User's Manual, Nov. 1999

C505L

8-Bit CMOS Microcontroller

Microcontrollers

Infineon technologies

Never stop thinking.

Edition 10.99

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

Attention please!

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

Terms of delivery and rights to technical change reserved.

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

Infineon Technologies is an approved CECC manufacturer.

Information

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

Warnings

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

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

User's Manual, Nov. 1999

C505L

8-Bit CMOS Microcontroller

Microcontrollers

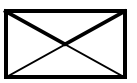
Never stop thinking.

C505L User's Manual
Revision History: 10.99

Previous Version: 04.99

Page	Subjects (major changes since last revision)
	Minor changes on title pages
several	V_{CC} is replaced by V_{DD}.
Table 3-3	Note 6 added.
Table 3-5	Note 2 added.
Figure 5-3, Figure 5-4, Figure 5-5	Figures changed.
Figure 6-40	New version of Figure imported.
Figure 7-1	New version of Figure imported.
Page 8-1	New variable time-out period for programmable watchdog timer.
Page 8-6	3rd indent text title in bold: external is removed.
Chapter 10	Throughout this chapter the term C505l-4E is replaced by C505L.
Chapter 10	Pages 10-7 to 10-14 were missing, now included.

We Listen to Your Comments
Any information within this document that you feel is wrong, unclear or missing? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:
mcdocu.comments@infineon.com



Contents Page

Contents **Page**

Contents **Page**

Contents **Page**

1 Introduction

The C505L microcontroller is a member of the Infineon Technologies C500 family of 8-bit microcontrollers. The C505 is fully compatible to the standard 8051 microcontroller. Additionally the C505L provides a 128-segment Liquid-Crystal Display (LCD) controller, real-rime clock, a 10-bit A/D Converter, on-chip RAM, and 32 Kbytes of on-chip OTP memory, extended power save provisions and RFI related improvements. With a maximum external clock rate of 20 MHz it achieves a 300 ns instruction cycle time.

The C505L contains a 32k × 8 one time programmable (OTP) program memory. This device operates with internal program memory only.

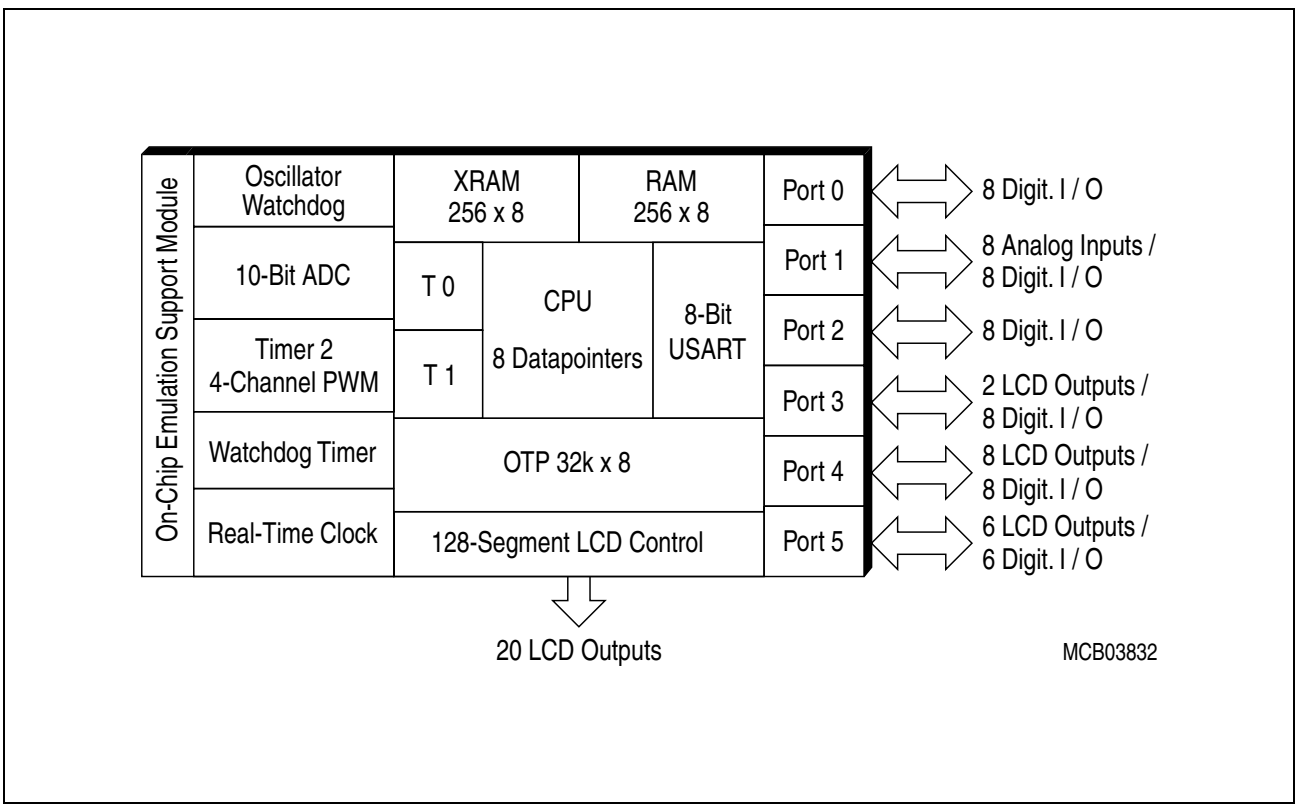


Figure 1-1
C505L Functional Units

Listed below is a summary of the main features of the C505L family:

- Fully compatible with the standard 8051 microcontroller
- Superset of the 8051 architecture with 8 datapointers
- Up to 20 MHz operating frequency
  - 375 ns instruction cycle time @ 16 MHz
  - 300 ns instruction cycle time @ 20 MHz (50% duty cycle)
- Program Memory
  - 32K bytes of on-chip OTP memory
  - Externally expandable up to 64 Kbytes
- 256-byte on-chip RAM
- 256-byte on-chip XRAM
- Five 8-bit and one 6-bit digital I/O ports (Port 5 with 6 bits only)
  - Port 1 with mixed analog/digital I/O capability
  - Port 3 with 2 LCD output lines as secondary functions
  - Port 4 and 5 with 8 and 6 LCD output lines respectively as secondary functions
- Three 16-bit timers/counters
  - Timer 0 / 1 (C501 compatible)
  - Timer 2 with 4 channels for 16-bit capture/compare operation
- 128-segment LCD Controller
  - 1/4 duty cycle drive
  - 4 row and 32 column outputs
  - On-chip programmable reference voltage generation
  - 20 dedicated LCD output lines (4 rows + 16 columns)
- Real-Time Clock
  - 47-bit digital clock counter
  - Input frequency of 32.768 kHz required
  - Operates in a special power down mode
- Full duplex serial interface with programmable baudrate generator (USART)
- 10-bit A/D Converter with 8 multiplexed inputs
- Twelve interrupt sources with four priority levels
- On-chip emulation support logic (Enhanced Hooks™ 1))
- Programmable 15-bit Watchdog Timer
- Oscillator Watchdog
- Fast power-on reset
- Power-saving modes
  - Slow-down mode
  - Idle mode (can be combined with slow-down mode)
  - 3 special power down modes
  - Software power-down mode with wake up capability through $\overline{\text{INT0}}$ pin or Real-Time Clock
- P-MQFP-80 package
- Temperature ranges:

SAB-C505L	T_A = 0 to 70 °C
SAF-C505L	T_A = – 40 to 85 °C
SAH-C505L	T_A = – 40 to 110 °C (max. operating frequency: 12 MHz)
SAK-C505L	T_A = – 40 to 125 °C (max. operating frequency: 12 MHz)

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

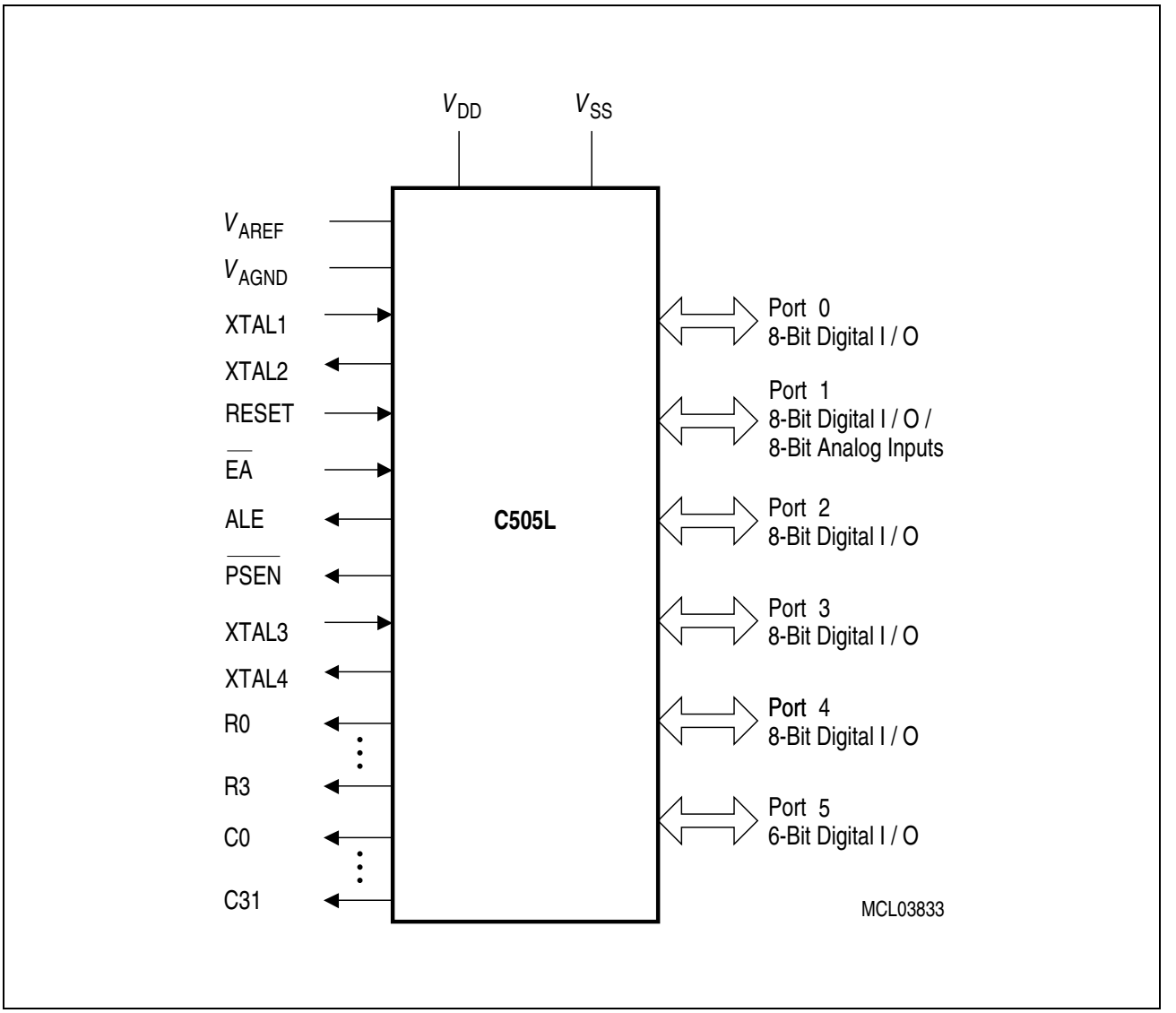


Figure 1-2
Logic Symbol

1.1 Pin Configuration

This section shows the pin configuration of the C505L in the P-MQFP-80 package.

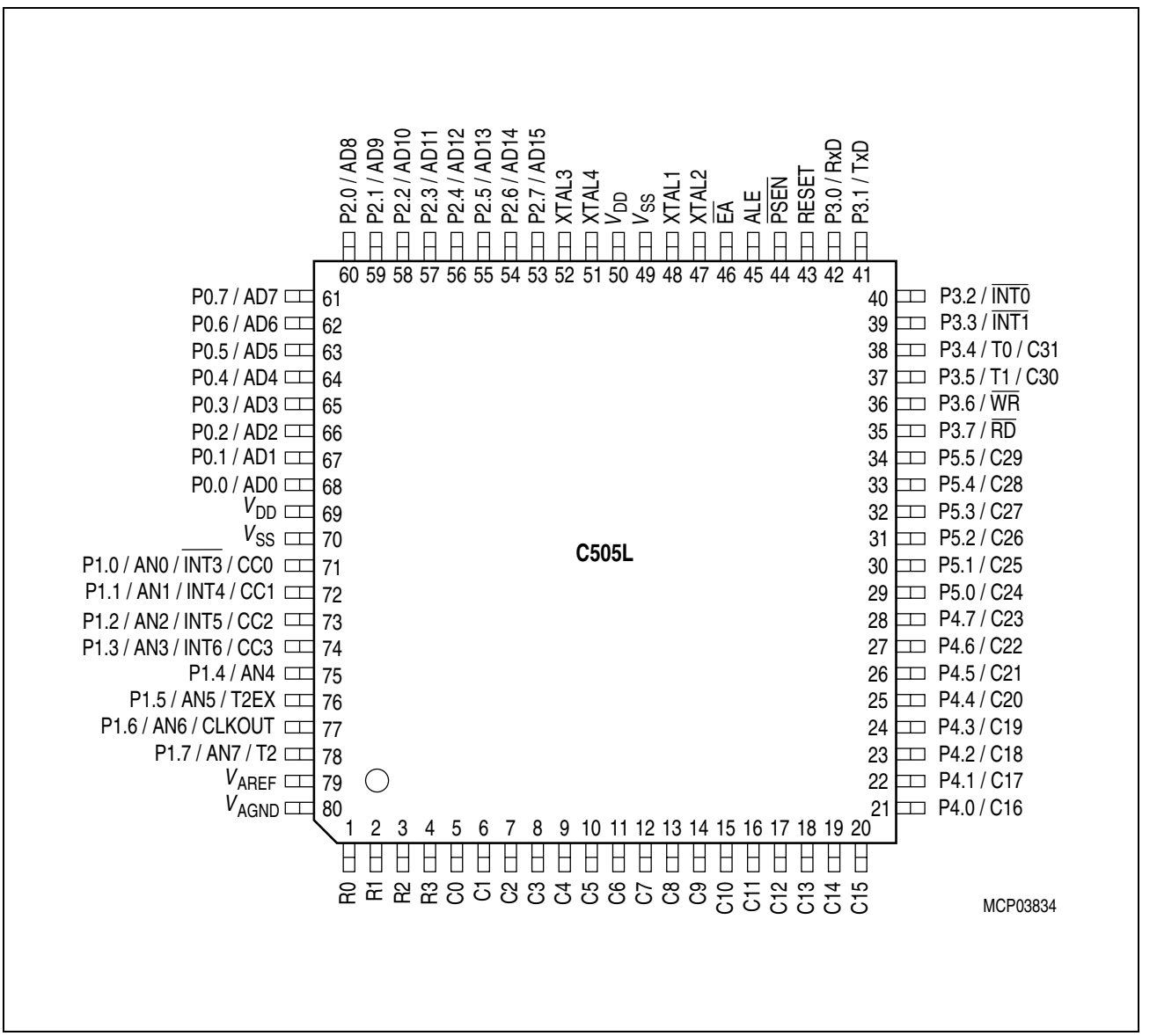


Figure 1-3
Pin Configuration, P-MQFP-80-1 Package (top view)

1.2 Pin Definitions and Functions

This section describes all external signals and functions of the C505L.

Table 1-1
Pin Definitions and Functions

Symbol	Pin Number	I/O*)	Function
R0-R3	1-4 1 2 3 4	O	**LCD Row Outputs** Output of LCD controller row lines. These pins are driven by the LCD controller and drive the row input lines of the external LCD display. Enabling the LCD Controller makes these pins available for LCD output levels. R0 LCD row output 0 R1 LCD row output 1 R2 LCD row output 2 R3 LCD row output 3 These pins should not be used for input.
C0-C15	5-20 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	O	**LCD Column Outputs** Output of LCD controller column lines 0 to 15. These pins are driven by the LCD controller and drive the column input lines of the external LCD display. Enabling the LCD controller makes these pins available for LCD output levels. C0 LCD column output 0 C1 LCD column output 1 C2 LCD column output 2 C3 LCD column output 3 C4 LCD column output 4 C5 LCD column output 5 C6 LCD column output 6 C7 LCD column output 7 C8 LCD column output 8 C9 LCD column output 9 C10 LCD column output 10 C11 LCD column output 11 C12 LCD column output 12 C13 LCD column output 13 C14 LCD column output 14 C15 LCD column output 15 These pins should not be used for input.

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
P4.0-P4.7	21-28	I/O	**Port 4** is a 8-bit quasi-bidirectional port with internal pull-up arrangement. Port 4 pins that have a 1 written to them are pulled high by the internal pull-up transistors and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup transistors. Port 4 pins can also be configured as LCD column outputs. The secondary functions are assigned to the pins of port 4 as follows: 21 P4.0 / C16 LCD column output 16 22 P4.1 / C17 LCD column output 17 23 P4.2 / C18 LCD column output 18 24 P4.3 / C19 LCD column output 19 25 P4.4 / C20 LCD column output 20 26 P4.5 / C21 LCD column output 21 27 P4.6 / C22 LCD column output 22 28 P4.7 / C23 LCD column output 23 These pins should not be used for input when configured as LCD output pins.
P5.0-P5.5	29-34	I/O	**Port 5** is a 6-bit quasi-bidirectional port with internal pull-up arrangement. Port 5 pins that have a 1 written to them are pulled high by internal pull-up transistors and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup transistors. Port 5 pins can also be configured as LCD column outputs. The secondary functions are assigned to the pins of port 5 as follows: 29 P5.0 / C24 LCD column output 24 30 P5.1 / C25 LCD column output 25 31 P5.2 / C26 LCD column output 26 32 P5.3 / C27 LCD column output 27 33 P5.4 / C28 LCD column output 28 34 P5.5 / C29 LCD column output 29 These pins should not be used for input when configured as LCD output pins.

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
P3.7-P3.0	35-42	I/O	**Port 3** is an 8-bit quasi-bidirectional port with internal pull-up arrangement. Port 3 pins that have a 1 written to them are pulled high by the internal pull-up transistors and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup transistors. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except for TxD and $\overline{WR}$). P3.4 and P3.5 can also be configured as LCD column outputs C31 and C30 respectively. These pins should not be used for input when configured as LCD output pins. The secondary functions are assigned to the pins of port 3 as follows: P3.0 / RxD — Receiver data input (asynch.) or data input/output (synch.) of serial interface P3.1 / TxD — Transmitter data output (asynch.) or clock output (synch.) of serial interface P3.2 / $\overline{INT0}$ — External interrupt 0 input / timer 0 gate control input P3.3 / $\overline{INT1}$ — External interrupt 1 input / timer 1 gate control input P3.4 / T0 / C31 — Timer 0 counter input / LCD column 31 output P3.5 / T1 / C30 — Timer 1 counter input / LCD column 30 output P3.6 / $\overline{WR}$ — $\overline{WR}$ control output; latches the data byte from port 0 into the external data memory P3.7 / $\overline{RD}$ — $\overline{RD}$ control output; enables the external data memory
	42		
	41		
	40		
	39		
	38		
	37		
	36		
	35		

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
RESET	43	I	**RESET** A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits power-on reset using only an external capacitor to V_{DD}.
PSEN	44	O	The **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 three oscillator periods except during external data memory accesses. Remains high during internal program execution. This pin should not be driven during reset operation.
ALE	45	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 three oscillator periods except during an external data memory access. When instructions are executed from internal program memory ($\overline{EA}$ = 1), the ALE generation can be disabled by bit EALE in SFR SYSCON. This pin should not be driven during reset operation.
EA	46	I	**External Access Enable** This pin must be held at high level. Instructions are fetched from the internal OTP memory when the PC is less than 8000_H. Instructions are fetched from external program memory, when the PC is greater than $7FFF_H$. This pin must not be held at low level.

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
XTAL2	47	O	**XTAL2** Output of the inverting oscillator amplifier.
XTAL1	48	I	**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. To operate above a frequency of 16 MHz, a duty cycle of 50% should be maintained. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics (refer to Data Sheet) must be observed.
XTAL4	51	O	**XTAL4** Output of the inverting real-time clock oscillator amplifier.
XTAL3	52	I	**XTAL3** Input to the inverting real-time clock oscillator amplifier. To drive the real-time clock from an external clock source, XTAL3 should be driven, while XTAL4 is left unconnected. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics (refer to Data Sheet) must be observed.

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
P2.7-P2.0	53-60	I/O	**Port 2** is a an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 2 pins that have a 1 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 transistors when issuing 1 s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 issues the contents of the P2 special function register and uses only the internal pullup resistors.
P0.7-P0.0	61-68	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have a 1 written to them float, and in that state can be used as high-impendance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application it uses strong internal pullup transistors when issuing 1 s.

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
P1.0-P1.7	71-78	I/O	**Port 1** is an 8-bit quasi-bidirectional port with internal pull-up arrangement. Port 1 pins can be used for digital input/output or as analog inputs to the A/D converter. Port 1 pins that have a 1 written to them are pulled high by internal pull-up transistors and in that state can be used as inputs. As inputs, port 1 pins being pulled low externally will source current (I_{IL}, in the DC characteristics) because of the internal pullup transistors. Port 1 pins are assigned to be used as analog inputs via the register P1ANA. As secondary digital functions, port 1 contains the interrupt, timer, clock, capture and compare pins. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except for compare functions). The secondary functions are assigned to the pins of port 1 as follows: 71 P1.0 / AN0 / $\overline{\text{INT3}}$ / CC0 Analog input channel 0 interrupt 3 input / capture/compare channel 0 I/O 72 P1.1 / AN1 / INT4 / CC1 Analog input channel 1/ interrupt 4 input / capture/compare channel 1 I/O 73 P1.2 / AN2 / INT5 / CC2 Analog input channel 2 / interrupt 5 input / capture/compare channel 2 I/O 74 P1.3 / AN3 / INT6 / CC3 Analog input channel 3 interrupt 6 input / capture/compare channel 3 I/O 75 P1.4 / AN4 Analog input channel 4 76 P1.5 / AN5 / T2EX Analog input channel 5 / timer 2 external reload / trigger input 77 P1.6 / AN6 / CLKOUT Analog input channel 6 / system clock output 78 P1.7 / AN7 / T2 Analog input channel 7 / timer/counter 2 input

*) I = Input
O = Output

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

Symbol	Pin Number	I/O*)	Function
V_{AREF}	79	–	**Reference voltage** for the A/D converter.
V_{AGND}	80	–	**Reference ground** for the A/D converter.
V_{SS}	49, 70	–	**Ground** (0 V)
V_{DD}	50, 69	–	**Power Supply** (+ 5 V)

*) I = Input
O = Output

2 Fundamental Structure

The C505L is fully compatible with the architecture of the standard 8051/C501 microcontroller family. While maintaining all architectural and operational characteristics of the C501, the C505L incorporates a Central Processing Unit (CPU) with 8 datapointers, an 10-bit A/D converter, a 4-channel capture/compare unit, a 128-segment LCD controller unit, a real-time clock unit, an XRAM data memory as well as some enhancements in the Fail Save Mechanism Unit. **Figure 2-1** shows a block diagram of the C505L.

