	–	40	–	t_{CLCL} – 10	ns
Address valid after $\overline{\text{PSEN}}$	t_{PXAV}*)	47	–	t_{CLCL} – 3	–	ns
Address to valid instr in	t_{AVIV}	–	190	–	$5t_{CLCL}$ – 60	ns
Address float to $\overline{\text{PSEN}}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the SAB-C502-L20/C502-2R20 to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 Drivers.

AC Characteristics for SAB-C502-L20 / C502-2R20

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{\text{CLCL}}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
$\overline{\text{RD}}$ pulse width	t_{RLRH}	200	–	$6t_{\text{CLCL}} - 100$	–	ns
$\overline{\text{WR}}$ pulse width	t_{WLWH}	200	–	$6t_{\text{CLCL}} - 100$	–	ns
Address hold after ALE	t_{LLAX2}	20	–	$t_{\text{CLCL}} - 30$	–	ns
$\overline{\text{RD}}$ to valid data in	t_{RLDV}	–	155	–	$5t_{\text{CLCL}} - 95$	ns
Data hold after $\overline{\text{RD}}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{\text{RD}}$	t_{RHDZ}	–	76	–	$2t_{\text{CLCL}} - 24$	ns
ALE to valid data in	t_{LLDV}	–	250	–	$8t_{\text{CLCL}} - 150$	ns
Address to valid data in	t_{AVDV}	–	285	–	$9t_{\text{CLCL}} - 165$	ns
ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{LLWL}	100	200	$3t_{\text{CLCL}} - 50$	$3t_{\text{CLCL}} + 50$	ns
Address valid to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	t_{AVWL}	70	–	$4t_{\text{CLCL}} - 130$	–	ns
$\overline{\text{WR}}$ or $\overline{\text{RD}}$ high to ALE high	t_{WHLH}	20	80	$t_{\text{CLCL}} - 30$	$t_{\text{CLCL}} + 30$	ns
Data valid to $\overline{\text{WR}}$ transition	t_{QVWX}	5	–	$t_{\text{CLCL}} - 45$	–	ns
Data setup before $\overline{\text{WR}}$	t_{QVWH}	200	–	$7t_{\text{CLCL}} - 150$	–	ns
Data hold after $\overline{\text{WR}}$	t_{WHQX}	10	–	$t_{\text{CLCL}} - 40$	–	ns
Address float after $\overline{\text{RD}}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 20 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	50	285.7	ns
High time	t_{CHCX}	12	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	12	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	12	ns
Fall time	t_{CHCL}	–	12	ns