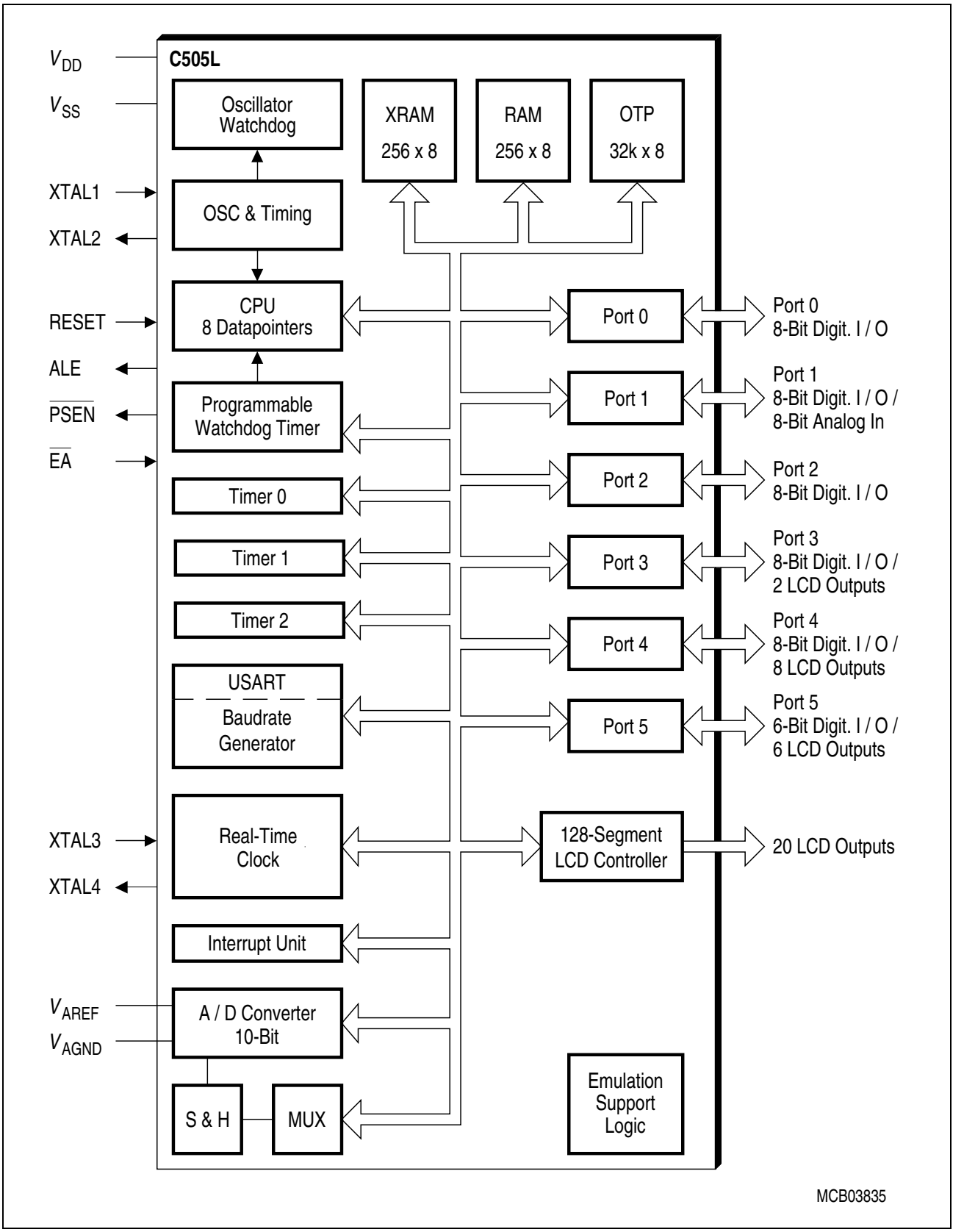


Figure 2-1
Block Diagram of the C505L

2.1 CPU

The C505L is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 16-MHz external clock, 58% of the instructions execute in 375 ns (20 MHz: 300 ns).

The CPU of the C505L 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 which 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 a Program Status Word (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, subtract, 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, complement, jump-if-set, 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.

The C505L, additionally, contains 8 datapointers compared to a standard 8051 microcontroller which has only one. For complex applications with peripherals (e.g. LCD controller) located in the external data memory space or extended data storage capacity this turned out to be a “bottle neck” for the 8051’s communication to the external world. Especially programming in high-level languages (PLM51, C51, 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.

Accumulator

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

Program Status Word

The PSW register contains several status bits that reflect the current state of the CPU.

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

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

Bit	Function
CY	Carry Flag Used by arithmetic instructions.
AC	Auxiliary Carry Flag Used by instructions which execute BCD operations.
F0	General Purpose Flag
RS1 RS0	Register Bank select control bits These bits are used to select one of the four register banks.
OV	Overflow Flag Used by arithmetic instructions.
F1	General Purpose Flag
P	Parity Flag Set/cleared by hardware after each instruction to indicate an odd/even number of “one” bits in the accumulator, i.e. even parity.

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

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 SP is initialized to 07_H after a reset. This causes the stack to begin a location = 08_H above register bank zero. The SP can be read or written under software control.

2.2 CPU Timing

The C505L has no clock prescaler. Therefore, a machine cycle of the C505L consists of 6 states (6 oscillator periods). Each state is divided into a phase 1 half and a phase 2 half. Thus, a machine cycle consists of 6 oscillator periods, numbered S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts one oscillator period. Typically, arithmetic and logic 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 XTAL1 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.

Execution of a one-cycle instruction begins at S1P2, when the opcode 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 opcode) 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 a 2-byte, 1-cycle instruction.

Most C505L 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 1-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.

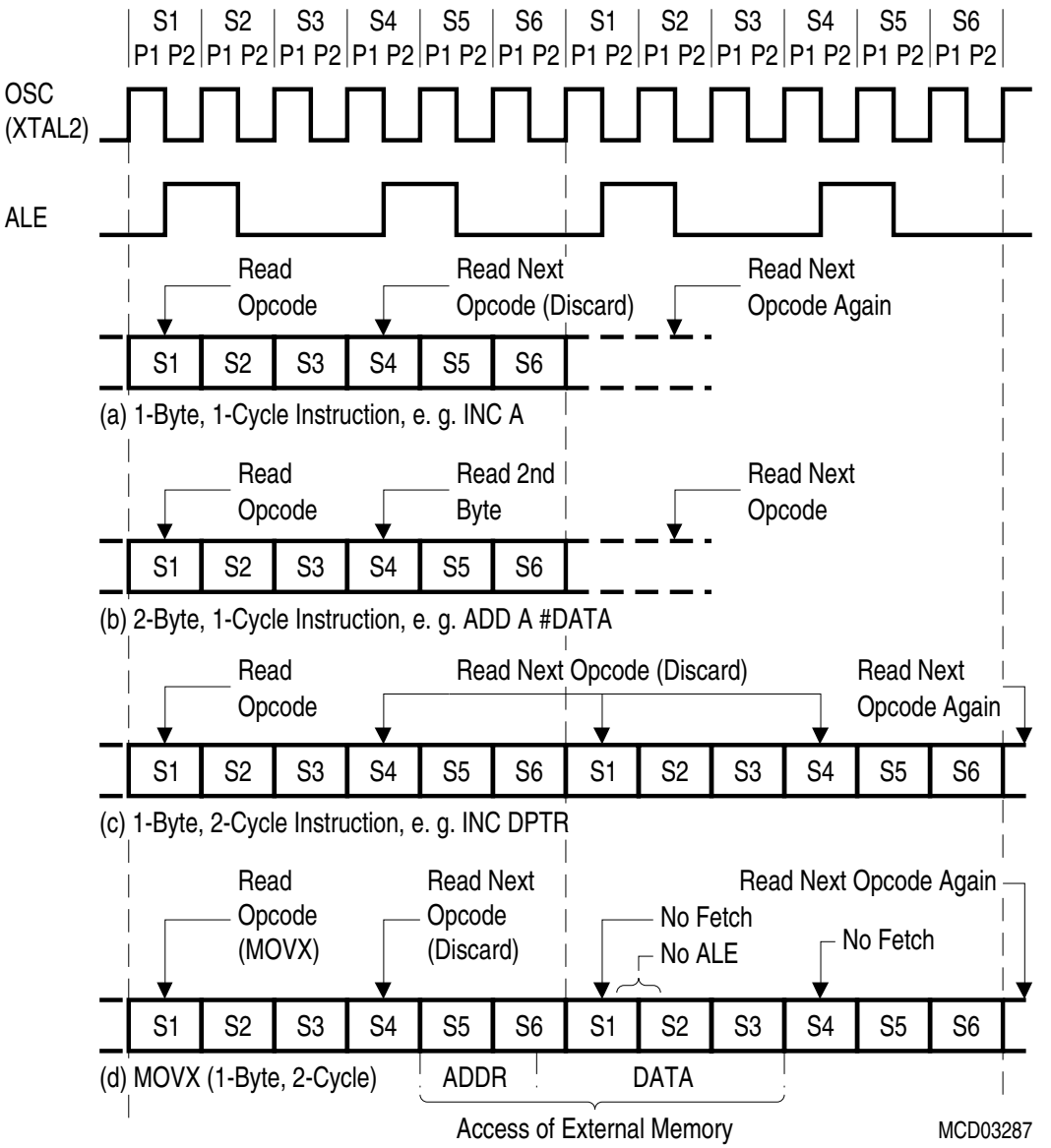


Figure 2-2
Fetch Execute Sequence

3 Memory Organization

The C505L CPU manipulates operands in the following address spaces:

- up to 64 Kbytes of program memory (32K on-chip OTP memory)
- up to 64 Kbytes of external data memory
- 256 bytes of internal data memory
- 256 bytes of internal XRAM data memory
- 20 bytes of LCD Controller registers
- 16 bytes of Real-Time Clock (RTC) registers
- A 128-byte Special Function Register (SFR) area

Figure 3-1 illustrates the memory address spaces of the C505L.

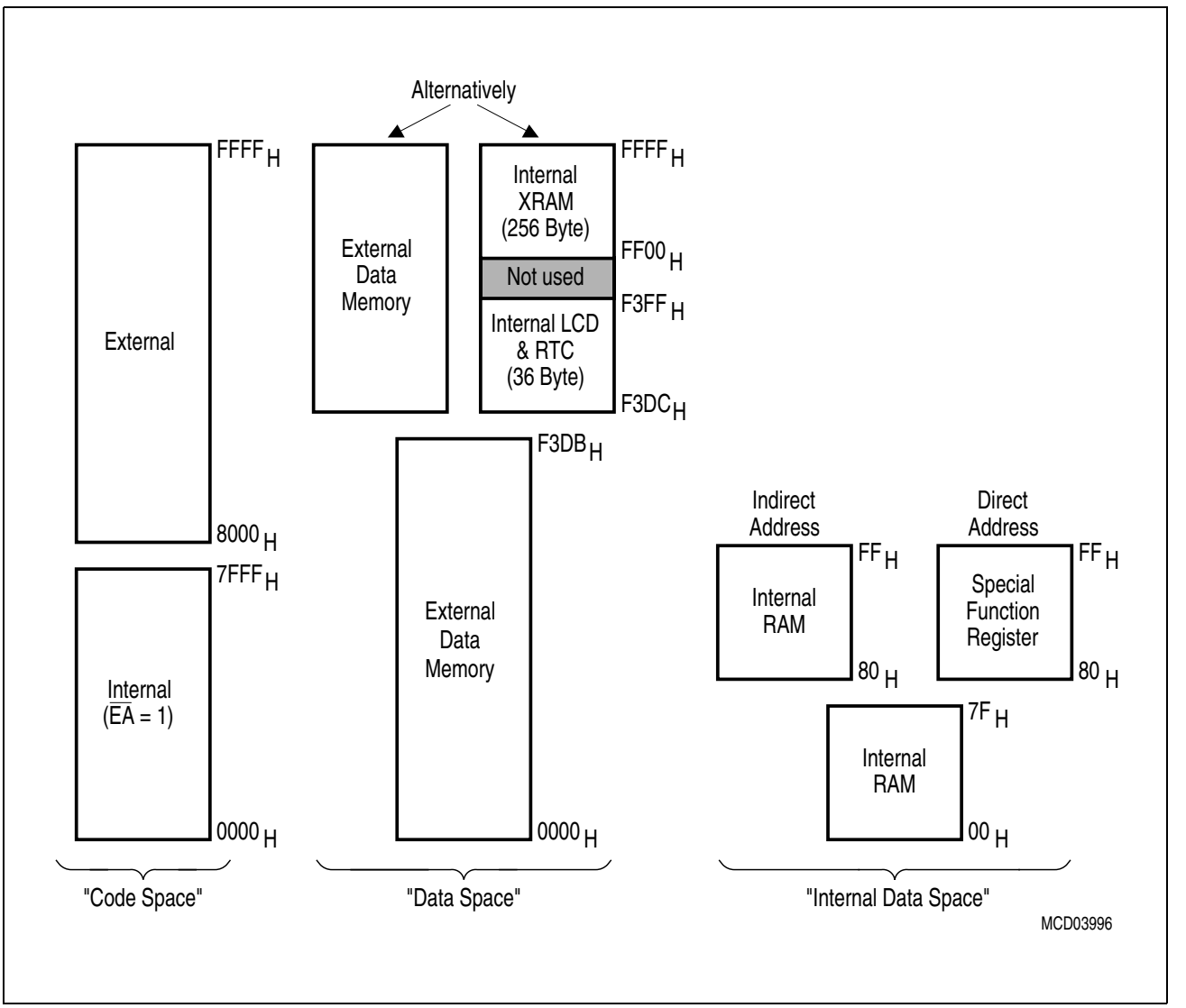


Figure 3-1
C505L Memory Map

3.1 Program Memory, “Code Space”

The C505L has 32 Kbytes of on-chip OTP memory which can be externally expanded up to 64 Kbytes. The C505L executes program code out of the internal OTP memory until the program counter address exceeds $7FFF_H$. Address locations 8000_H through $FFFF_H$ are then fetched from the external program memory. The $\overline{EA}$ pin is always held high. It is recommended to not set the **EA** pin at low level as this may cause the device to function in a manner that is not defined.

3.2 Data Memory, “Data Space”

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

The external data memory can be expanded up to 64 Kbytes, and can be accessed by instructions that use a 16-bit or an 8-bit address. The internal LCD controller, the RTC, both peripherals, and the internal XRAM are located in the external memory address area at addresses $F3DC_H$ to $F3EF_H$, $F3F0_H$ to $F3FF_H$ and $FF00_H$ to $FFFF_H$ respectively. The LCD controller registers, the RTC registers and internal XRAM can therefore be accessed using MOVX instructions with addresses pointing to the respective address areas.

3.3 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks of eight General Purpose Registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the Program Status Word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in **Chapter 2**). This allows fast context switching, which is useful when entering subroutines or interrupt service routines.

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

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

3.4 XRAM Operation

The XRAM in the C505L is a memory area that is logically located at the upper end of the external data memory space, but is integrated on the chip. Because the XRAM is used in the same way as external data memory, the same instruction types (MOVX), must be used for accessing the XRAM.

3.4.1 XRAM/LCD Controller/RTC Access Control

Two bits in SFR SYSCON, XMAP0 and XMAP1, control the accesses to XRAM, the LCD Controller and the RTC. XMAP0 is a general access enable/disable control bit, and XMAP1 controls the external signal generation during XRAM/LCD controller/RTC accesses.

Special Function Register SYSCON (Address $B1_H$) **Reset Value: $XX10XX01_B$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$B1_H$	–	–	EALE	RMAP	–	–	XMAP1	XMAP0	SYSCON

The shaded bits are not described in this section.

Bit	Function
XMAP1	XRAM/LCD Controller/RTC visible access control Control bit for $\overline{RD}$/$\overline{WR}$ signals during XRAM/LCD Controller/RTC accesses. If addresses are outside the XRAM/LCD Controller/RTC address range or if XRAM is disabled, this bit has no effect. XMAP1 = 0: The signals $\overline{RD}$ and $\overline{WR}$ are not activated during accesses to the XRAM/LCD Controller/RTC. XMAP1 = 1: Ports 0, 2 and the signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM/LCD Controller/RTC. In this mode, address and data information during XRAM/LCD Controller/RTC accesses are visible externally.
XMAP0	Global XRAM/LCD Controller/RTC access enable/disable control XMAP0 = 0: The access to XRAM, LCD Controller and RTC are enabled. XMAP0 = 1: The access to XRAM, LCD Controller and RTC are disabled (default after reset!). All MOVX accesses are performed via the external bus. Further, this bit is hardware protected.
–	Reserved bits for future use. Read by CPU returns undefined values.

When bit XMAP1 in SFR SYSCON is set, during all accesses to XRAM, LCD Controller and RTC $\overline{RD}$ and $\overline{WR}$ become active and port 0 and 2 drive the actual address/data information which is read/written from/to XRAM/LCD Controller/RTC. This feature allows to check the internal data transfers to XRAM, LCD Controller and the RTC. When port 0 and 2 are used for I/O purposes, the XMAP1 bit should not be set. Otherwise the I/O function of the port 0 and port 2 lines is interrupted.

After a reset operation, bit XMAP0 is set. This means that accesses to LCD Controller, RTC and the internal XRAM are generally disabled. In this case, all accesses using MOVX instructions within the address range of $F3DC_H$ to $F3EF_H$, $F3F0_H$ to $F3FF_H$ and $FF00_H$ to $FFFF_H$ generate external data memory bus cycles. When XMAP0 is cleared, accesses to LCD Controller, the RTC and the internal XRAM are enabled and all accesses using MOVX instructions with an address in the range as above will access the LCD Controller, RTC and the internal XRAM respectively. Internal accesses (XMAP0 = 0) in the address range gap from $F400_H$ to $FEFF_H$ (as shown in **Figure 3-1**) will have undefined data.

Bit XMAP0 is hardware-protected. If it is cleared once, it cannot be set by software. Only a reset operation will set the XMAP0 bit again. This hardware protection mechanism is done by an asymmetric latch at XMAP0 bit. An unintentional disabling of LCD Controller, the RTC and the internal XRAM could be dangerous since indeterminate values could be read from the external bus. To avoid this the XMAP0 bit is forced to ‘1’ only by a reset operation. Additionally, an internal capacitor is charged during reset. Therefore, the reset state is a disabled LCD Controller, disabled RTC and disabled internal XRAM. Because of the charge time of the capacitor, once the XMAP0 bit is written to ‘0’ (that is, discharging the capacitor) the bit cannot be set to ‘1’ again by software. On the other hand any distortion (software hang-up, noise, …) is not able to charge this capacitor, either. That is, the stable status is with the LCD Controller, the RTC and internal XRAM are enabled.

The “clear” instruction for the XMAP0 bit should be integrated in the program initialization routine before XRAM/LCD Controller/RTC is used. In extremely noisy systems the user may have redundant “clear” instructions.

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

The XRAM, LCD Controller and RTC can be accessed by two read/write instructions that use the 16-bit DPTR for indirect addressing. These instructions are:

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

For accessing the XRAM, the effective address stored in DPTR must be in the range of $FF00_H$ to $FFFF_H$. For accessing the LCD Controller, the effective address stored in DPTR must be in the range of $F3DC_H$ to $F3EF_H$. For accessing the RTC, the effective address stored in DPTR must be in the range of $F3F0_H$ to $F3FF_H$.

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

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

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

A special page register is implemented in the C505L to make it possible to access the XRAM/LCD Controller/RTC also with the MOVX @Ri instructions. XPAGE serves the same function for the XRAM, LCD Controller and RTC as Port 2 for external data memory.

Special Function Register XPAGE (Address 91_H) **Reset Value: 00_H**

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

Bit	Function
XPAGE.7-0	XRAM/LCD Controller/RTC high address XPAGE.7-0 is the address part A15-A8 when 8-bit MOVX instructions are used to access internal XRAM/LCD Controller/RTC.

Figures 3-2 to **3-4** show the dependencies of XPAGE- and Port 2 - addressing in order to explain the differences in accessing XRAM/LCD Controller/RTC, external RAM and to show what to do when Port 2 is used as an I/O-port.

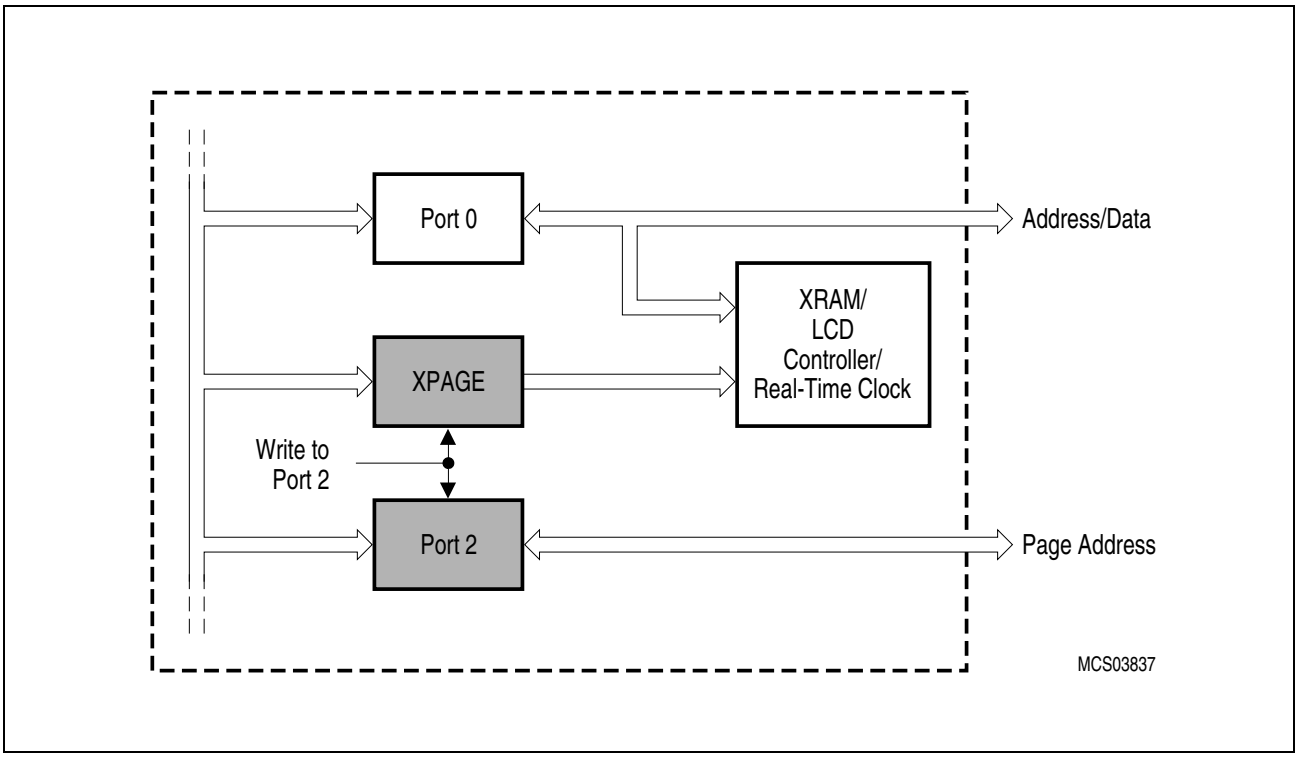


Figure 3-2
Write Page Address to Port 2

“MOV P2, pageaddress” will write the page address to port 2 and the XPAGE-Register.

When external RAM is to be accessed in the XRAM/LCD Controller/RTC address range, these modules should remain disabled after reset. When additional external RAM is to be addressed in an address range < $F3DC_H$, the XRAM/LCD Controller/RTC may remain enabled and there is no need to overwrite XPAGE by a second move.

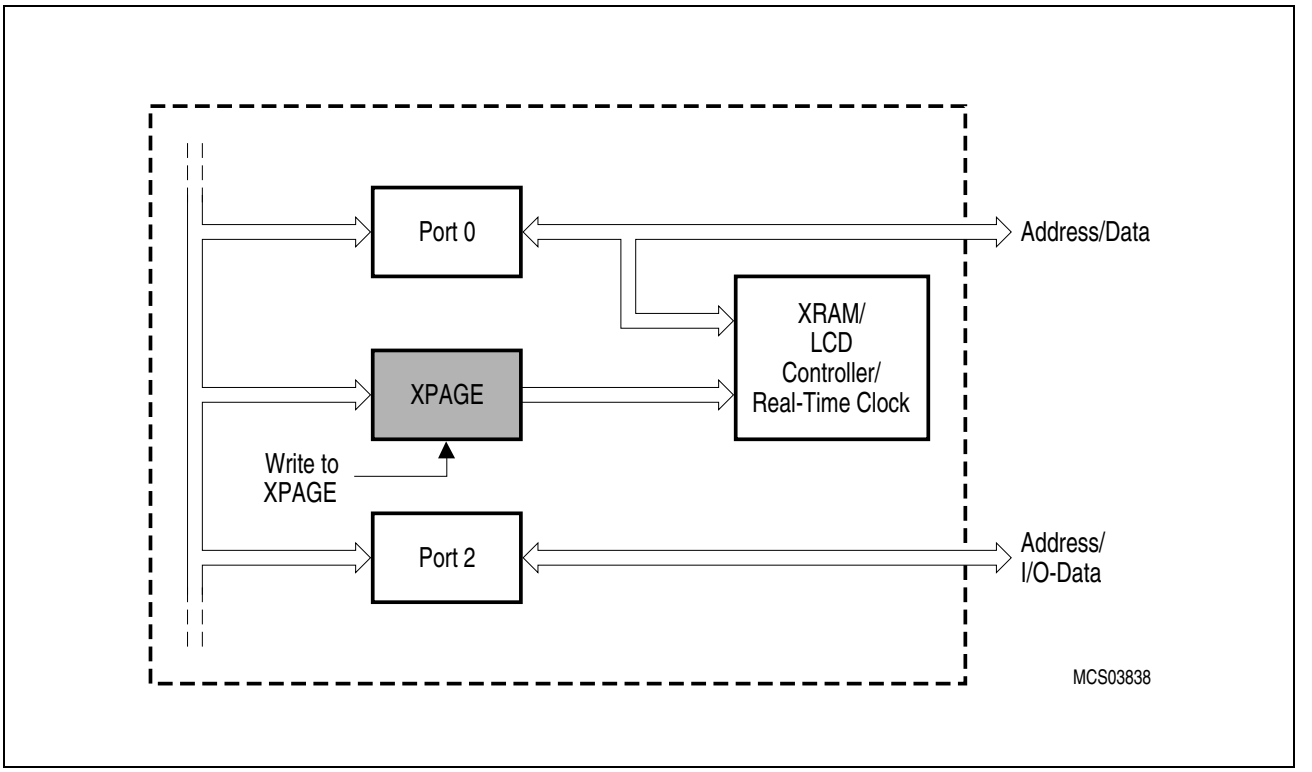


Figure 3-3
Write Page Address to XPAGE

“MOV XPAGE, pageaddress” will write the page address only to the XPAGE register. Port 2 is available for addresses or I/O data.

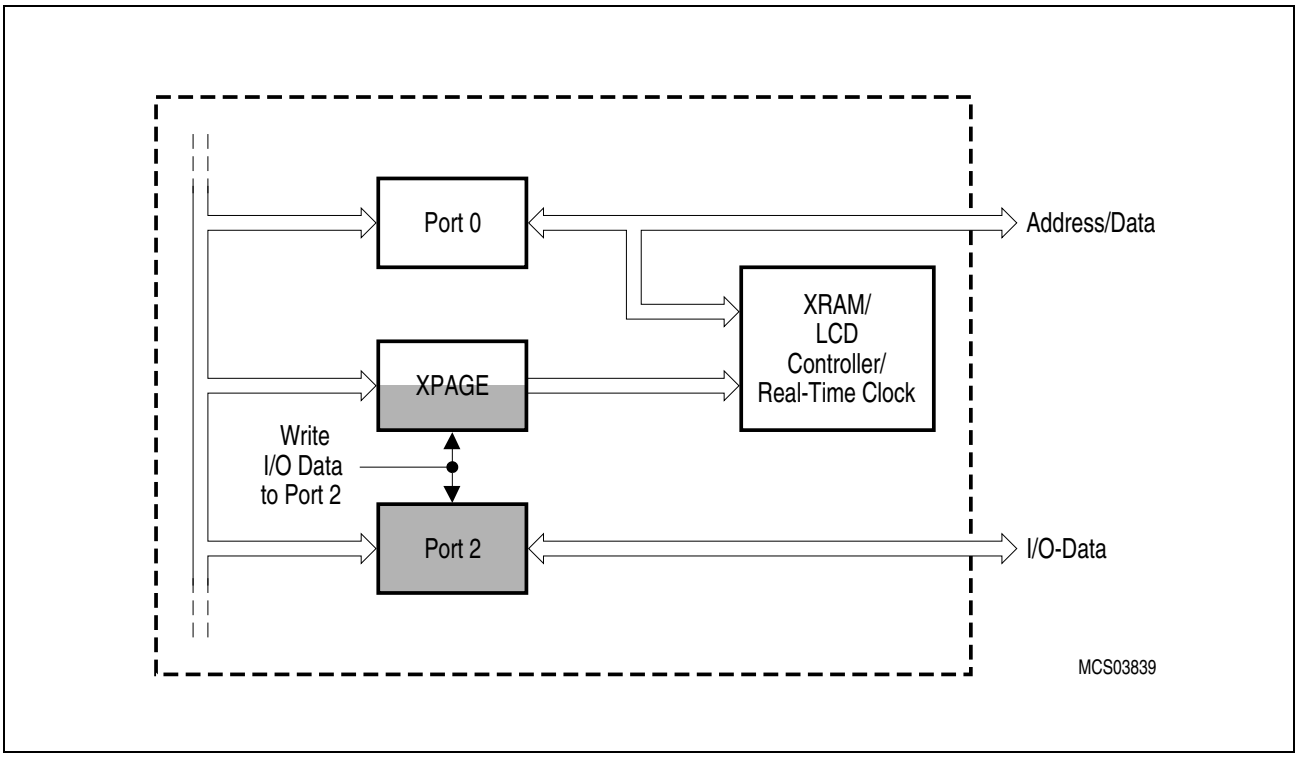


Figure 3-4
Use of Port 2 as I/O Port

On a write to port 2, the XRAM/LCD Controller/RTC address in XPAGE register will be overwritten because of the concurrent write to port 2 and XPAGE register. Therefore, 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 the page address.

Example:

I/O data at port 2 shall be AA_H. A byte shall be fetched from XRAM at address $FF30_H$.

```
MOV     R0, #30H          ;
MOV     P2, #0AAH         ; P2 shows AAH and XPAGE contains AAH
MOV     XPAGE, #0FFH      ; P2 still shows AAH but XRAM is addressed
MOVX    A, @R0            ; the contents of XRAM at FF30H 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/LCD Controller/RTC address range, an external access is performed. For the C505L the content of XPAGE must be $F7_H$ - FF_H in order to use the XRAM/LCD Controller/RTC.

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

a) Access to XRAM/LCD Contr./RTC: The upper address byte must be written to XPAGE or P2; both writes select the XRAM/LCD Controller/RTC address range.

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

3.4.4 Reset Operation of the XRAM

The contents of the XRAM are not affected by a reset. After power-up the contents are undefined, although they remain unchanged during and after a reset as long as the power supply is not turned off. If a reset occurs during a write operation to XRAM, the content of a XRAM memory location depends on the cycle in which the active reset signal is detected (MOVX is a 2-cycle instruction):

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

3.4.5 Behavior of Port 0 and Port 2

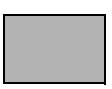
The behavior of port 0 and port 2 during a MOVX access depends on the control bits in register SYSCON. **Table 3-1** lists the various operating conditions. It shows the following characteristics:

a) Use of P0 and P2 pins during the MOVX access.
   Bus: The pins work as external address/data bus. If (internal) XRAM/LCD Controller/RTC are accessed, the data written to the XRAM/LCD Controller/RTC can be seen on the bus in debug mode.
   I/0: The pins work as Input/Output lines under control of their latch.
b) Activation of the **RD** and **WR** pin during the access.
c) Use of internal (XRAM/LCD Controller/RTC) or external XDATA memory.

The shaded areas in the table describe how each C5xx device without on-chip XRAM/LCD Controller/RTC behaves. For simplicity, the references in this table to the on-chip XRAM also cover the LCD Controller and the RTC accesses.

Table 3-2
Behavior of P0/P2 and $\overline{RD}$/$\overline{WR}$ During MOVX Accesses

		XMAP1, XMAP0		
		00	**10**	**X1**
MOVX @DPTR	DPTR < XRAM address range	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 ≥ XRAM address range	a) P0/P2→I/O b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0/P2→Bus ($\overline{RD}$/$\overline{WR}$-Data) 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 < XRAM addr. page range	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 ≥ XRAM addr.page range	a) P0/P2→I/O b) $\overline{RD}$/$\overline{WR}$ inactive c) XRAM is used	a) P0→Bus ($\overline{RD}$/$\overline{WR}$-Data) 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 8051/C501 family.

3.5 Special Function Registers

All registers, except for the program counter and the four GPR banks reside in the SFR area. The SFR area consists of two portions: the standard SFR area and the mapped SFR area. Some of the C505L's SFRs (PCON1, VR0, VR1 and VR2) are located in the mapped SFR area. For accessing the mapped SFR area, bit RMAP in SFR SYSCON must be set. All other SFRs are located in the standard SFR area which is accessed when RMAP is cleared ("0").

The registers and data locations of the LCD Controller (LCD-SFRs) and the RTC (RTC-SFRs) are located in the external data memory area at addresses $F3DD_H$ to $F3EF_H$ and $F3F0_H$ to $F3FF_H$ respectively. Details about the access of these registers is described in **Section 3.4.1** of this chapter.

Special Function Register SYSCON (Address $B1_H$) **Reset Value: $XX100X01_B$**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$B1_H$	–	–	EALE	RMAP	–	–	XMAP1	XMAP0	SYSCON

The shaded bits are not described in this section.

Bit	Function
RMAP	SFR map bit RMAP = 0: Access to the non-mapped (standard) SFR area is enabled. RMAP = 1: Access to the mapped SFR area is enabled.
–	Reserved bits for future use. Read by CPU returns undefined values.

As long as bit RMAP is set, mapped SFR area can be accessed. This bit is not cleared automatically by hardware. Thus, when non-mapped/mapped registers are to be accessed, the bit RMAP must be cleared/set respectively by software.

All SFRs with addresses where address bits 0-2 are 0 (e.g. 80_H, 88_H, 90_H, 98_H, …, $F8_H$, FF_H) are bit-addressable.

The 51 SFRs in the standard and mapped SFR area include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. The SFRs of the C505L are listed in **Table 3-3** and **Table 3-4**. In **Table 3-3** they are organized in groups which refer to the functional blocks of the C505L. The LCD and RTC-SFRs are also included in **Table 3-3**. **Table 3-4** illustrates the contents of the SFRs in numeric order of their addresses. **Table 3-5** lists the LCD and the RTC-SFRs in numeric order of their addresses.

Table 3-3
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
	SYSCON 2)	System Control Register	$B1_H$	$XX10XX01_B$ 3)
	VR0 4)	Version Register 0	FC_H	$C5_H$
	VR1 4)	Version Register 1	FD_H	85_H
	VR2 4)	Version Register 2	FE_H	5)
A/D-Converter	ADCON0 2)	A/D Converter Control Register 0	**$D8_H$** 1)	$00X00000_B$ 3)
	ADCON1	A/D Converter Control Register 1	DC_H	$01XXX000_B$ 3)
	ADDATH	A/D Converter Data Register High Byte	$D9_H$	00_H
	ADDATL	A/D Converter Data Register Low Byte	DA_H	$00XXXXXX_B$ 3)
	P1ANA 2)	Port 1 Analog Input Selection Register	90_H 4)	FF_H
Interrupt System	IEN0 2)	Interrupt Enable Register 0	**$A8_H$** 1)	00_H
	IEN1 2)	Interrupt Enable Register 1	**$B8_H$** 1)	00_H
	IP0 2)	Interrupt Priority Register 0	$A9_H$	00_H
	IP1	Interrupt Priority Register 1	$B9_H$	$XX000000_B$ 3)
	TCON 2)	Timer Control Register	**88_H** 1)	00_H
	T2CON 2)	Timer 2 Control Register	**$C8_H$** 1)	$00X00000_B$
	SCON 2)	Serial Channel Control Register	**98_H** 1)	00_H
	IRCON	Interrupt Request Control Register	**$C0_H$** 1)	00_H
XRAM	XPAGE	Page Address Register for Extended on-chip XRAM, LCD Controller and RTC	91_H	00_H
	SYSCON 2)	System Control Register	$B1_H$	$XX10XX01_B$ 3)
Ports	P0	Port 0	**80_H** 1)	FF_H
	P1	Port 1	**90_H** 1)	FF_H
	P1ANA 2)	Port 1 Analog Input Selection Register	**90_H** 1) 4)	FF_H
	P2	Port 2	**$A0_H$** 1)	FF_H
	P3	Port 3	**$B0_H$** 1)	FF_H
	P4	Port 4	**$E8_H$** 1)	00_B
	P5	Port 5	**$F8_H$** 1)	$XX111111_B$

1) Bit-addressable SFRs
2) This SFR is listed repeatedly since some bits of it also belong to other functional blocks.
3) “X” means that the value is undefined and the location is reserved.
4) This SFR is a mapped SFR. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
5) The content of this SFR varies with the actual step of the C505L (e.g. 01_H for the first step).

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

Block	Symbol	Name	Address	Contents after Reset
Serial Channel	ADCON0 2)	A/D Converter Control Register 0	**$D8_H$** 1)	$00X00000_B$ 3)
	PCON 2)	Power Control Register	87_H	00_H
	SBUF	Serial Channel Buffer Register	99_H	XX_H 3)
	SCON	Serial Channel Control Register	**98_H** 1)	00_H
	SRELL	Serial Channel Reload Register, low byte	AA_H	$D9_H$
	SRELH	Serial Channel Reload Register, high byte	BA_H	$XXXXXX11_B$ 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
Compare/ Capture Unit / Timer 2	CCEN	Comp./Capture Enable Reg.	$C1_H$	00_H 3)
	CCH1	Comp./Capture Reg. 1, High Byte	$C3_H$	00_H
	CCH2	Comp./Capture Reg. 2, High Byte	$C5_H$	00_H
	CCH3	Comp./Capture Reg. 3, High Byte	$C7_H$	00_H
	CCL1	Comp./Capture Reg. 1, Low Byte	$C2_H$	00_H
	CCL2	Comp./Capture Reg. 2, Low Byte	$C4_H$	00_H
	CCL3	Comp./Capture Reg. 3, Low Byte	$C6_H$	00_H
	CRCH	Reload Register High Byte	CB_H	00_H
	CRCL	Reload 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
	T2CON	Timer 2 Control Register	**$C8_H$** 1)	$00X00000_B$ 3)
	IEN0 2)	Interrupt Enable Register 0	**$A8_H$** 1)	00_H
	IEN1 2)	Interrupt Enable Register 1	**$B8_H$** 1)	00_H
Watchdog	WDTREL	Watchdog Timer Reload Register	86_H	00_H
	IEN0 2)	Interrupt Enable Register 0	**$A8_H$** 1)	00_H
	IEN1 2)	Interrupt Enable Register 1	**$B8_H$** 1)	00_H
	IP0 2)	Interrupt Priority Register 0	$A9_H$	00_H
Power Save Modes	PCON 2)	Power Control Register	87_H	00_H
	PCON1 4)	Power Control Register 1	**88_H** 1)	$0XX0XXXX_B$ 3)

1) Bit-addressable SFRs
2) This SFR is listed repeatedly since some bits of it also belong to other functional blocks.
3) "X" means that the value is undefined and the location is reserved
4) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.

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

Block	Symbol	Name	Address	Contents after Reset
LCD Controller	DAC0[6)]	D/A Conversion Register	$F3DC_H$	00_H
	LCON[6)]	LCD Control Register	$F3DD_H$	00_H
	LCRL[6)]	LCD Timer Reload Low Register	$F3DE_H$	00_H
	LCRH[6)]	LCD Timer Reload High Register	$F3DF_H$	00_H
	DIGn[5), 6)]	LCD Digit Register 'n' [5)]	$F3En_H$	00_H[5)]
Real-Time Clock	RTCON[6)]	Real-Time Clock Control Register	$F3F0_H$	00_H
	RTCR0[6)]	Real-Time Clock Initialization Register 0	$F3F1_H$	00_H
	RTCR1[6)]	Real-Time Clock Initialization Register 1	$F3F2_H$	00_H
	RTCR2[6)]	Real-Time Clock Initialization Register 2	$F3F3_H$	00_H
	RTCR3[6)]	Real-Time Clock Initialization Register 3	$F3F4_H$	00_H
	RTCR4[6)]	Real-Time Clock Initialization Register 4	$F3F5_H$	00_H
	CLREG0[6)]	Clock Count Register 0	$F3F6_H$	00_H
	CLREG1[6)]	Clock Count Register 1	$F3F7_H$	00_H
	CLREG2[6)]	Clock Count Register 2	$F3F8_H$	00_H
	CLREG3[6)]	Clock Count Register 3	$F3F9_H$	00_H
	CLREG4[6)]	Clock Count Register 4	$F3FA_H$	00_H
	RTINT0[6)]	Real-Time Clock Interrupt Register 0	$F3FB_H$	00_H
	RTINT1[6)]	Real-Time Clock Interrupt Register 1	$F3FC_H$	00_H
	RTINT2[6)]	Real-Time Clock Interrupt Register 2	$F3FD_H$	00_H
	RTINT3[6)]	Real-Time Clock Interrupt Register 3	$F3FE_H$	00_H
	RTINT4[6)]	Real-Time Clock Interrupt Register 4	$F3FF_H$	00_H

1) Bit-addressable SFRs
2) This SFR is listed repeatedly since some bits of it also belong to other functional blocks.
3) "X" means that the value is undefined and the location is reserved.
4) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
5) The notation "n" (n = 0 to F) in the LCD Digit Register address definition defines the number of the related LCD digit.
6) This register is located in the on-chip external data memory area.

Table 3-4
Contents of the SFRs, SFRs in Numeric Order of Their Addresses

Addr	Register	Content after Reset[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80_H[2]	P0	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
81_H	SP	07_H	.7	.6	.5	.4	.3	.2	.1	.0
82_H	DPL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
83_H	DPH	00_H	.7	.6	.5	.4	.3	.2	.1	.0
86_H	WDTREL	00_H	WDT PSEL	.6	.5	.4	.3	.2	.1	.0
87_H	PCON	00_H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE
88_H[2]	TCON	00_H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
88_H[3]	PCON1	$0XX0\text{-}XXXX_B$	EWPD	–	–	WS	–	–	–	–
89_H	TMOD	00_H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
$8A_H$	TL0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8B_H$	TL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8C_H$	TH0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8D_H$	TH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
90_H[2]	P1	FF_H	T2	CLK-OUT	T2EX	.4	.3	INT5	INT4	.0
90_H[3]	P1ANA	FF_H	EAN7	EAN6	EAN5	EAN4	EAN3	EAN2	EAN1	EAN0
91_H	XPAGE	00_H	.7	.6	.5	.4	.3	.2	.1	.0
92_H	DPSEL	$XXXX\text{-}X000_B$	–	–	–	–	–	.2	.1	.0
98_H[2]	SCON	00_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99_H	SBUF	XX_H	.7	.6	.5	.4	.3	.2	.1	.0
$A0_H$[2]	P2	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
$A8_H$[2]	IEN0	00_H	EA	WDT	ET2	ES	ET1	EX1	ET0	EX0
$A9_H$	IP0	00_H	OWDS	WDTS	.5	.4	.3	.2	.1	.0
AA_H	SRELL	$D9_H$	.7	.6	.5	.4	.3	.2	.1	.0

1) “X” means that the value is undefined and the location is reserved
2) Bit-addressable SFRs
3) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.

Table 3-4
Contents of the SFRs, SFRs in Numeric Order of Their Addresses (cont'd)

Addr	Register	Content after Reset[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$B0_H$[2]	P3	FF_H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD
$B1_H$	SYSCON	XX10-XX01$_B$	–	–	EALE	RMAP	–	–	XMAP1	XMAP0
$B8_H$[2]	IEN1	00_H	EXEN2	SWDT	EX6	EX5	EX4	EX3	ESWI	EADC
$B9_H$	IP1	XX00-0000$_B$	–	–	.5	.4	.3	.2	.1	.0
BA_H	SRELH	XXXX-XX11$_B$	–	–	–	–	–	–	.1	.0
$C0_H$[2]	IRCON	00_H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	SWI	IADC
$C1_H$	CCEN	00_H	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0
$C2_H$	CCL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C3_H$	CCH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C4_H$	CCL2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C5_H$	CCH2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C6_H$	CCL3	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C7_H$	CCH3	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C8_H$[2]	T2CON	00X0-0000$_B$	T2PS	I3FR	–	T2R1	T2R0	T2CM	T2I1	T2I0
CA_H	CRCL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CB_H	CRCH	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CC_H	TL2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
CD_H	TH2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$D0_H$[2]	PSW	00_H	CY	AC	F0	RS1	RS0	OV	F1	P
$D8_H$[2]	ADCON0	00X0-0000$_B$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0
$D9_H$	ADDATH	00_H	.9	.8	.7	.6	.5	.4	.3	.2
DA_H	ADDATL	00XX-XXXX$_B$	.1	.0	–	–	–	–	–	–

1) "X" means that the value is undefined and the location is reserved
2) Bit-addressable SFRs

Table 3-4
Contents of the SFRs, SFRs in Numeric Order of Their Addresses (cont'd)

Addr	Register	Content after Reset[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DC_H	ADCON1	01XX-X 000_B	ADCL1	ADCL0	–	–	–	MX2	MX1	MX0
$E0_H$[2]	ACC	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E8_H$[2]	P4	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F0_H$[2]	B	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F8_H$[2]	P5	XX00-00 00_H	–	–	.5	.4	.3	.2	.1	.0
FC_H[3)4)]	VR0	$C5_H$	1	1	0	0	0	1	0	1
FD_H[3)4)]	VR1	85_H	0	0	0	0	0	1	0	1
FE_H[3)4)]	VR2	[5]	.7	.6	.5	.4	.3	.2	.1	.0

1) "X" means that the value is undefined and the location is reserved.
2) Bit-addressable SFRs.
3) SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.
4) These are read-only registers.
5) The content of this SFR varies with the actual of the step C505L (e.g. 01_H for the first step).

Table 3-5
Contents of the LCD and the RTC Registers in Numeric Order of Their Addresses

Addr.	Register	Content after Reset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$F3DC_H$	DAC0[2)]	00_H	S7	S6	S5	S4	S3	S2	S1	S0
$F3DD_H$	LCON[2)]	00_H	DSB1	DSB0	0	0	0	0	CSEL	LCEN
$F3DE_H$	LCRL[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3DF_H$	LCRH[2)]	00_H	SLT	.14	.13	.12	.11	.10	.9	.8
$F3En_H$	DIGn[1), 2)]	00_H	SEGF	SEGA	SEGG	SEGB	SEGE	SEGC	SEGH	SEGD
$F3F0_H$	RTCON[2)]	00_H	0	0	0	0	RTPD	IRTC	ERTC	RTCS
$F3F1_H$	RTCR0[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F2_H$	RTCR1[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F3_H$	RTCR2[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F4_H$	RTCR3[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F5_H$	RTCR4[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F6_H$	CLREG0[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F7_H$	CLREG1[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F8_H$	CLREG2[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3F9_H$	CLREG3[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FA_H$	CLREG4[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FB_H$	RTINT0[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FC_H$	RTINT1[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FD_H$	RTINT2[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FE_H$	RTINT3[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3FF_H$	RTINT4[2)]	00_H	.7	.6	.5	.4	.3	.2	.1	.0

1) The notation “n” (n = 0 to F) in the LCD Digit Register address definition defines the number of the related LCD digit.
2) This register is located in the on-chip external data memory area.

4 External Bus Interface

The C505L allows for external memory expansion. The functionality and implementation of the external bus interface is identical to the common interface for the 8051 architecture with one exception: If the C505L is used in systems with no external memory the generation of the ALE signal can be suppressed. Resetting bit EALE in SFR SYSCON register, the ALE signal will be gated off. This feature reduces RFI emissions of the system.

4.1 Accessing External Memory

It is possible to distinguish between accesses to external program memory, external data memory or to 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).

4.1.1 Role of P0 and P2 as Data/Address Bus

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

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

Whenever a 16-bit address is used, the high byte of the address comes out on port 2, where it is held for the duration of the read or write cycle. During this time, the port 2 lines are disconnected from the port 2 latch (the SFR).

Thus the port 2 latch does not have to contain 1 s, 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.

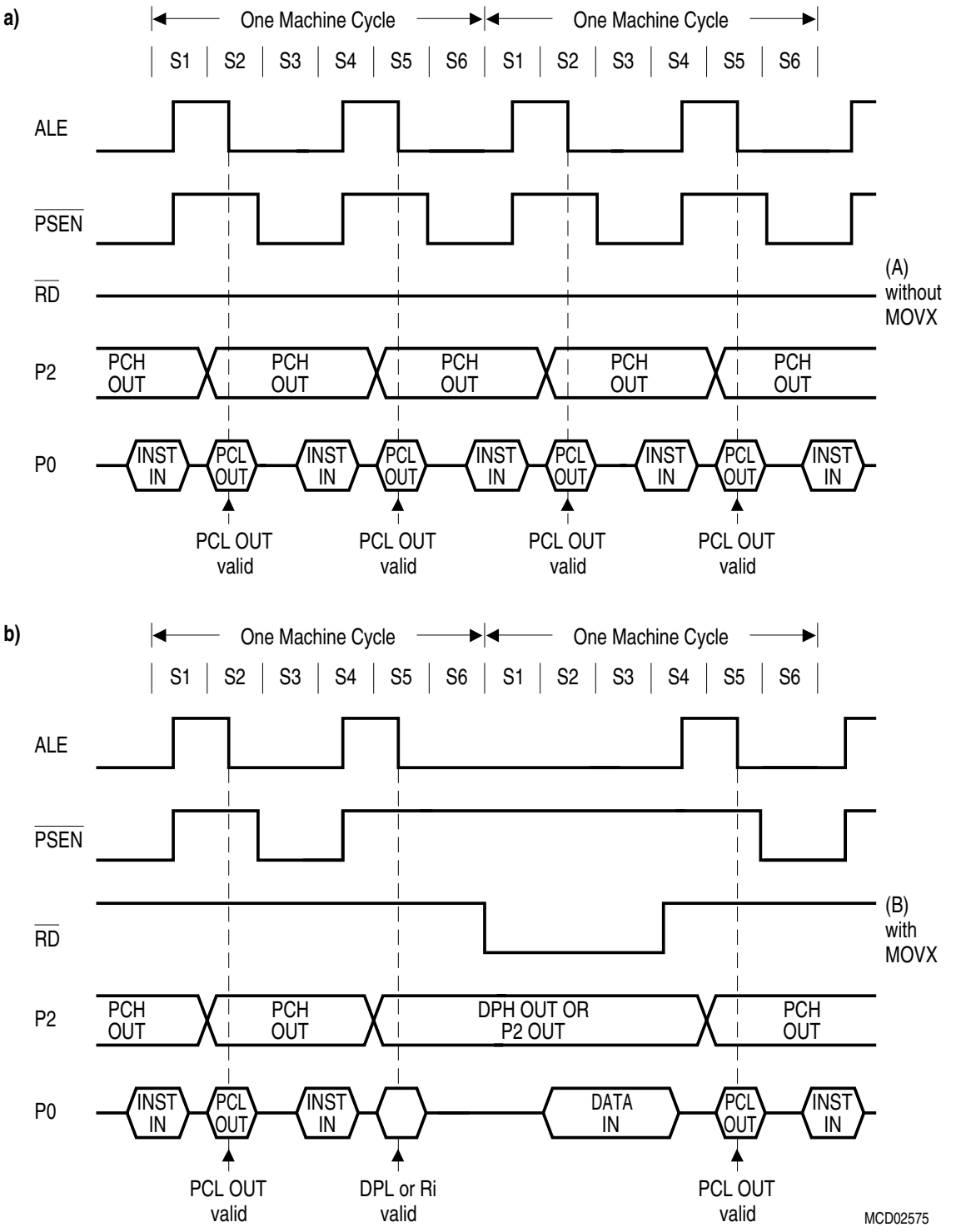


Figure 4-1
External Program Memory Execution

4.1.2 Timing

The timing of the external bus interface and the relationship between the control signals ALE, $\overline{\text{PSEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$ and information on port 0 and port 2 are illustrated in **Figures 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.

4.1.3 External Program Memory Access

The external program memory is accessed whenever the program counter (PC) content is greater than $7FFF_H$, provided the $\overline{\text{EA}}$ pin is held at high level at reset.

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

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

The read strobe for external program memory fetches is $\overline{\text{PSEN}}$. It is not activated for internal program memory fetches. When the CPU is accessing external program memory, $\overline{\text{PSEN}}$ is activated twice every instruction cycle (except during a MOVX instruction) 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 6 oscillator periods. A complete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$**,** takes 3 oscillator periods. The execution sequence for these two types of read cycles is shown in **Figures 4-1 a)** and **b)**.

4.3 Overlapping External Data and Program Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is used for storing data. In the C505L, the external program and data memory spaces can be combined by the logical-AND of $\overline{\text{PSEN}}$ and $\overline{\text{RD}}$. A positive result from this AND operation produces a low-active 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.

4.4 ALE, Address Latch Enable

The C505L allows to switch off the ALE output signal. If the internal OTP is used ($\overline{EA}$ = 1 and PC $\leq 7FFF_H$) and ALE is switched off by EALE = 0, then, ALE will only go active during external data memory accesses (MOVX instructions).

After a hardware reset, ALE generation is enabled.

Special Function Register SYSCON (Address $B1_H$) **Reset Value: $XX10XX01_B$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$B1_H$	–	–	EALE	RMAP	–	–	XMAP1	XMAP0	SYSCON

The shaded bits are not described in this section.

Bit	Function
EALE	Enable ALE output EALE = 0: ALE generation is disabled; disables ALE signal generation during internal code memory accesses; ALE is automatically generated during MOVX instructions EALE = 1: ALE generation is enabled (default after reset)
–	Reserved bits for future use. Read by CPU returns undefined values.

4.5 Enhanced Hooks Emulation Concept

The Enhanced Hooks Emulation Concept of the C500 microcontroller family is a new, innovative way to control the execution of C500 MCUs, and to gain extensive information on the internal operation of the controllers. Emulation of on-chip ROM-based programs is possible, too. Each C500 production chip has built-in logic for the support of the Enhanced Hooks Emulation Concept. Therefore, no costly bond-out chips are necessary for emulation, ensuring that emulation and production chips are identical.

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

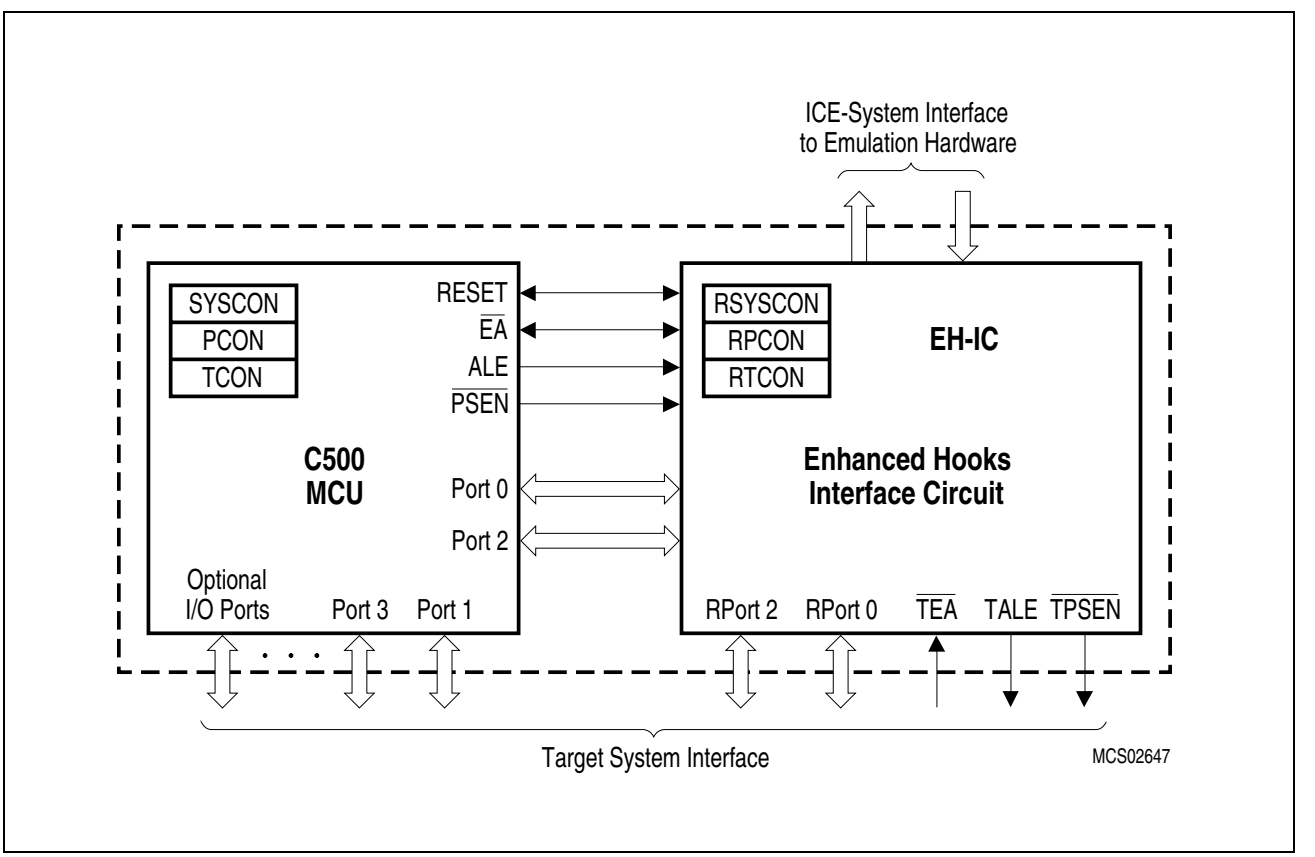


Figure 4-2
Basic C500 MCU Enhanced Hooks Concept Configuration

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

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

4.6 Eight Datapointers for Faster External Bus Access

4.6.1 The Importance of Additional Datapointers

The standard 8051 architecture provides just one 16-bit pointer for indirect addressing of external devices (memories, peripherals, latches, etc.). Except for a 16-bit “move immediate” to this datapointer and an increment instruction, any other pointer handling is bytewise. For complex applications with peripherals (e.g. LCD Controller or Real-Time Clock) located in the external data memory space, or extended data storage capacity, bytewise pointer handling turned out to be a “bottle neck” for the 8051’s communication to the external world. In particular, programming in high-level languages (PLM51, C51, 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.

4.6.2 How the eight Datapointers of the C505L are Implemented

Simply adding more datapointers is not suitable because of the need to keep up 100% compatibility with the 8051 instruction set. That instruction set allows the handling of only one single 16-bit datapointer (DPTR, consisting of the two 8-bit SFRs DPH and DPL).

To meet both of the above requirements (speed up external accesses, 100% compatibility with 8051 architecture), the C505L contains a set of eight 16-bit registers 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 desired 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 C505L that handle the datapointer therefore affect only pointer addressed by DPSEL at that very moment, rather than the other 7 pointers.

Figure 4-3 illustrates the addressing mechanism: A 3-bit field in register DPSEL points to the currently used DPTRx. Any standard 8051 instruction (e.g. MOVX @DPTR, A - transfer a byte from accumulator to an external location addressed by DPTR) now uses this activated DPTRx.

Special Function Register DPSEL (Address 92_H) **Reset Value: $XXXXX000_B$**

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

Bit	Function
DPSEL.2-0	Data pointer select bits DPSEL.2-0 defines the number of the actual active data pointer.DPTR0-7.

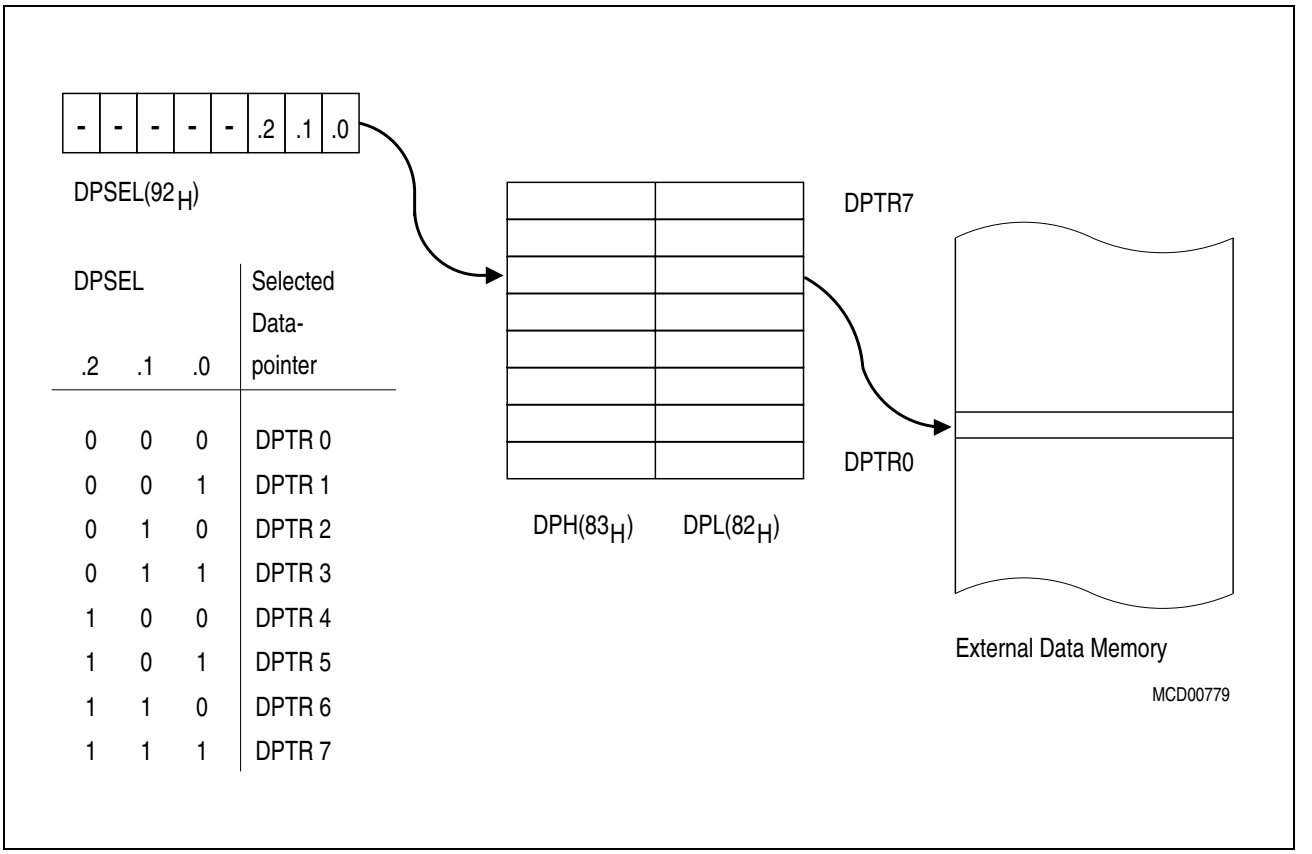


Figure 4-3
Accessing of External Data Memory via Multiple Datapointers

4.6.3 Advantages of Multiple Datapointers

Using the addressing mechanism described above for external data memory results in less code and faster execution of external accesses. Whenever the contents of the datapointer must be altered between two or more 16-bit addresses, one instruction that selects a new datapointer does this job. If the program uses just one datapointer, then it has to save the old value (with two 8-bit instructions) and load the new address, byte-by-byte. This takes more time and requires additional space in the internal RAM.

4.6.4 Application Example and Performance Analysis

The following example demonstrates the use of multiple data pointers in a table transfer from the code memory to external data memory.

Start address of ROM source table: $1FFF_H$
Start address of table in external RAM: $2FA0_H$

Example 1: Using Only One Datapointer (Code for a C501)

Initialization Routine

```
MOV     LOW(SRC_PTR), #0FF_H;     Initialize shadow_variables with source_pointer
MOV     HIGH(SRC_PTR), #1F_H
MOV     LOW(DES_PTR), #0A0_H;     Initialize shadow_variables with destination_pointer
MOV     HIGH(DES_PTR), #2F_H
```

Table Look-up Routine under Real Time Conditions

```
                                  ;                                  Number of cycles
PUSH    DPL                       ;Save old datapointer              2
PUSH    DPH                       ;                                  2
MOV     DPL, LOW(SRC_PTR)         ;Load Source Pointer               2
MOV     DPH, HIGH(SRC_PTR)        ;                                  2
;INC    DPTR                      Increment and check for end of table (execution time
;CJNE   …                         not relevant for this consideration)  –
MOVC    A,@DPTR                   ;Fetch source data byte from ROM table  2
MOV     LOW(SRC_PTR), DPL         ;Save source_pointer and           2
MOV     HIGH(SRC_PTR), DPH        ;load destination_pointer          2
MOV     DPL, LOW(DES_PTR)         ;                                  2
MOV     DPH, HIGH(DES_PTR)        ;                                  2
INC     DPTR                      ;Increment destination_pointer
                                  ;(ex. time not relevant)           –
MOVX    @DPTR, A                  ;Transfer byte to destination address  2
MOV     LOW(DES_PTR), DPL         ;Save destination_pointer          2
MOV     HIGH(DES_PTR),DPH         ;                                  2
POP     DPH                       ;Restore old datapointer           2
POP     DPL                       ;                                  2

;                                 Total execution time (machine cycles): 28
```

Example 2: Using Two Datapointers (Code for a C505L)

Initialization Routine

```
MOV     DPSEL, #06H                ;Initialize DPTR6 with source pointer
MOV     DPTR, #1FFFH
MOV     DPSEL, #07H                ;Initialize DPTR7 with destination pointer
MOV     DPTR, #2FA0H
```

Table Look-up Routine under Real-time Conditions

```
                                   ;                                    Number of cycles
PUSH    DPSEL                      ;Save old source pointer             2
MOV     DPSEL, #06H                ;Load source pointer                 2
;INC    DPTR                       Increment and check for end of table (execution time
;CJNE   …                          not relevant for this consideration) –
MOVC    A,@DPTR                    ;Fetch source data byte from ROM table 2
MOV     DPSEL, #07H                ;Save source_pointer and
                                   ;load destination_pointer            2
MOVX    @DPTR, A                   ;Transfer byte to destination address 2
POP     DPSEL                      ;Save destination pointer and
                                   ;restore old datapointer             2