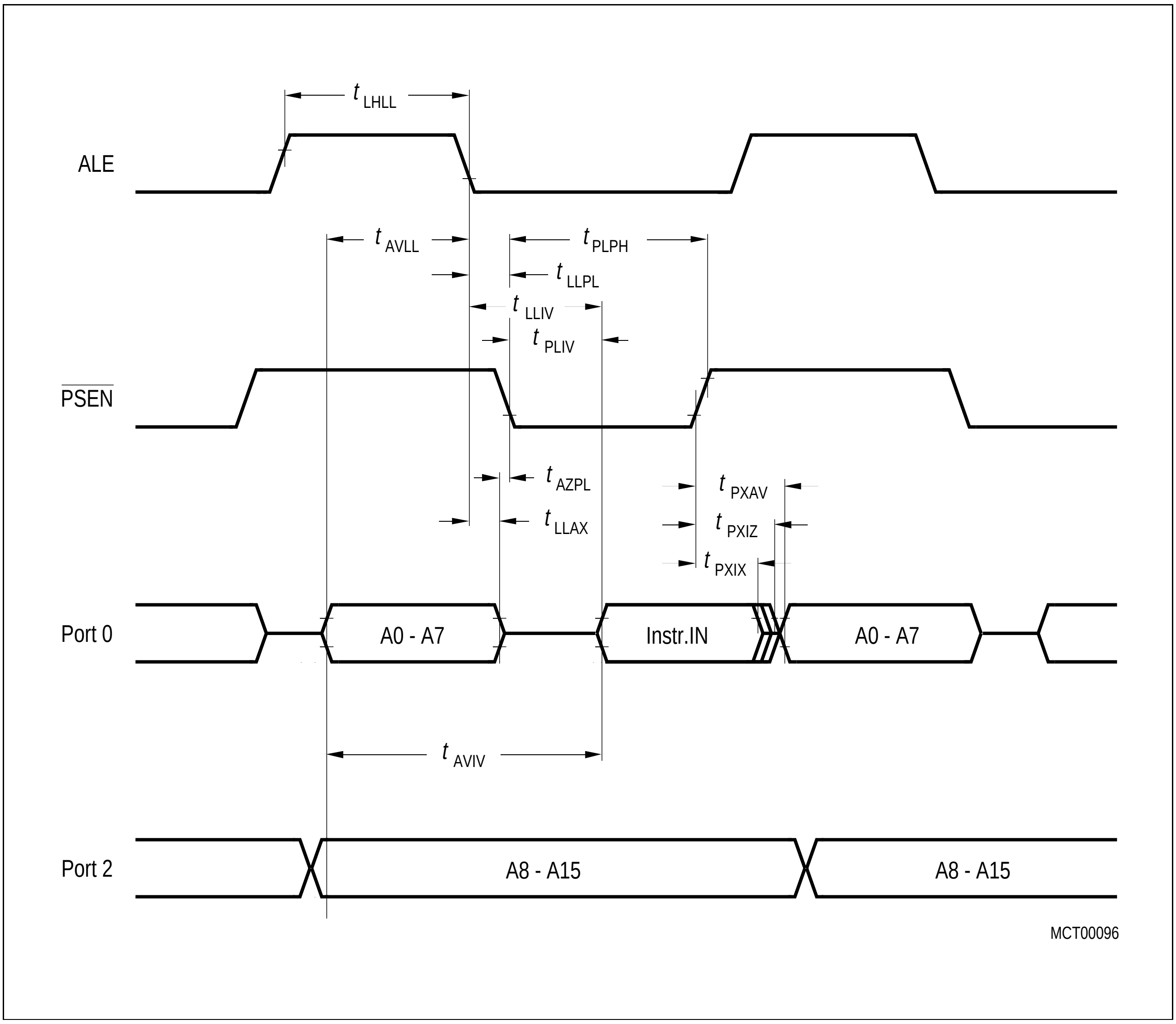


Figure 10-1
Program Memory Read Cycle

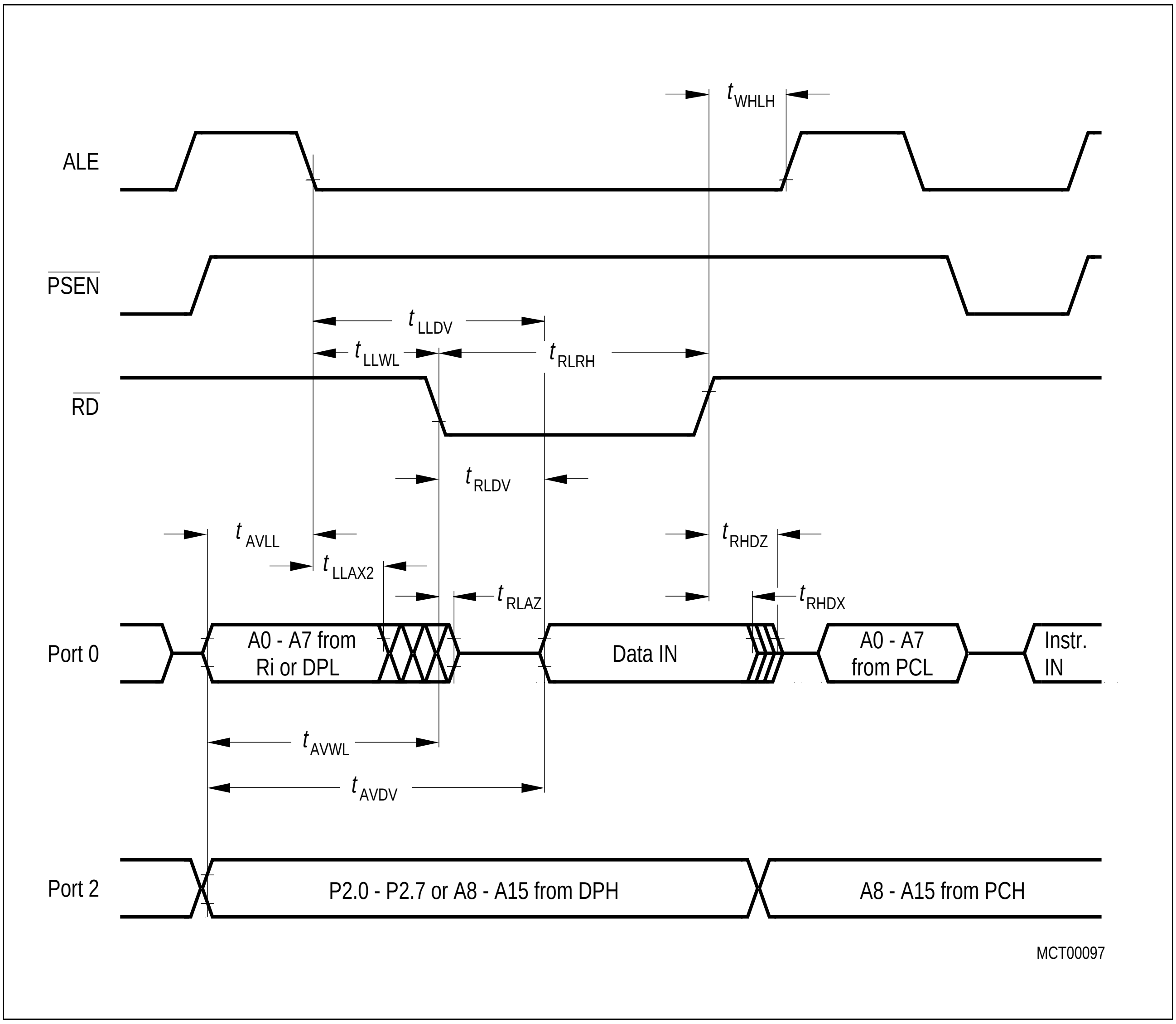