;                                  Total execution time (machine cycles): 12
```

The example above shows that utilization of the C505L's multiple datapointers can make external bus accesses two times as fast as with a standard 8051 or 8051 derivative. Here, four data variables in the internal RAM and two additional stack bytes were spared, too. For some applications where all eight datapointers are employed, a C505L program has up to 24 byte (16 variables and 8 stack bytes) of the internal RAM free for other use.

5 System Reset

5.1 Hardware Reset Operation

The hardware reset function incorporated in the C505L allows for an easy automatic start-up with 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 commonly done when the power-down mode is to be terminated.

In addition to the hardware reset, which is applied externally to the C505L, there are two internal reset sources: The watchdog timer, and the oscillator watchdog. This chapter deals only 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 two machine cycles (12 oscillator periods) while the oscillator is running. With the oscillator running, the internal reset is executed during the second machine cycle, and is repeated every cycle until RESET goes low again.

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

At the reset pin, a pulldown resistor is internally connected to V_{SS} to allow a power-up reset with an external capacitor only. When V_{DD} is applied, an automatic power-up reset can be caused by connecting the reset pin to V_{DD} via a capacitor. After V_{DD} has been turned on, the capacitor must hold the voltage level at the reset pin for a specific time in order to effect a complete reset.

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

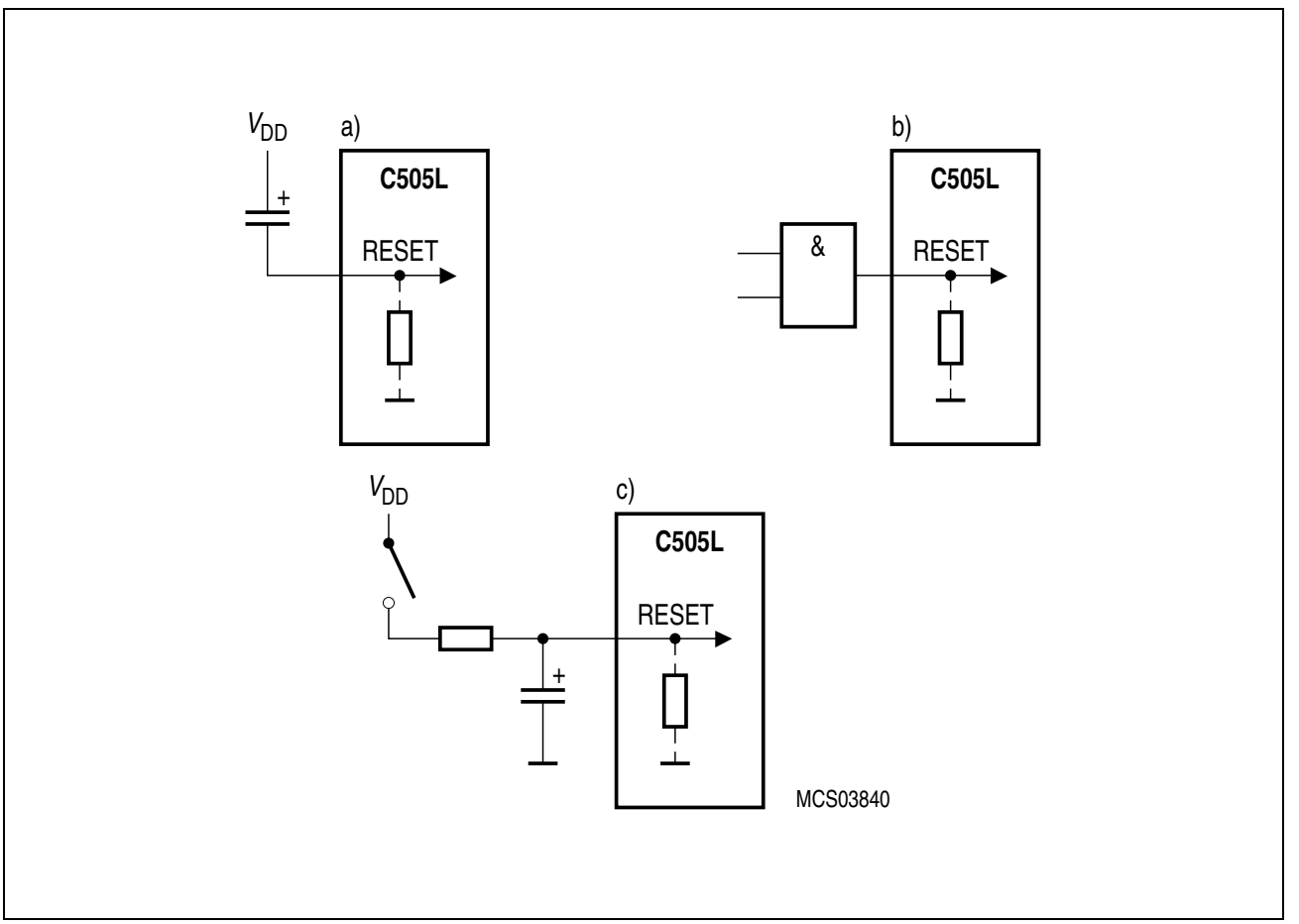


Figure 5-1
Reset Circuitries

A correct reset leaves the processor in a defined state. The program execution starts at location 0000_H. After reset is accomplished internally, the port latches of ports 0 to 4 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, 3 and 4) output a one (1). Port 2 lines output a one after reset. The internal SFRs are set to their initial states as defined in **Table 3-2**.

The contents of the internal RAM and XRAM of the C505L are not affected by a reset. After power-up the contents are undefined, and they remain unchanged during a reset if the power supply is not turned off.

5.2 Fast Internal Reset after Power-On

The C505L 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 do not enter their default reset states 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. If a crystal is used, the start up time of the oscillator is relatively long (typ. 10 ms). During this time period, the pins are in an undefined state which could have severe effects, especially to actuators connected to port pins.

In the C505L, 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). If the watchdog circuitry detects a failure condition for the on-chip oscillator because the latter has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator), 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 5-2**).

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

- Typ.: 18 µs
- Max.: 34 µs

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

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

Although the oscillator watchdog provides a fast internal reset, it is also necessary to apply an external reset signal when powering up. The reasons are:

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

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

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

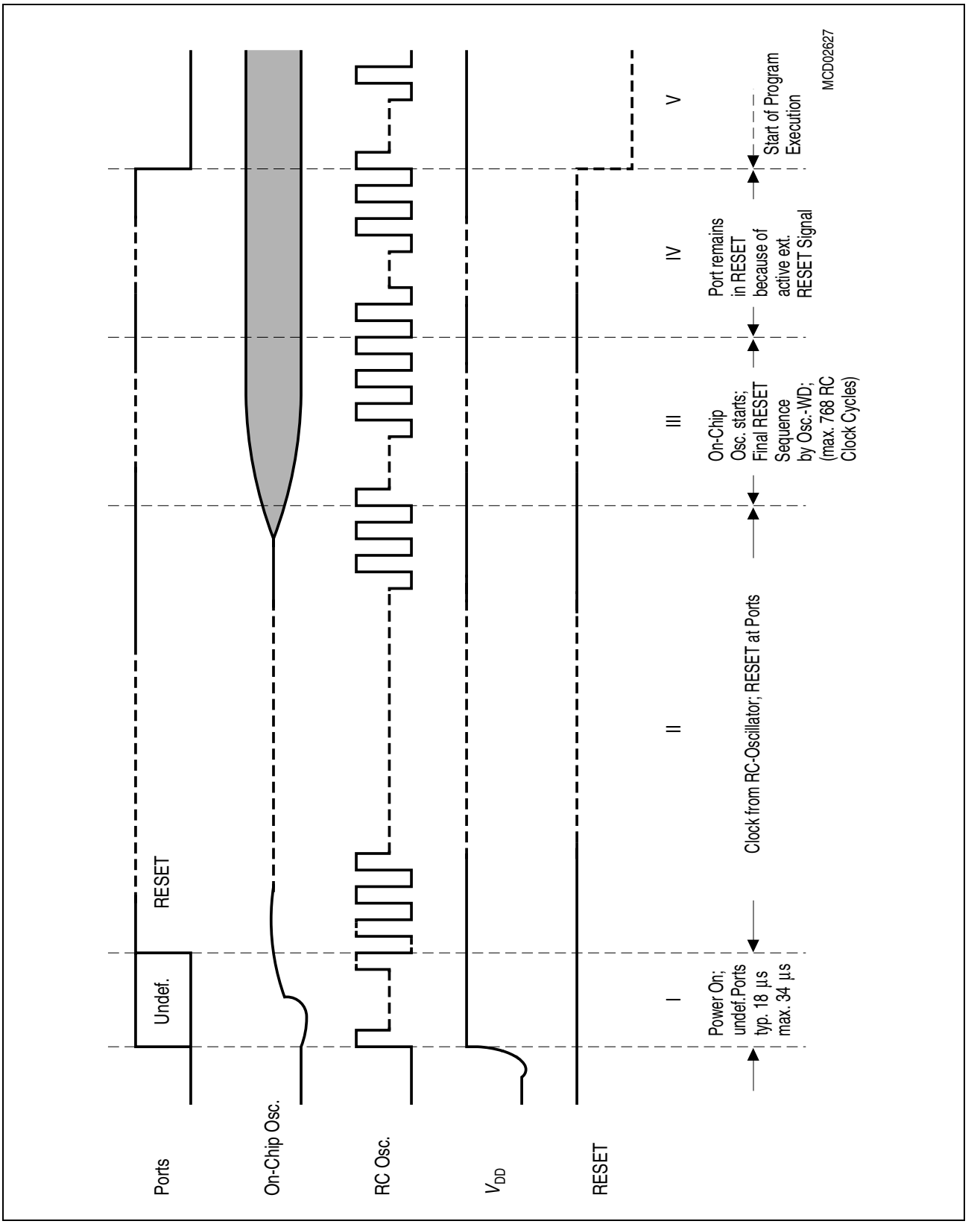


Figure 5-2
Power-On Reset of the C505L

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 with the internal CPU timing. When the reset is found active (high level), the internal reset procedure is started. It takes two complete machine cycles to put the complete device into its correct reset state, i.e. all Special Function Registers (SFRs) containing their default values, the port latches containing ‘1’ s etc.

Note that this reset procedure may also be performed by the oscillator watchdog if there is no clock available at the device. The oscillator watchdog provides an auxiliary clock at the XTAL1 and XTAL2 pins for performing a complete reset without another. The RESET signal must be active for at least one machine cycle; after this time the C505L remains in its reset state as long as the signal is active. When the signal goes inactive, this transition is recognized in the subsequent state 5 phase 2 of the machine cycle. Then the processor starts program execution in the subsequent state 5 phase 1. One phase later (state 5 phase 2), the first falling edge at pin ALE occurs.

5.4 Oscillator and Clock Circuit

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

Figure 5-3 shows the recommended oscillator circuit.

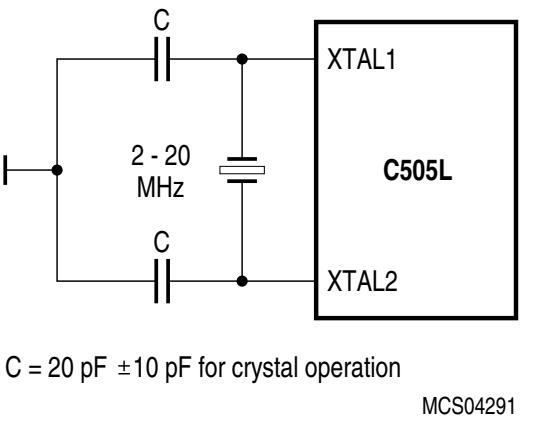


Figure 5-3
Recommended Oscillator Circuit

In this application, the on-chip oscillator is used as a crystal-controlled, positive-reactance oscillator. A more detailed schematic is given in **Figure 5-4**. The oscillator is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. The crystal specifications and capacitances are non-critical. In this circuit, 20 pF can be used as single capacitance at any frequency together with a good-quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. If a ceramic resonator is used, the two capacitors normally have different values depending on the oscillator frequency. We recommend consulting the manufacturer of the ceramic resonator for value specifications of these capacitors.

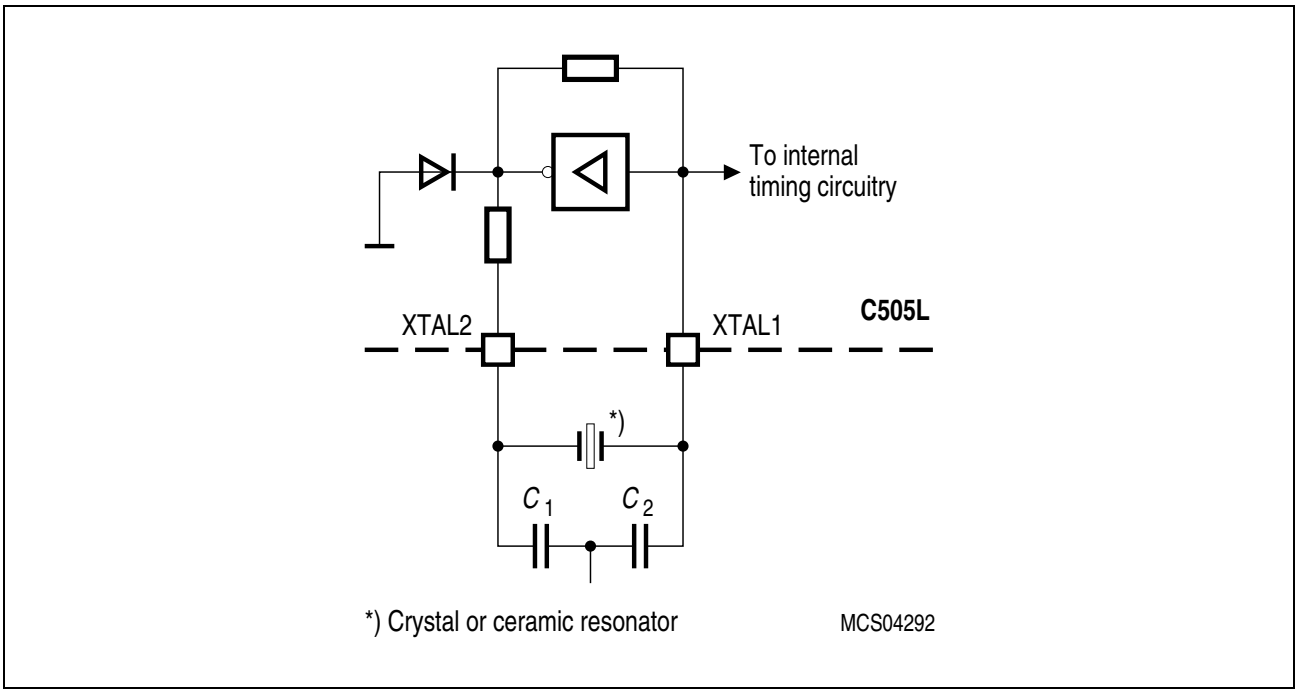


Figure 5-4
On-Chip Oscillator Circuitry

To drive the C505L with an external clock source, the external clock signal has to be applied to XTAL1, as shown in **Figure 5-5**. XTAL2 has to be left unconnected. A pullup resistor is recommended to increase the noise margin, but is optional if V_{OH} of the driving gate corresponds to the V_{IH1} specification of XTAL1 (refer to Data Sheet).

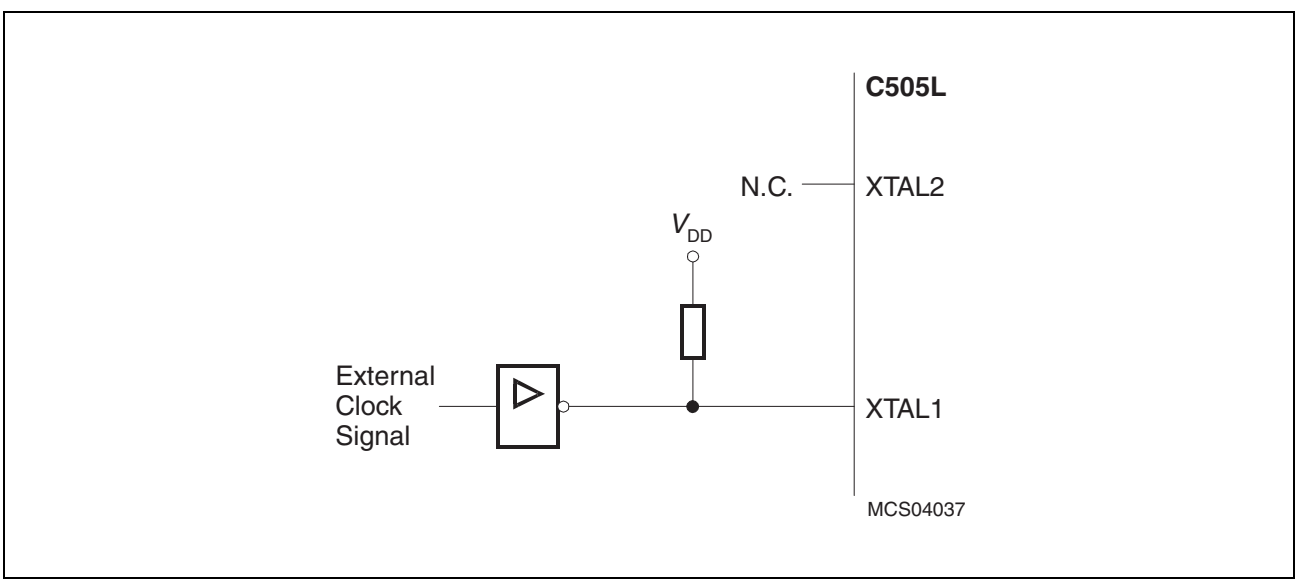


Figure 5-5
External Clock Source

5.5 System Clock Output

For peripheral devices requiring a system clock, the C505L provides a clock output signal derived from the oscillator frequency as an alternate output function on pin P1.6/CLKOUT. If bit CLK is set (bit 6 of SFR ADCON0), a clock signal with 1/6 of the oscillator frequency is gated to pin P1.6/CLKOUT. To use this function, the port pin must be programmed to a 1, which is also the default after reset.

Special Function Register ADCON0 (Address $D8_H$) **Reset Value: $00X00000_B$**

	MSB							LSB	
Bit No.	DF_H	DE_H	DD_H	DC_H	DB_H	DA_H	$D9_H$	$D8_H$	
$D8_H$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	ADCON0

The shaded bits are not used for clock output control.

Bit	Function
CLK	Clockout enable bit When set, pin P1.6/CLKOUT outputs the system clock which is 1/6 of the oscillator frequency.
–	Reserved bits for future use. Read by CPU returns undefined values.

The system clock is high during S3P1 and S3P2 of every machine cycle and low during all other states. Thus, the duty cycle of the clock signal is 1:6. Associated with a MOVX instruction, the system clock coincides with the last state (S3) in which a $\overline{RD}$ or $\overline{WR}$ signal is active. A timing diagram of the system clock output is shown in **Figure 5-6**.

Note: During slow-down operation, the frequency of the CLKOUT signal is divided by 32.

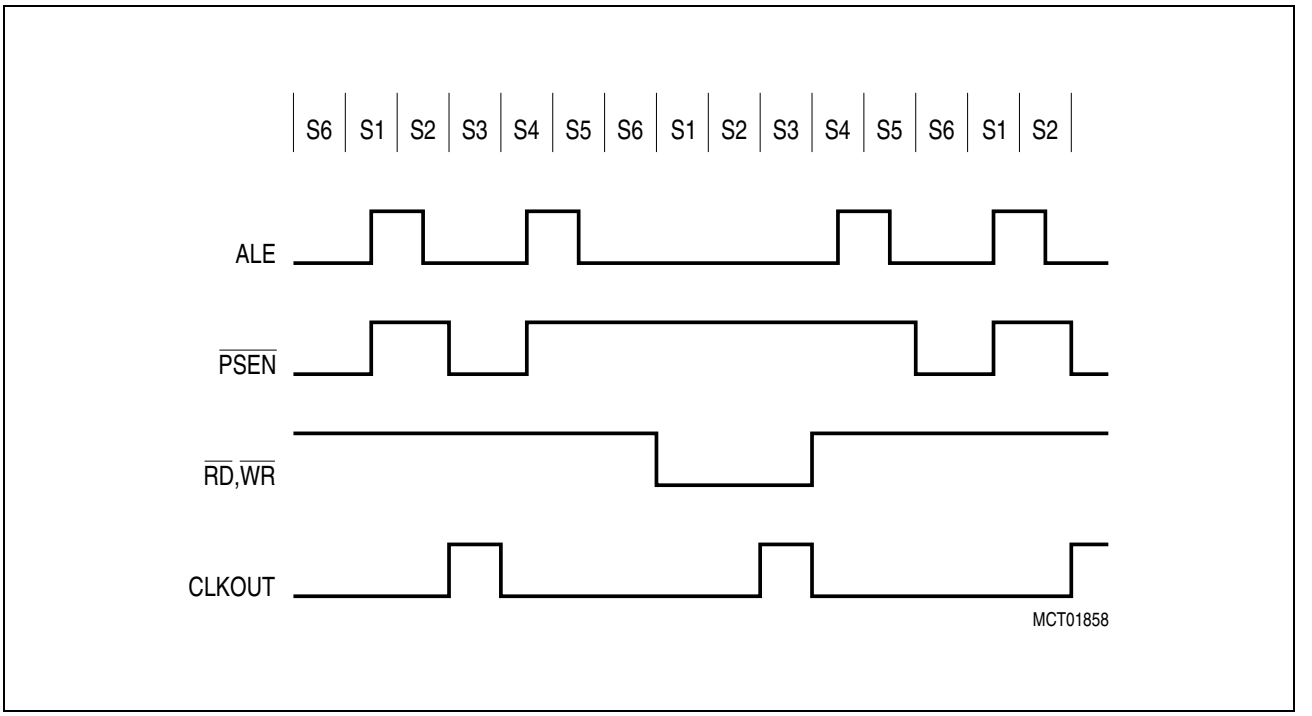


Figure 5-6
Timing Diagram - System Clock Output

6 On-Chip Peripheral Components

This chapter gives detailed information about all on-chip peripherals of the C505L except for the integrated interrupt controller, which is described separately in **Chapter 7**.

6.1 Parallel I/O