Figure 10-2
Data Memory Read Cycle

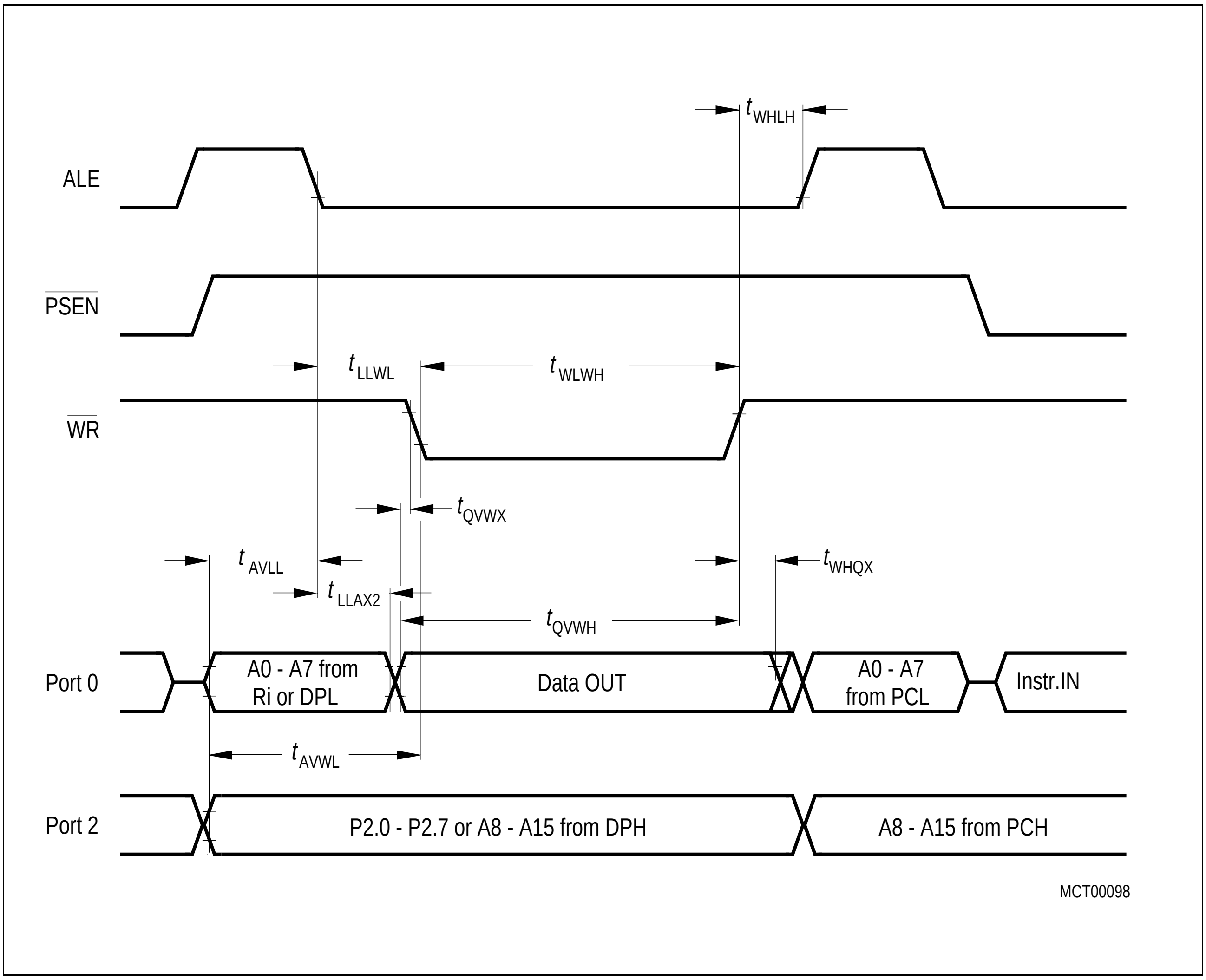


Figure 10-3
Data Memory Write Cycle