The C505L has five 8-bit and one 6-bit (port 5) digital I/O ports. Port 0 is an open-drain bidirectional I/O port, while ports 1 through 5 are quasi-bidirectional I/O ports with internal pull-up resistors. When configured as inputs, ports 1-5 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 Special Function register (SFR) contents. In this function, port 0 is not an open-drain port, but uses a strong internal pull-up FET.

The C505L has 36 output lines (4 rows and 32 columns) for LCD voltage output. Of these, 20 are dedicated output lines: R0-R3 and C0-C15. Fourteen LCD output lines are used as alternate functions of the bits of port 4 (C16-C23) and port 5 (C24-C29). Two LCD output lines C30 and C31 are used as alternate functions of P3.5 / T1 and P3.4 / T0 respectively.

6.1.1 Port Structures

The C505L generally allows digital I/O on 46 lines, grouped into five 8-bit digital and one 6-bit digital I/O 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-P5 are performed via their corresponding SFRs. Depending on the specific ports, multiple functions are assigned to the port pins. Therefore, the parallel I/O ports of the C505L can be grouped into six different types which are listed in **Table 6-1**.

Table 6-1
C505L Port Structure Types

Type	Description
A	**Standard digital I/O ports which can also be used for external address/data bus**
B	Standard multifunctional **digital I/O port lines**
C	Mixed digital/analog **I/O port lines with programmable analog input function**
D	LCD Output Lines
E	Standard digital I/O or LCD output lines
F	Standard multifunctional digital I/O or LCD output lines

Type A and B port pins are standard C501-compatible I/O port lines, which can be used for digital I/O. The type A ports (port 0 and port 2) are also designed for accessing external data or program memory. Type B port lines are located at port 3 (except P3.4 and P3.5), and are used for digital I/O or for other alternate functions as described in the pin description. Type D port lines provide the LCD controller outputs R0-R3 and C0-C15 as primary functions. Type E port lines are located at port 4 and port 5 and provide the LCD controller output lines as alternate functions. Type F port lines

are at P3.4 / T0 and P3.5 / T1 and have a digital alternate input each, apart from LCD output functions.

The C505L provides eight analog input lines that are implemented as mixed digital/analog inputs (type C). The 8 analog inputs, AN0-AN7, are located at the port 1 pins P1.0 to P1.7. After reset, all analog inputs are disabled and the related pins of port 1 are configured as digital inputs. The analog function of the specific port 1 pins are enabled by bits in the SFRs P1ANA. Writing a 0 to a bit position of P1ANA assigns the corresponding pin to operate as analog input.

Note: P1ANA is a mapped SFR and can only be accessed if bit RMAP in SFR SYSCON is set.

As already mentioned, ports 1, 3, 4 and 5 are provided for multiple alternate functions. These functions are listed in **Table 6-2**.

Table 6-2
Alternate Functions of Port 1, 3, 4 and 5

Port	Second / third Function	Port Type	Function
P1.0	AN0 / $\overline{\text{INT3}}$ / CC0	C	Analog input channel 0 / External Interrupt 3 / Compare/Capture channel 0 input/output
P1.1	AN1 / INT4 /CC1	C	Analog input channel 1 / External Interrupt 4 input / Compare/Capture channel 1 input/output
P1.2	AN2 / INT5 /CC2	C	Analog input channel 2 / External Interrupt 5 input / Compare/Capture channel 2 input/output
P1.3	AN3 / INT6 / CC3	C	Analog input channel 3 / External Interrupt 6 input / Compare/Capture channel 3 input/output
P1.4	AN4	C	Analog input channel 4
P1.5	AN5 / T2EX	C	Analog input channel 5 / Timer 2 external reload / trigger input
P1.6	AN6 / CLKOUT	C	Analog input channel 6 / System clock output
P1.7	AN7 / T2	C	Analog input channel 7 / Timer 2 external count input
P3.0	RxD	B	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TxD	B	Serial port's transmitter data output (asynchronous) or data clock output (synchronous)
P3.2	$\overline{\text{INT0}}$	B	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	B	External interrupt 1 input, timer 1 gate control
P3.4	T0 / C31	F	Timer 0 external counter input / LCD column 31 output
P3.5	T1 / C30	F	Timer 1 external counter input / LCD column 30 output
P3.6	$\overline{\text{WR}}$	B	External data memory write strobe
P3.7	$\overline{\text{RD}}$	B	External data memory read strobe
P4.0	C16	E	LCD column 16 output
P4.1	C17	E	LCD column 17 output
P4.2	C18	E	LCD column 18 output
P4.3	C19	E	LCD column 19 output
P4.4	C20	E	LCD column 20 output
P4.5	C21	E	LCD column 21 output
P4.6	C22	E	LCD column 22 output
P4.7	C23	E	LCD column 23 output
P5.0	C24	E	LCD column 24 output
P5.1	C25	E	LCD column 25 output
P5.2	C26	E	LCD column 26 output
P5.3	C27	E	LCD column 27 output
P5.4	C28	E	LCD column 28 output
P5.5	C29	E	LCD column 29 output

6.1.2 Standard I/O Port Circuitry

Figure 6-1 is a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the five I/O-ports. The bit latch (one bit in the port's SFR) is represented as a type-D flip-flop that 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 P5) activate the "read-latch" signal, while others activate the "read-pin" signal.

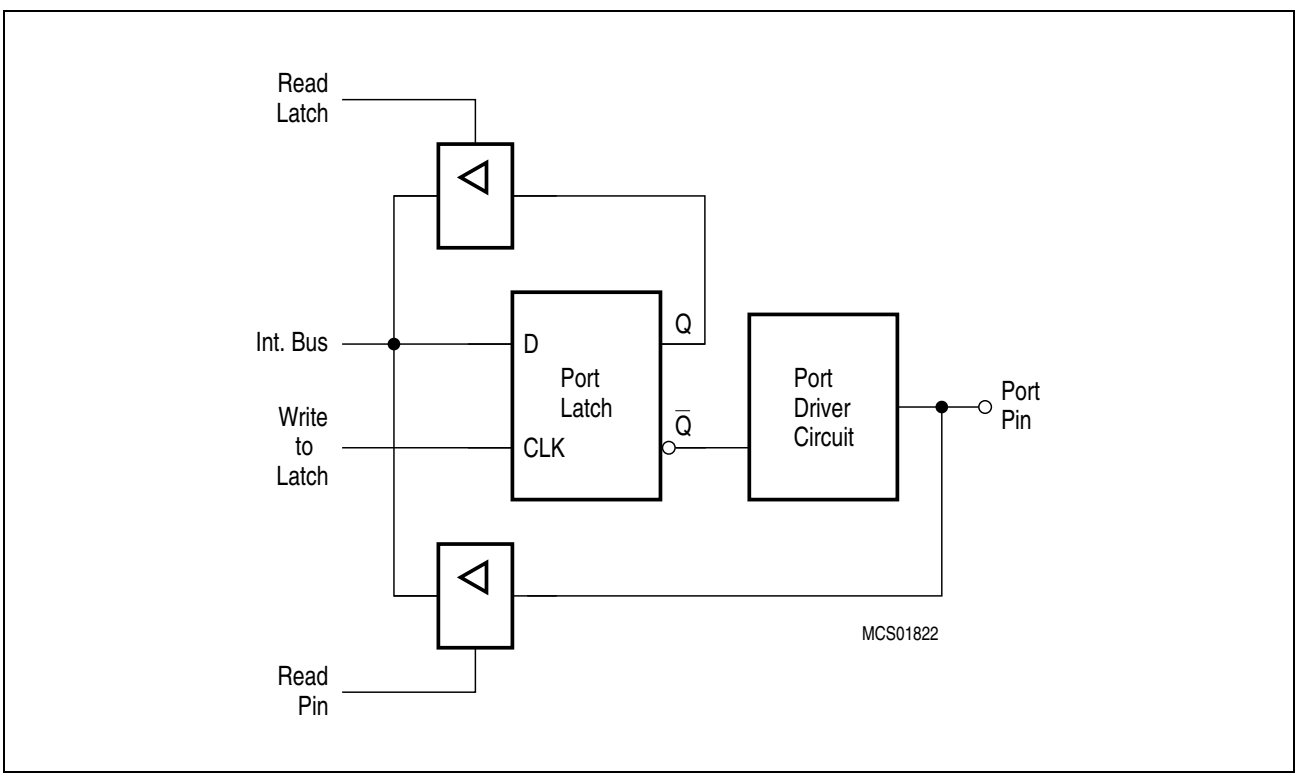


Figure 6-1
Basic Structure of a Port Circuit

The output drivers of Port 1 to 5 have internal pull-up FETs (see **Figure 6-2**). Each I/O line can be used independently as an input or output. To be used as an input, the port bit stored in the bit latch must contain 1 (that means for **Figure 6-2**: **Q** = 0), which turns off the output driver FET n1. Then, for ports 1 to 5, the pin is pulled high by the internal pull-ups, but can be pulled low by an external source. When externally pulled low, the port pins source current (I_{IL} or I_{TL}). For this reason, these ports are called “quasi-bidirectional”.

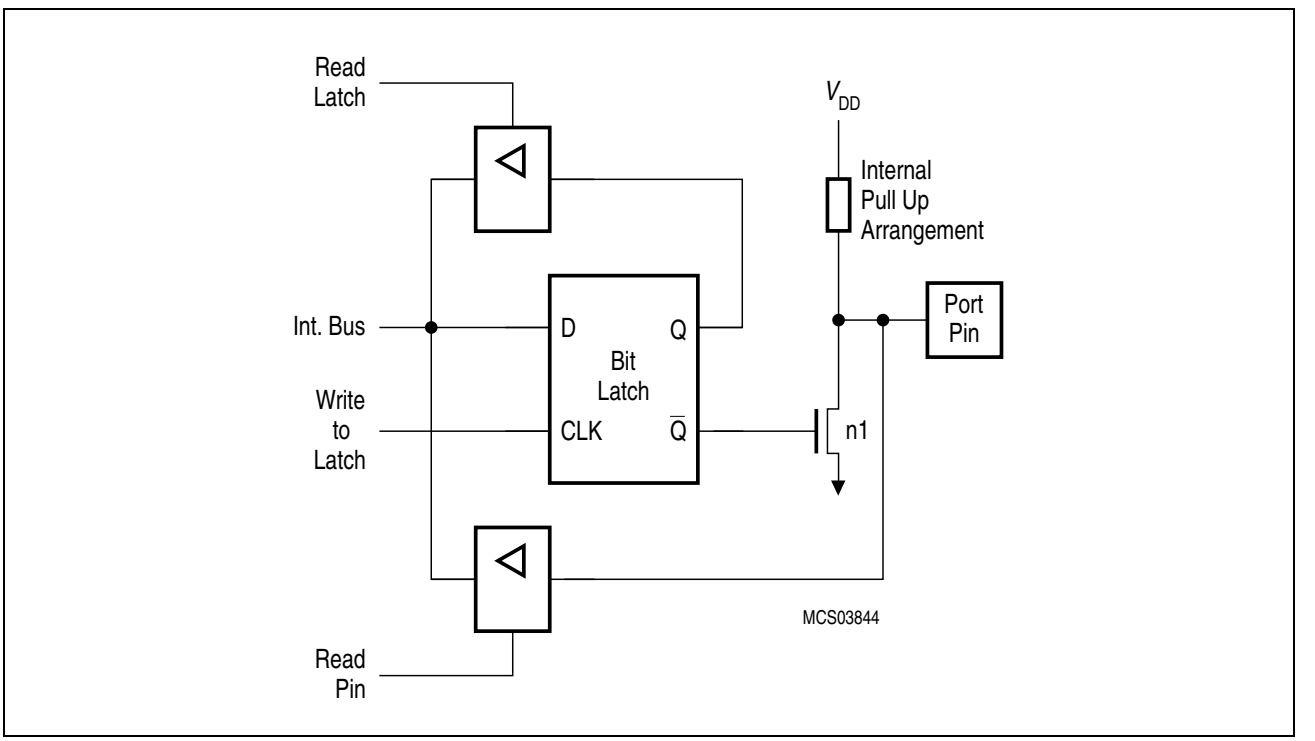


Figure 6-2
Basic Output Driver Circuit of Ports 1 to 5

6.1.2.1 Port 0 Circuitry

Port 0, in contrast to ports 1 to 5, is considered to be a “true” bidirectional, because the port 0 pins float when configured as inputs. Thus, this port differs in not having internal pull-ups. The pull-up FET in the P0 output driver (see **Figure 6-3**) is used only when the port is emitting 1s during external memory accesses. Otherwise, the pull-up is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a 1 to the port latch leaves both output FETs off and the pin floats. In that condition, the pin 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 pull-ups are required.

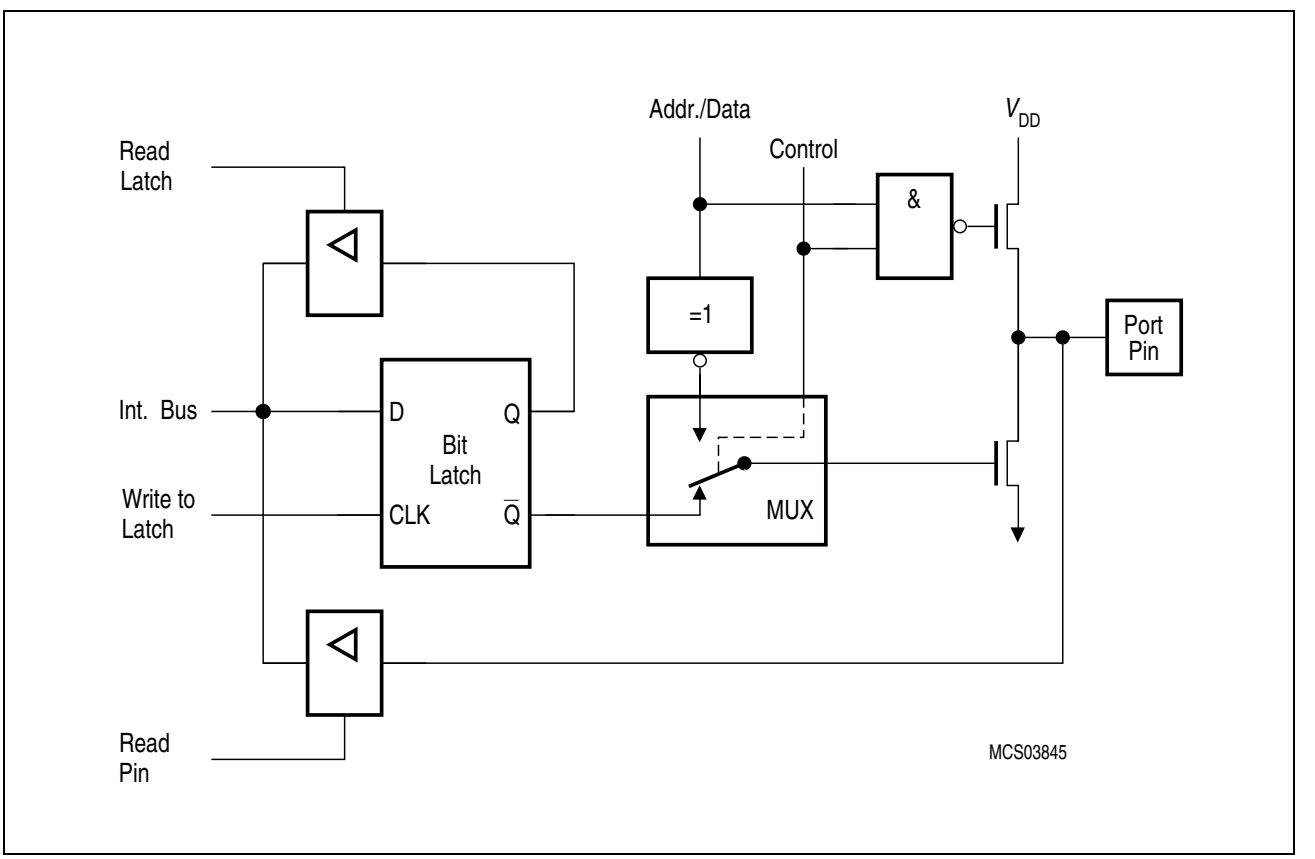


Figure 6-3
Port 0 Circuit

6.1.2.2 Port 1 and Port 3 Circuitry

The pins of ports 1 and 3 are multifunctional. They are port pins and also serve to implement the special features as listed in **Table 6-2**.

Figure 6-4 is 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 1 or the pulldown FET will be on and the port pin will be stuck at 0. After reset, all port latches contain 1 s.

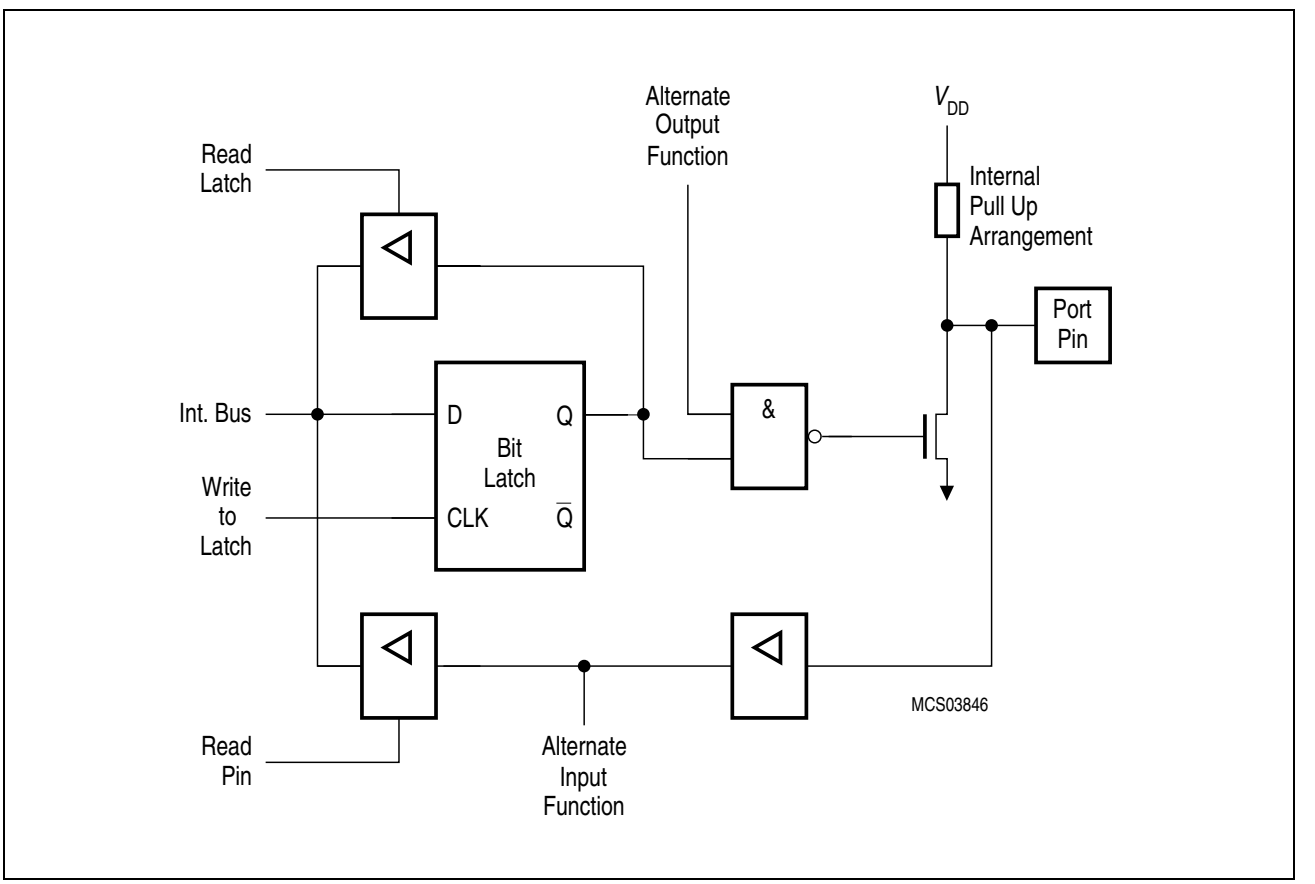


Figure 6-4
Ports 1 and 3

The LCD output functions of Port 3.4/T0 and P3.5/T1 pins are of type F.

6.1.2.3 Port 2 Circuitry

As shown in **Figure 6-3** and **Figure 6-5**, 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 these ports cannot be used as general purpose I/O, even if not all address lines are used externally. The switching is done by an internal control signal dependent on the input level at the $\overline{EA}$ pin and/or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P0/P2 SFR remains unchanged. Since it is an address/data bus, port 0 uses a pull-up FET as shown in **Figure 6-3**. When a 16-bit address is used, port 2 uses the additional strong pull-up p1 (**Figure 6-5a**) to emit 1s for the entire external memory cycle instead of the weak 1s (p2 and p3) used during normal port activity.

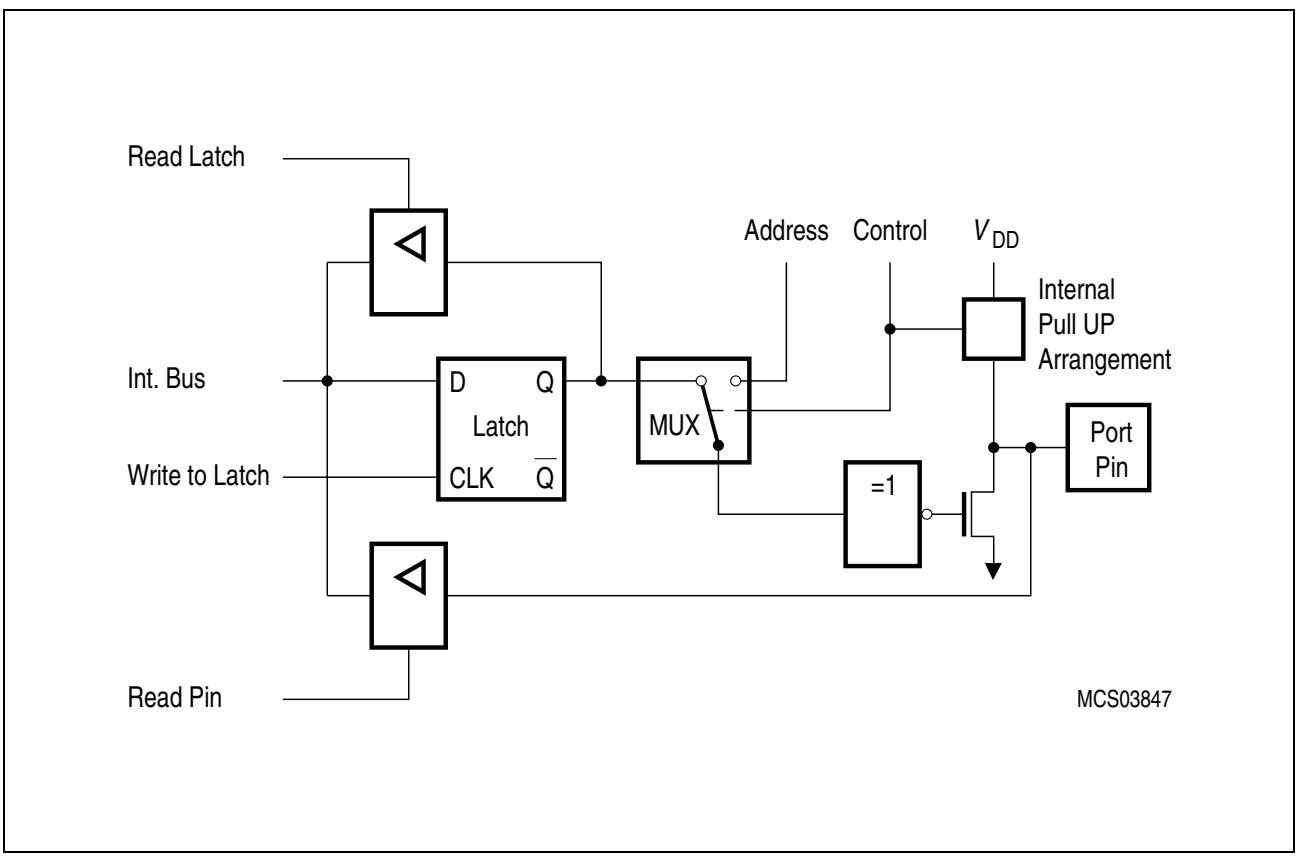


Figure 6-5
Port 2 Circuit

If no external bus cycles are generated using data or code memory accesses, port 0 can be used for I/O functions.

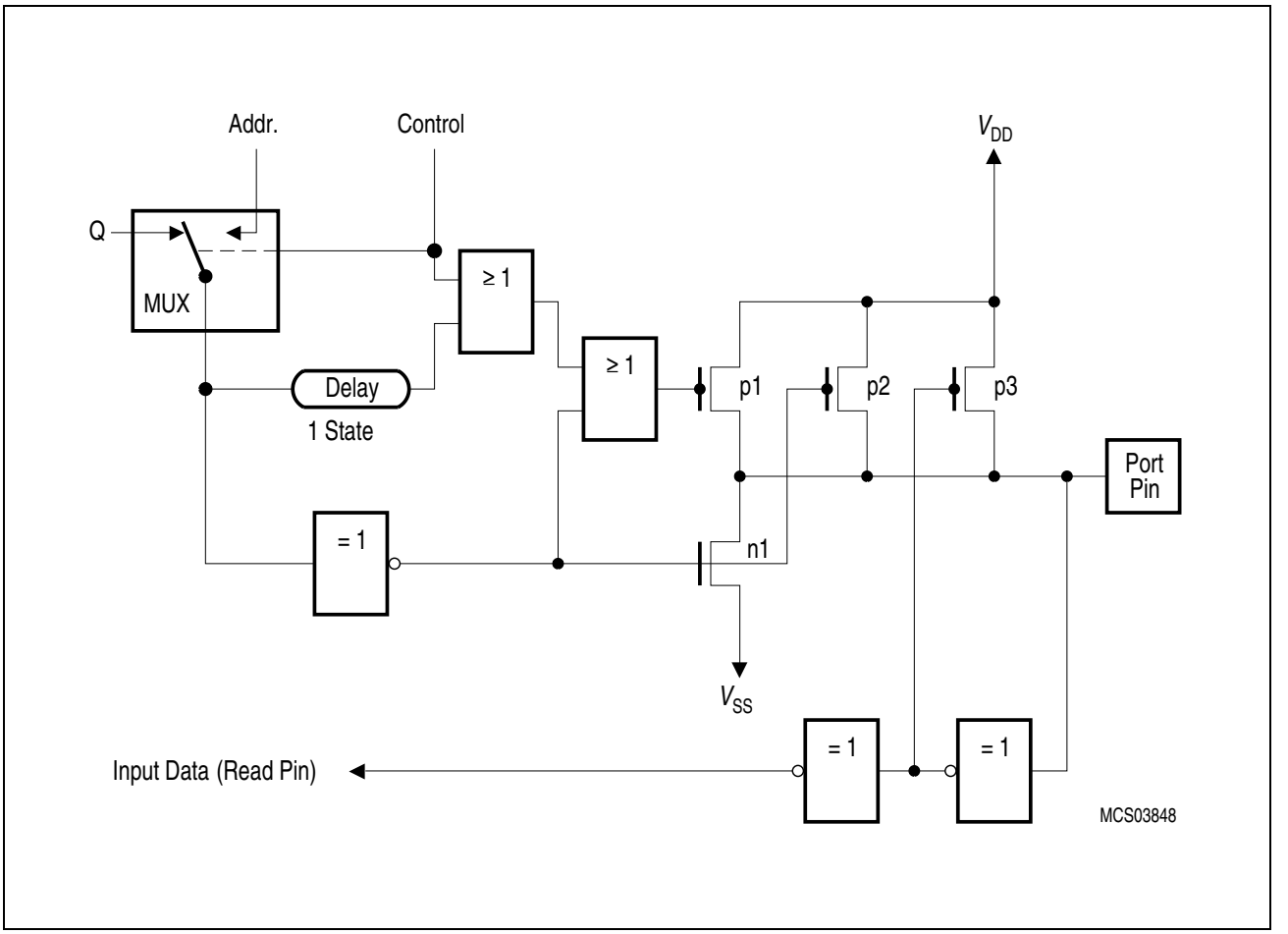


Figure 6-5a
Port 2 Pull-up Arrangement

Port 2 in I/O function works similar to the Type B port driver circuitry (see **Section 6.1.3.1**) whereas in address output function, it works similar to Port 0 circuitry.

6.1.3 Detailed Output Driver Circuitry

The pull-ups mentioned before and included in **Figure 6-2, 6-4** and **6-5** are pull-up arrangements. The differences of the port types available in the C505L are described in the following sections.

6.1.3.1 Type B Port Driver Circuitry

Figure 6-6 shows the output driver circuit of the type B multifunctional digital I/O port lines. The basic circuitry of these ports is shown in **Figure 6-4**. The pull-up arrangement of type B port lines has one n-channel pulldown FET and three pull-up FETs:

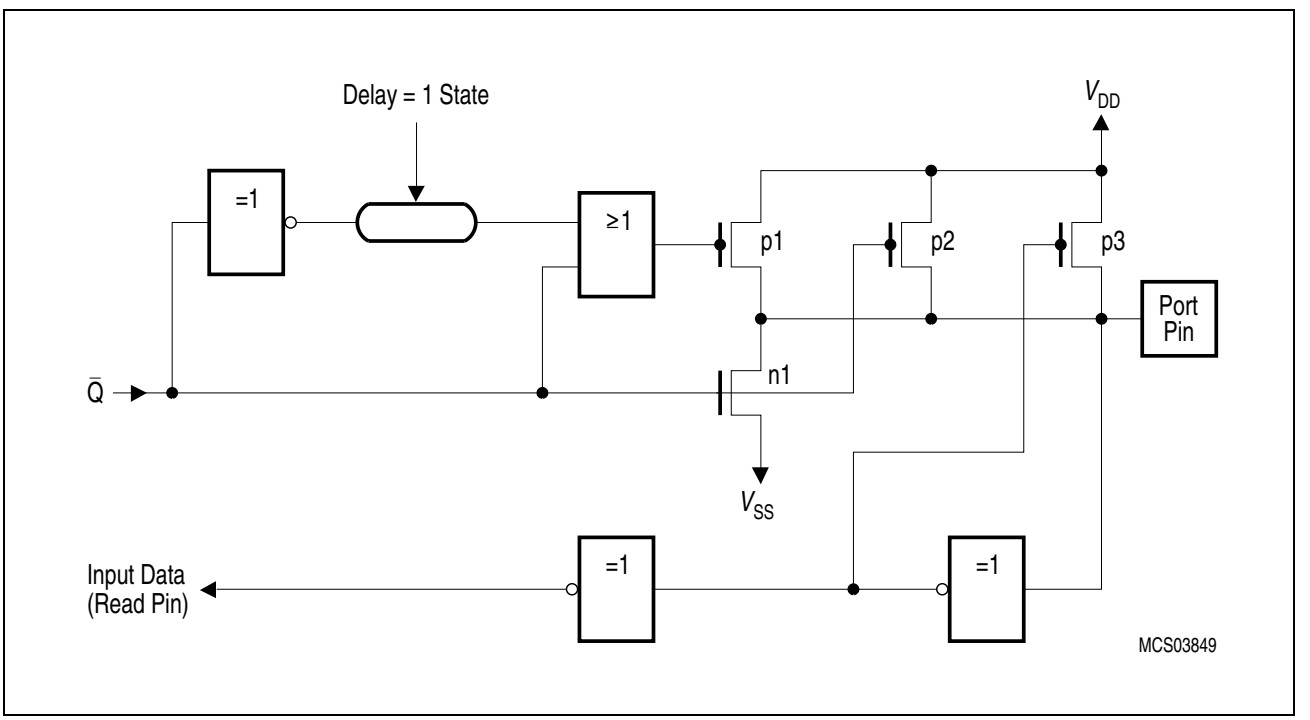


Figure 6-6
Driver Circuit of Type B Port Pins

- The **pull-down FET n1** is of n-channel type. It is a very strong driver transistor that is capable of sinking high currents (I_{OL}); it is activated only if a 0 is programmed to the port pin. A short circuit to V_{DD} must be avoided if the transistor is turned on, since the high current might destroy the FET. This also means that no 0 must not be programmed into the latch of a pin that is used as input.
- The **pull-up FET p1** is of p-channel type. It is activated for 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 pull-up can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.
- The **pull-up FET p2** is of p-channel type. It is always activated when a 1 is in the port latch, thus providing the logic high output level. This pull-up FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g., when used as input with logic low input level.
- The **pull-up 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 pull-up current if a logic high level is 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 pull-up FET p2 is active and sources the current I_{IL}. If, in addition, the pull-up FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, 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.

Activating and deactivating of the four different transistors translates into four states. 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 weak 1 until the first 0-to-1 transition on the latch occurs. Until then, the output level might stay below the trip point of the external circuitry.

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

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

6.1.3.2 Type C Port Driver Circuitry

Figure 6-7 shows the C505L's port driver circuit of the type C: Mixed digital/analog I/O port 1 lines. The analog function is selected by the bits in the SFR P1ANA. When the analog function is selected, all output driver transistors (p1, p2, p3 and n1) are switched off.

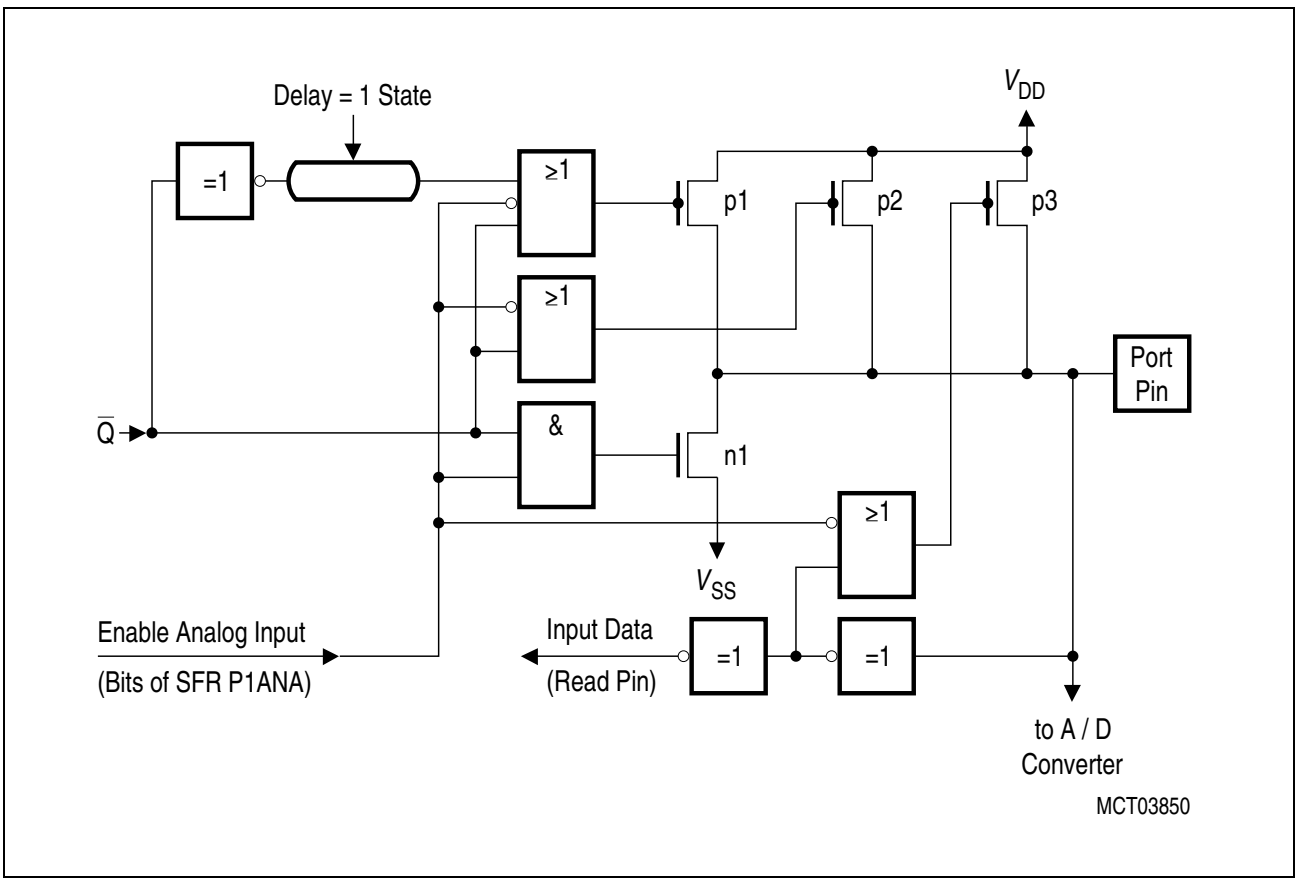


Figure 6-7
Driver Circuit of Type C Port Pins

6.1.3.3 Type D Port Driver Circuitry

Figure 6-8 shows the C505L's port driver circuit of the type D. These pins are used for the dedicated LCD Outputs, R0-R3 and C0-C15. The p-channel transistor p2 is a weak pull-up transistor similar to the p2 transistor in other digital I/O ports. After a reset operation, the LCD Controller remains disabled. At this point, the weak pull-up is enabled. When, the LCD controller is enabled by bit LCEN in SFR SYSCON, this transistor is switched off, making the LCD output available at the port pin. LCD levels are in the range mentioned in the DC Specifications (refer to Data Sheet).

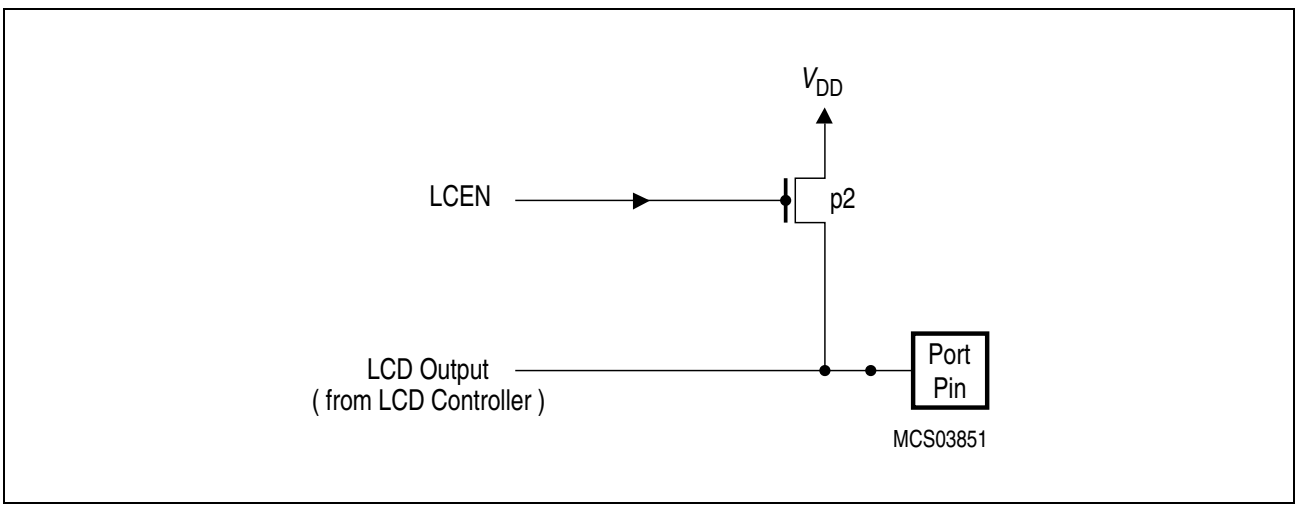


Figure 6-8
Driver Circuit of Type D Port Pins

6.1.3.4 Type E and F Port Driver Circuitry

Figure 6-9 describes the output structure of Type E port 4 and 5 pins. Such pins have both digital I/O and LCD output functions. Ports 4 and 5 have no digital alternate function possible. When the LCD output is enabled, all digital output drivers are switched off; the respective output signals from the LCD controller are selected; and LCD voltage levels are available at the pins shown in **Table 6-2**. After reset, the LCD output functions remain disabled due the LCEN bit having been cleared in the register LCON. When the LCD Controller is enabled with bit LCEN, the bits DSB0 and DSB1 in the register LCON will enable/disable the LCD output lines C16-C23 and C24-C31, respectively.

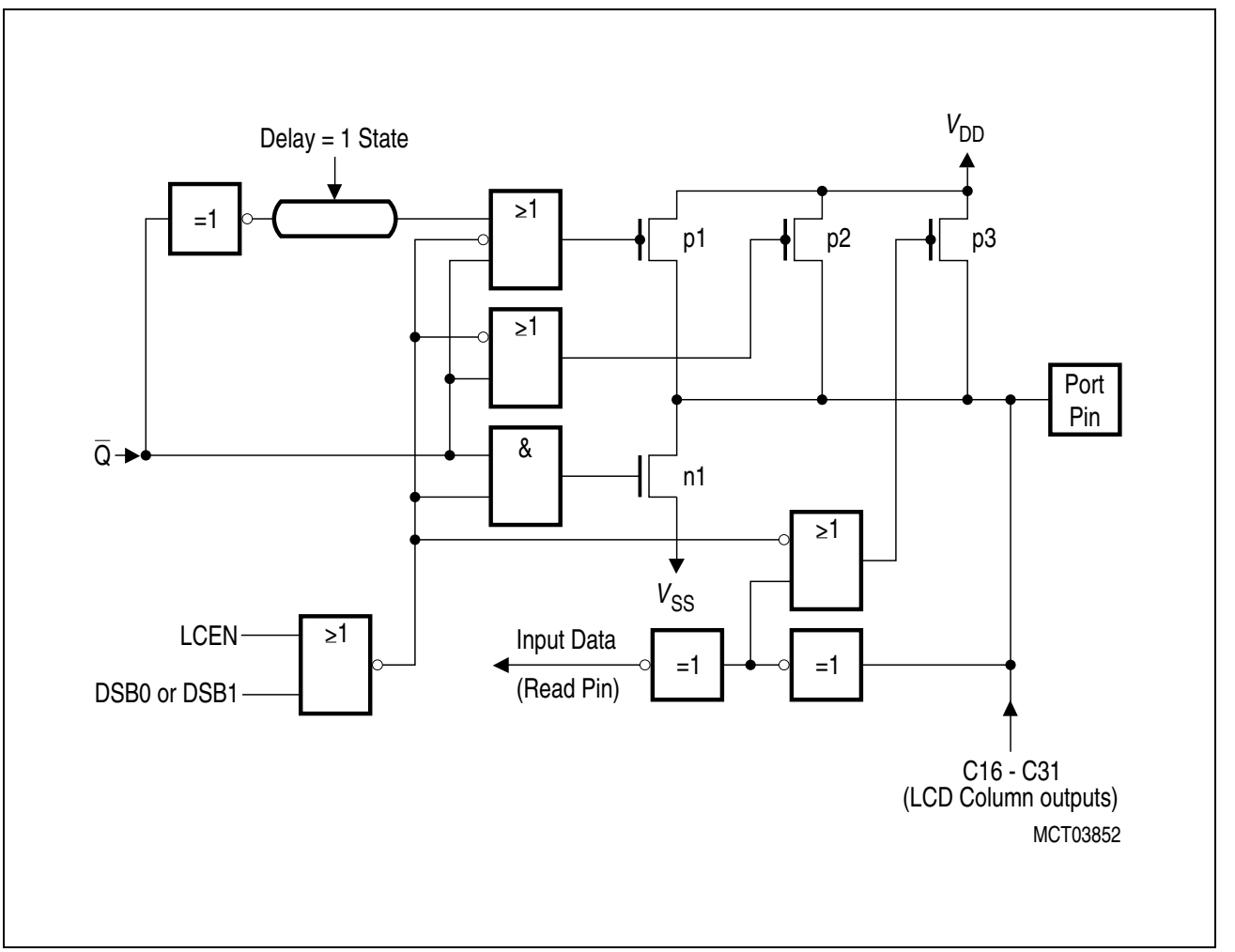


Figure 6-9
Driver Circuit of Type E and Type F Port Pins

The digital I/O function of the Type F pins is similar to that of Type E digital I/O pins, with one difference. The digital alternate function is available for P3.4/T0/C31 and P3.5/T1/C30 only (see **Section 6.1.2.2**).

6.1.4 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 6-10** 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, certain requirements must be met with regards to the pulse length of signals, in order to ensure that signal edges are 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.

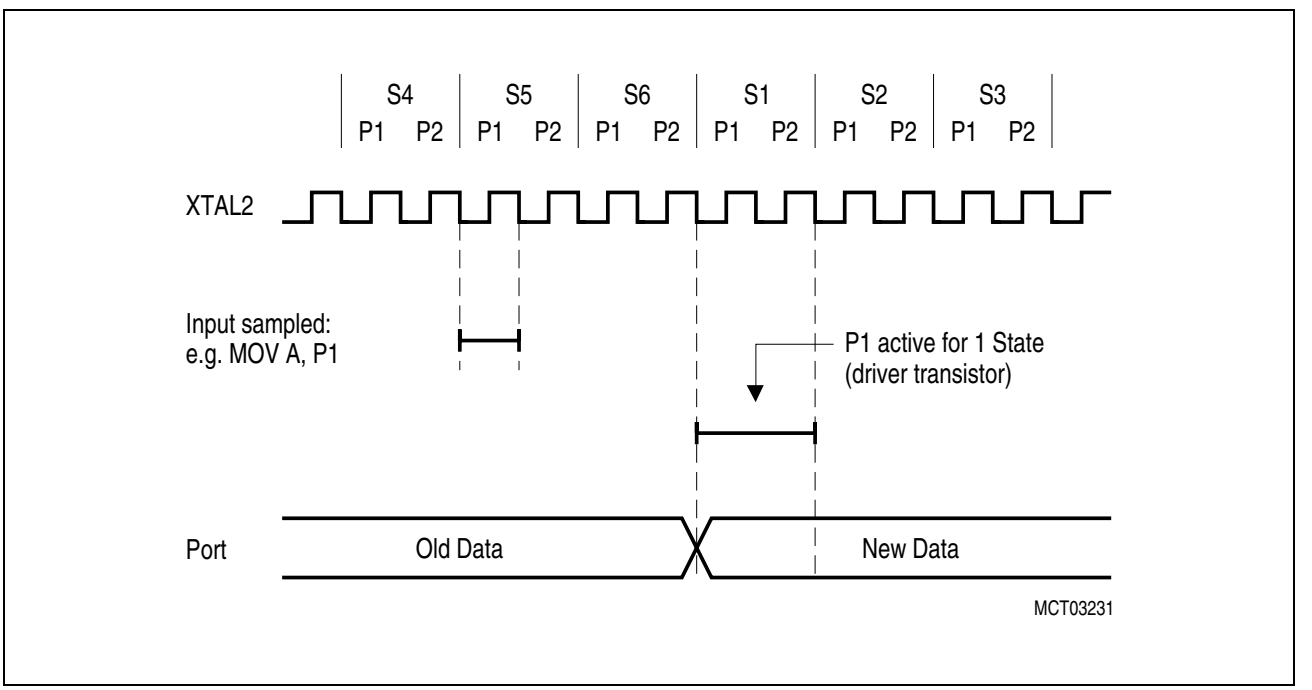


Figure 6-10
Port Timing

6.1.5 Port Loading and Interfacing

The output buffers of ports 1 to 5 can drive TTL inputs directly. The maximum port load that guarantees correct logic output levels is specified in the DC characteristics in the Data Sheet of the C505L. The corresponding parameters are V_{OL} and V_{OH}.

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

When used as inputs, ports 1 to 5 are not floating but have internal pull-up transistors. The driving devices must be capable of sinking a sufficient current if a logic low level shall is applied to the port pin. Parameters I_{TL} and I_{IL} in the DC characteristics of the Data Sheet specify these currents. Port 1 may be programmed to analog input function, but has floating inputs in these cases.

6.1.6 Read-Modify-Write Feature of Ports 0 to 5

Some port-reading instructions read the latch and others read the pin. Instructions that read the latch rather than the pin do read a value, possibly change it, and then rewrite it to the latch. These “read-modify-write” instructions are listed in **Table 6-3**. If the destination is a port or a port pin, these instructions read the latch rather than the pin.

Note that all other instructions that can be used to read a port just read the port pin. Reading from either the latch or the pin is performed by reading the SFR P0, P2 and P3; for example, “MOV A, P3” reads the value from port 3 pins, while “ANL P3, #0AAH” reads from the latch, modifies the value and writes it back to the latch.

It is not obvious that the last three instructions in **Table 6-3** are read-modify-write instructions, but they are because they read the port byte - all 8 bits - and modify the addressed bit, then write the complete byte back to the latch.

Table 6-3
“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, LABEL
MOV Px.y,C	Move carry bit to bit y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

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

6.2 Timers/Counters

The C505L contains three 16-bit timers/counters (timer 0, 1, and 2) which are useful in many applications for timing and counting.

In “timer” function, the timer register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 6 oscillator periods, the counter rate is 1/6 of the oscillator frequency.

In “counter” function, the timer register is incremented in response to a 1-to-0 transition (falling edge) at the corresponding external input pin (T0, T1, or T2, which provide alternate functions of P3.4, P3.5 and P1.7, respectively). In the counter 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 (12 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/12 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but it must be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

6.2.1 Timer/Counter 0 and 1

Timer/counter 0 and 1 of the C505L are fully compatible with timer/counter 0 and 1 of the C501 and can be used in the same four operating modes:

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

Mode 1: 16-bit timer/counter