ROM Verification Characteristics for SAB-C502-2R

ROM Verification Mode 1

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

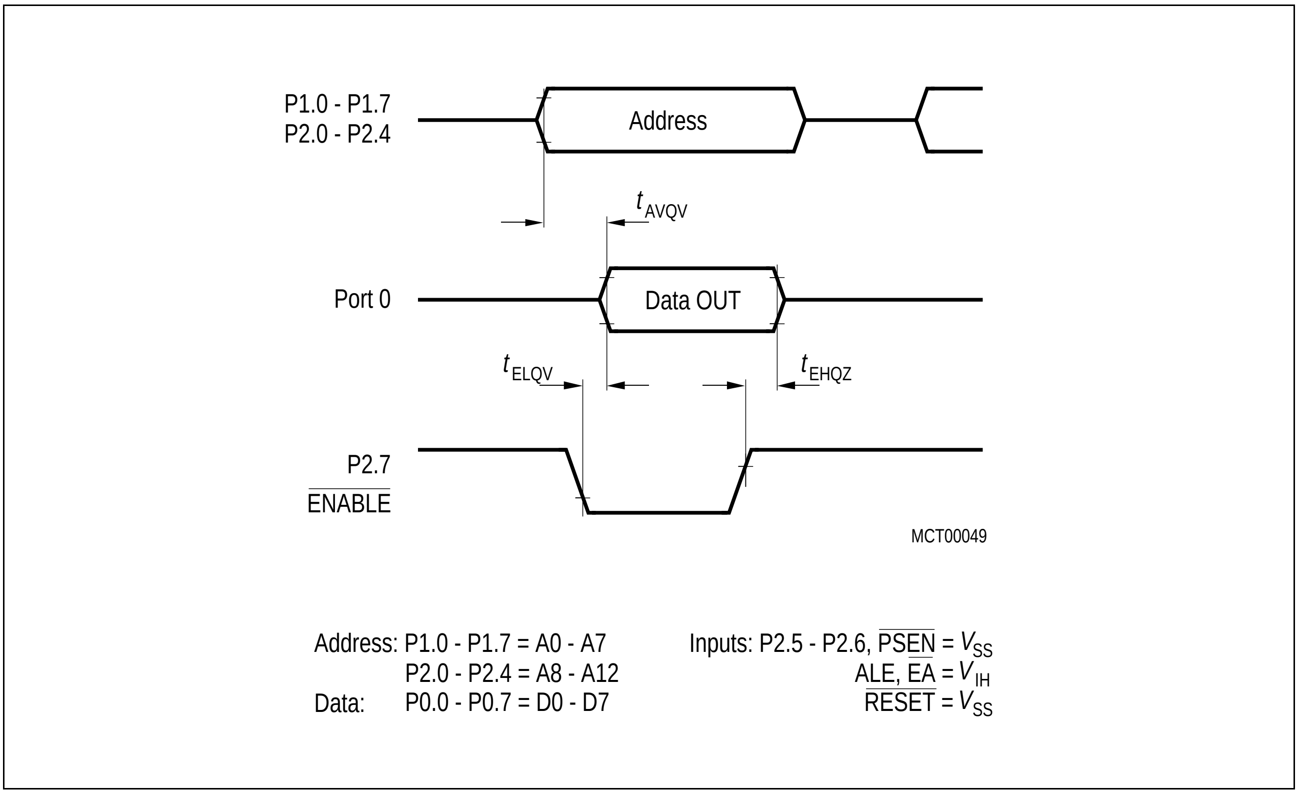
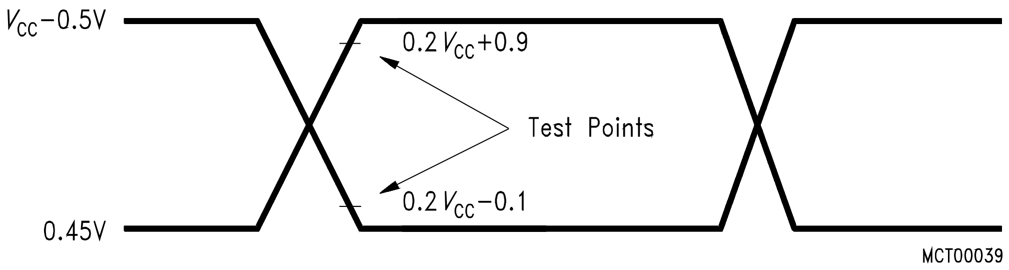
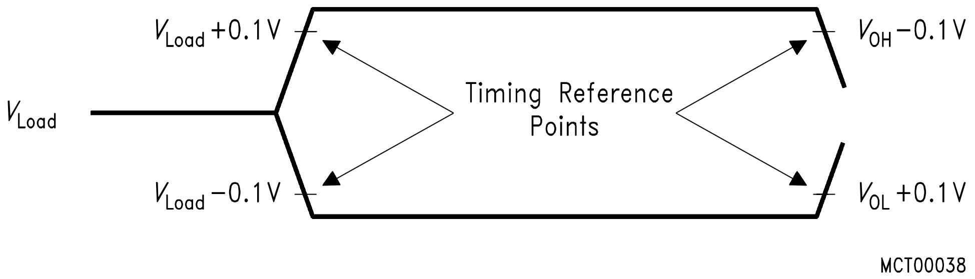

P1.0 - P1.7
P2.0 - P2.4
Address
t_{AVQV}
Port 0
Data OUT
t_{ELQV}
t_{EHQZ}
P2.7
$\overline{\text{ENABLE}}$
MCT00049
Address: P1.0 - P1.7 = A0 - A7
P2.0 - P2.4 = A8 - A12
Data: P0.0 - P0.7 = D0 - D7
Inputs: P2.5 - P2.6, $\overline{\text{PSEN}}$ = V_{SS}
ALE, $\overline{\text{EA}}$ = V_{IH}
$\overline{\text{RESET}}$ = V_{SS}

Figure 10-4
ROM Verification Mode 1



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

Figure 10-5
AC Testing: Input, Output Waveforms



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

Figure 10-6
AC Testing: Float Waveforms

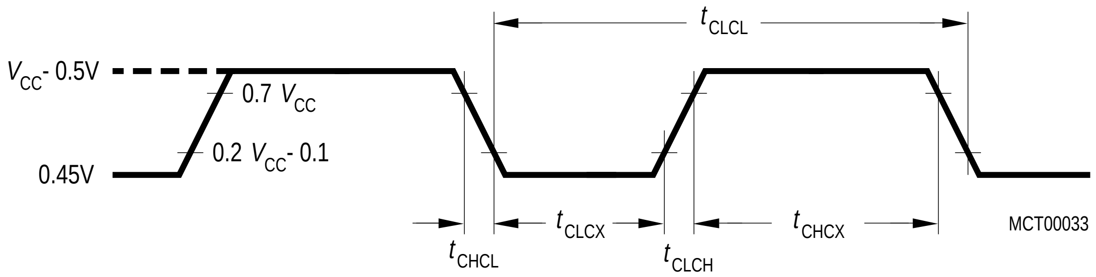


Figure 10-7
External Clock Cycle

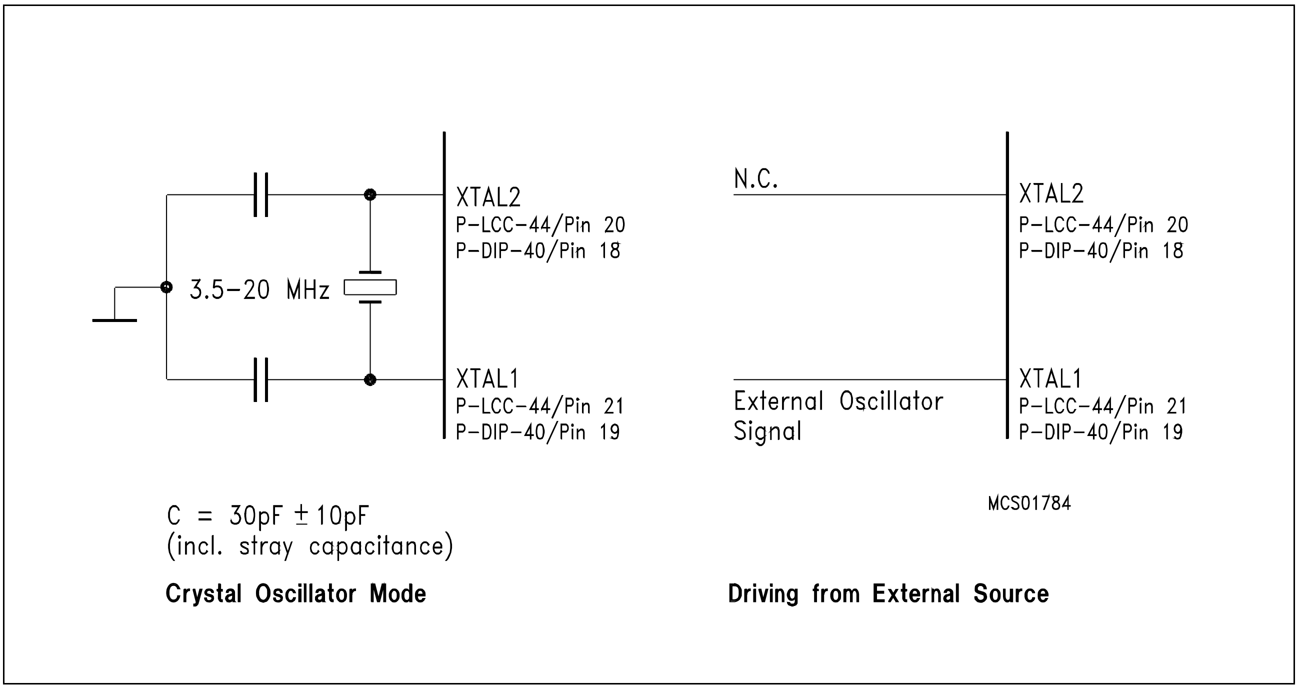
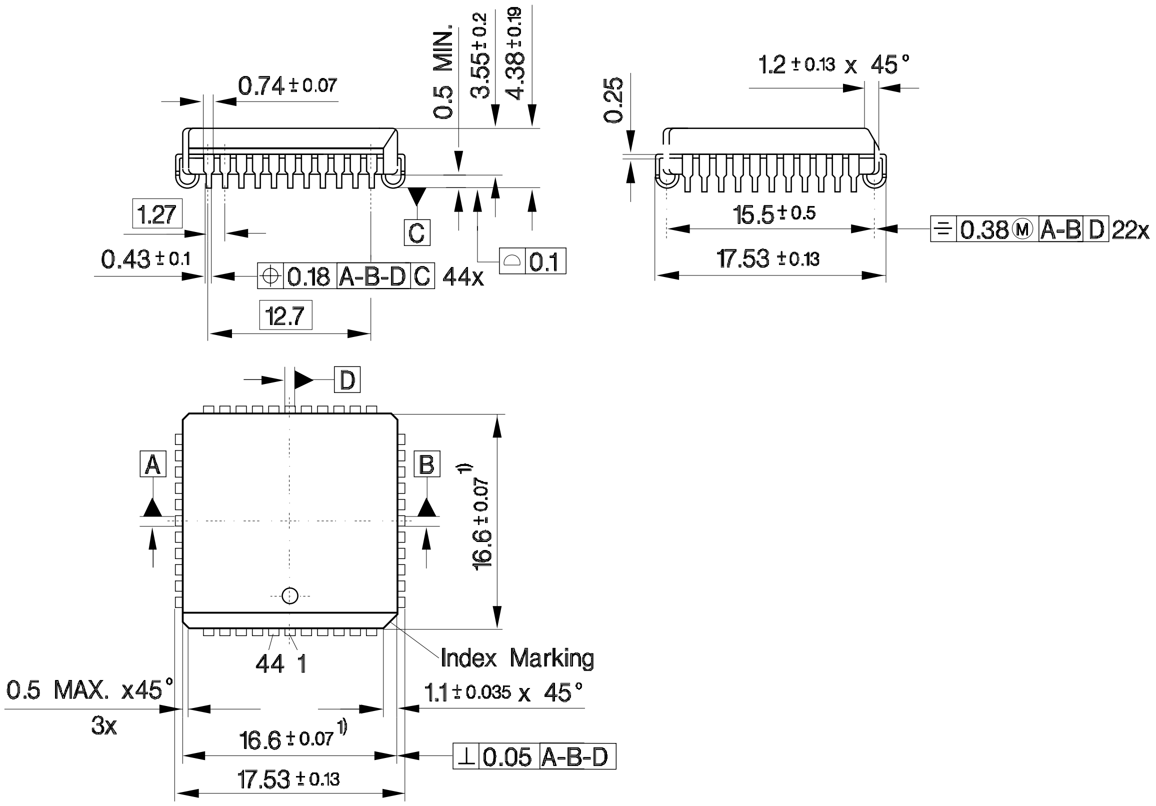


Figure 10-8
Recommended Oscillator Circuits

Package Outlines

P-LCC-44-1
(Plastic Leaded Chip Carrier)



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

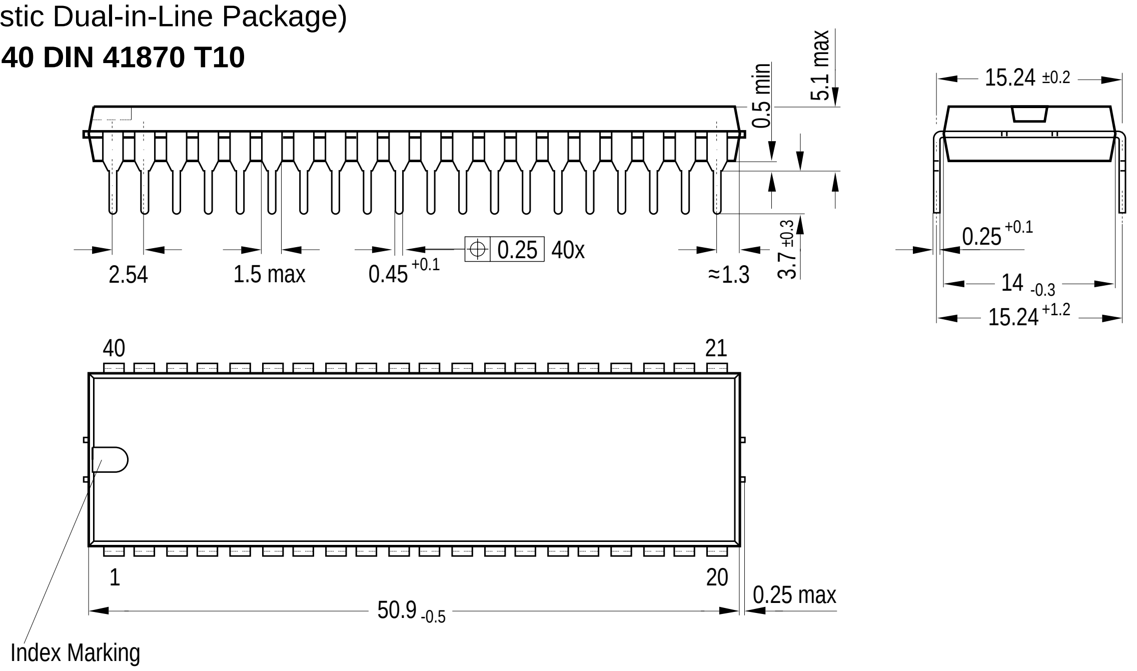
Sorts of Packing

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

SMD = Surface Mounted Device

Dimensions in mm

Plastic Package, P-DIP-40
(Plastic Dual-in-Line Package)
20B40 DIN 41870 T10



Sorts of Packing

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

SMD = Surface Mounted Device

Dimensions in mm