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

Mode 3: Timer/counter 0 is configured as one 8-bit timer/counter and one 8-bit timer; 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) that 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 (SFRs), TCON and TMOD.

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

6.2.1.1 Timer/Counter 0 and 1 Registers

Six SFRs control the timer/counter 0 and 1 operation:

- TL0/TH0 and TL1/TH1 - counter registers, low and high part
- TCON and TMOD - control and mode select registers

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

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

Bit	Function
TLx.7-0 x = 0-1	Timer/counter 0/1 low register

Operating Mode	Description
0	“TLx” holds the 5-bit prescaler value.
1	“TLx” holds the lower 8-bit part of the 16-bit timer/counter value.
2	“TLx” holds the 8-bit timer/counter value.
3	TL0 holds the 8-bit timer/counter value; TL1 is not used.

Bit	Function
THx.7-0 x = 0-1	Timer/counter 0/1 high register

Operating Mode	Description
0	“THx” holds the 8-bit timer/counter value.
1	“THx” holds the higher 8-bit part of the 16-bit timer/counter value
2	“THx” holds the 8-bit reload value.
3	TH0 holds the 8-bit timer value; TH1 is not used.

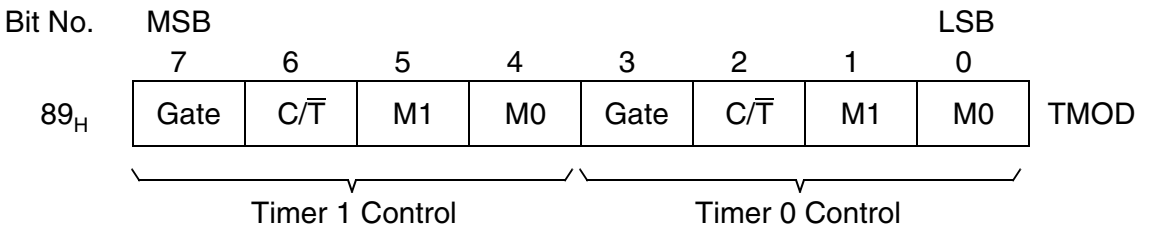
Special Function Register TCON (Address 88_H) **Reset Value: 00_H**

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

The shaded bits are not used for controlling timer/counter 0 and 1.

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

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



Bit	Function
GATE	Gating control When set, timer/counter “x” is enabled only while “INT x” pin is high and “TRx” control bit is set. When cleared, timer “x” is enabled whenever “TRx” control bit is set.
C/**T**	Counter or timer select bit Set for counter operation (input from “Tx” input pin). Cleared for timer operation (input from internal system clock).
M1 M0	Mode select bits

M1	M0	Function
0	0	8-bit timer/counter: “THx” operates as 8-bit timer/counter “TLx” serves as 5-bit prescaler
0	1	16-bit timer/counter. “THx” and “TLx” are cascaded; there is no prescaler
1	0	8-bit auto-reload timer/counter. “THx” holds a value which is to be reloaded into “TLx” each time it overflows
1	1	Timer 0: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits. Timer 1: Timer/counter 1 stops

6.2.1.2 Mode 0

Putting either timer/counter 0 or timer/counter 1 into mode 0 configures it as an 8-bit timer/counter with a divide-by-32 prescaler. **Figure 6-11** 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 1s to all 0 s, it sets the timer overflow flag TF0. The overflow flag TF0 then can be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or $\overline{INT0}$ = 1. Setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$, in order to facilitate pulse width measurements. TR0 is a control bit in the SFR TCON; Gate is in TMOD.

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

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR0, TF0, TH0, TL0 and $\overline{INT0}$ for the corresponding timer 1 signals in **Figure 6-11**. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

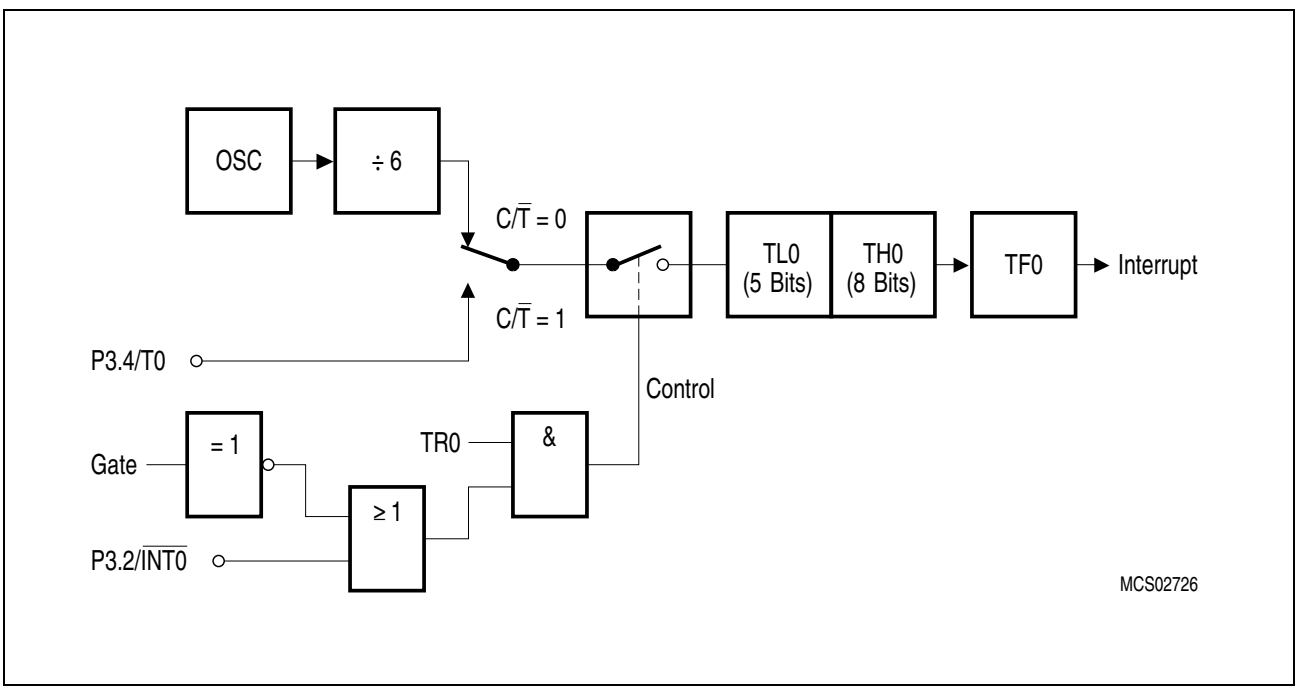


Figure 6-11
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

6.2.1.3 Mode 1

Mode 1 is the same as mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in **Figure 6-12**.

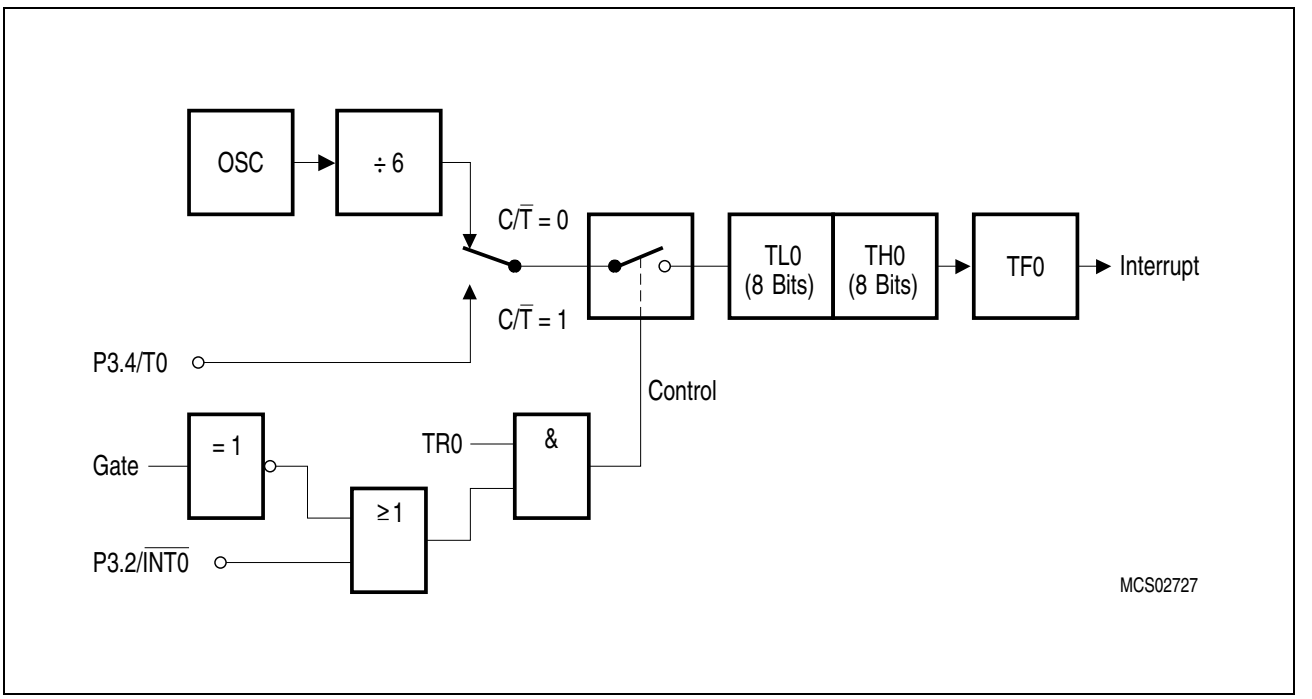


Figure 6-12
Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

6.2.1.4 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in **Figure 6-13**. 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.

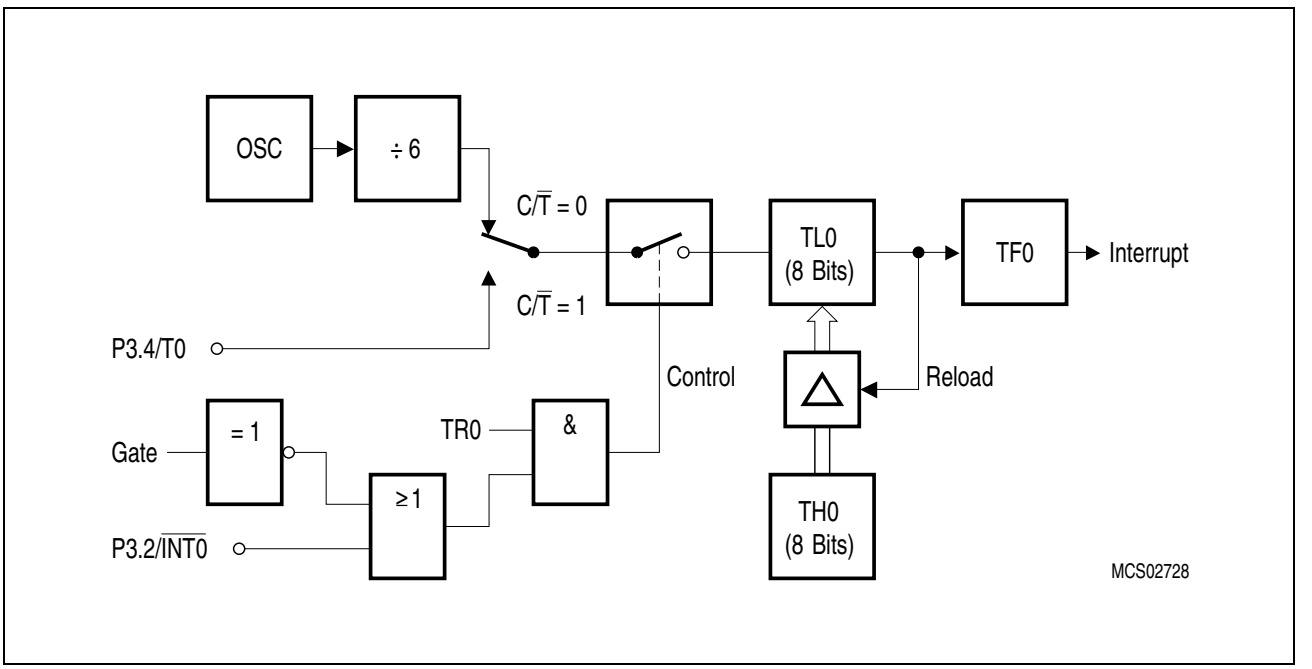


Figure 6-13
Timer/Counter 0, Mode 2: 8-Bit Timer/Counter with Auto-Reload

6.2.1.5 Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on timer 0 is shown in **Figure 6-14**. 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 any mode as long as an interrupt from timer 1 itself is not required.

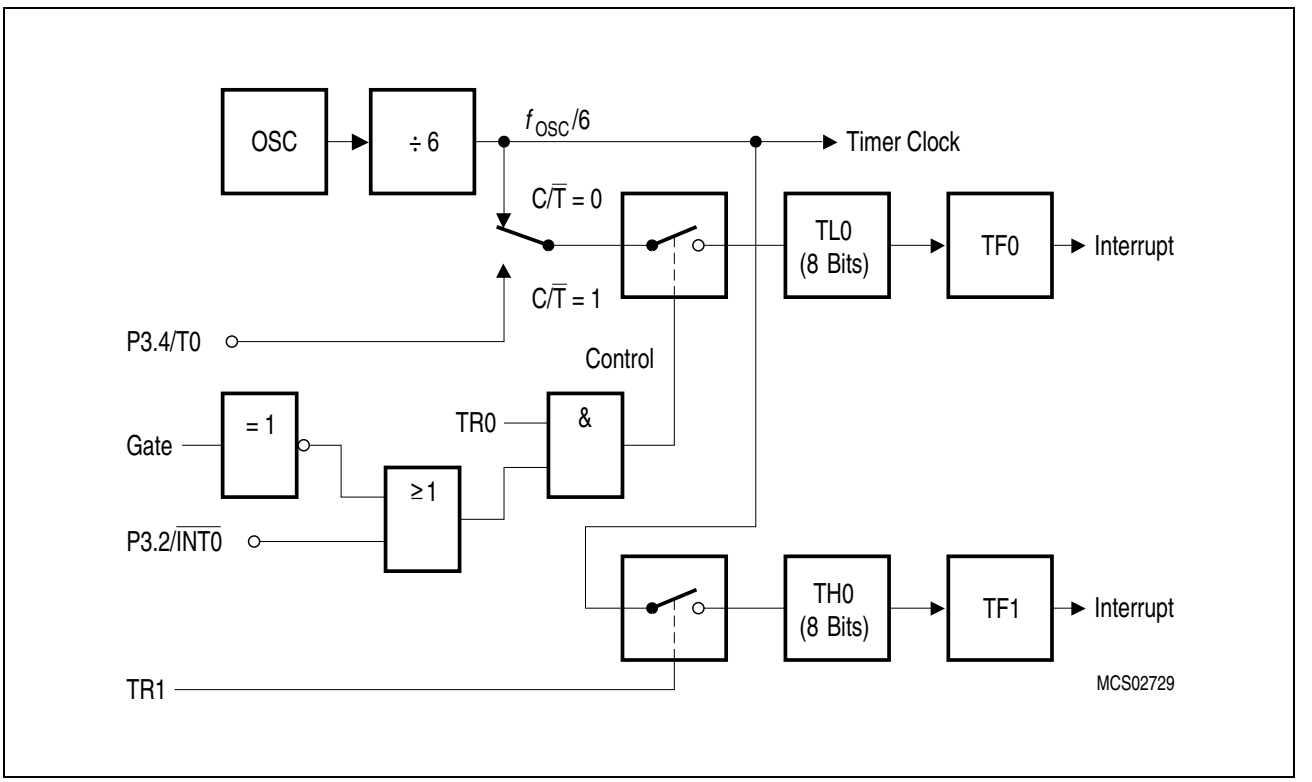


Figure 6-14
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

6.2.2 Timer/Counter 2 with Additional Compare/Capture/Reload

Timer 2 has additional compare/capture/reload features that make it one of the most powerful peripheral units of the C505L. It can be used for all kinds of digital signal generation and event capturing, such as pulse generation, pulse width modulation (PWM), pulse width measuring, etc.

Timer 2 is designed to support various automotive control applications as well as industrial applications (e.g., frequency generation, digital-to-analog conversion, process control). Please note that the functionality of this timer is not equivalent to timer 2 of the C501.

The C505L's timer 2 allows the following operating modes in combination with the compare/capture/reload registers:

- Compare: Up to 4 PWM output signals with 65535 steps at maximum, and 300-ns resolution
- Capture: Up to 4 high speed capture inputs with 300-ns resolution
- Reload: Modulation of timer 2 cycle time

The block diagram in **Figure 6-15** shows the general configuration of timer 2 with the additional compare/capture/reload registers. The I/O pins that can be used for timer 2 control are located as multifunctional port functions at port 1 (see **Table 6-4**).

Table 6-4
Alternate Port Functions of Timer 2

Pin Symbol	Function
P1.0/AN0/$\overline{\text{INT3}}$/CC0	Compare output/capture input for CRC register
P1.1/AN1/INT4/CC1	Compare output/capture input for CC register 1
P1.2/AN2/INT5/CC2	Compare output/capture input for CC register 2
P1.3/AN3/INT6/CC3	Compare output/capture input for CC register 3
P1.5/AN5/T2EX	External reload trigger input
P1.7/AN7/T2	External count or gate input to timer 2

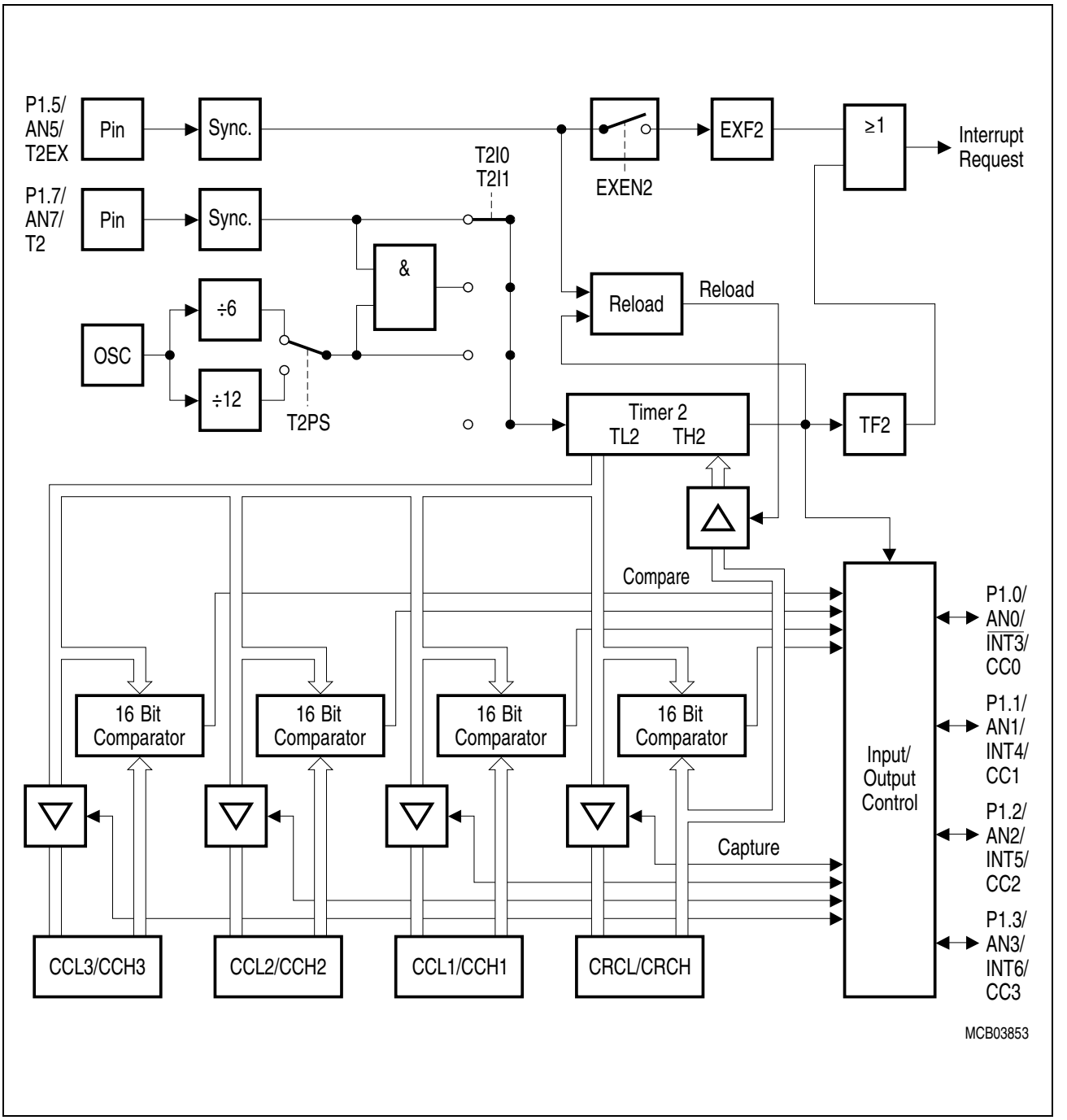


Figure 6-15
Timer 2 Block Diagram

6.2.2.1 Timer 2 Registers

This chapter describes all timer 2 SFRs. The interrupt-related SFRs are also included in this section. **Table 6-5** summarizes all timer 2 SFRs.

Table 6-5
Special Function Registers of the Timer 2 Unit

Name	Description	Address
T2CON	Timer 2 control register	$C8_H$
TL2	Timer 2, low byte	CC_H
TH2	Timer 2, high byte	CD_H
CCEN	Compare/capture enable register	$C1_H$
CRCL	Compare/reload/capture register, low byte	CA_H
CRCH	Compare/reload/capture register, high byte	CB_H
CCL1	Compare/capture register 1, low byte	$C2_H$
CCH1	Compare/capture register 1, high byte	$C3_H$
CCL2	Compare/capture register 2, low byte	$C4_H$
CCH2	Compare/capture register 2, high byte	$C5_H$
CCL3	Compare/capture register 3, low byte	$C6_H$
CCH3	Compare/capture register 3, high byte	$C7_H$
IEN0	Interrupt enable register 0	$A8_H$
IEN1	Interrupt enable register 1	$B8_H$
IRCON	Interrupt control register	$C0_H$

The T2CON timer 2 control register is a bit-addressable register which controls the timer 2 function and the compare mode of registers CRC, CC1 to CC3.

Special Function Register T2CON (Address $C8_H$) **Reset Value: $00X00000_B$**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	T2PS	I3FR	–	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

The shaded bits are not used for controlling timer/counter 2.

Bit	Function
T2PS	Prescaler select bit When set, timer 2 is clocked in the “timer” or “gated timer” function with 1/12 of the oscillator frequency. When cleared, timer 2 is clocked with 1/6 of the oscillator frequency. T2PS must be 0 for the counter operation of timer 2.
I3FR	External interrupt 3 falling/rising edge flag Used for capture function in combination with register CRC. If set, a capture to register CRC (if enabled) will occur on a positive transition at pin P1.0/AN0/$\overline{INT3}$/CC0
T2R1 T2R0	Timer 2 reload mode selection

T2R1	T2R0	Function
0	X	Reload disabled
1	0	Mode 0: auto-reload upon timer 2 overflow (TF2)
1	1	Mode 1: reload on falling edge at pin P1.5 / AN5 / T2EX

T2CM	Compare mode bit for registers CRC, CC1 through CC3 When set, compare mode 1 is selected. T2CM = 0 selects compare mode 0.
T2I1 T2I0	Timer 2 input selection

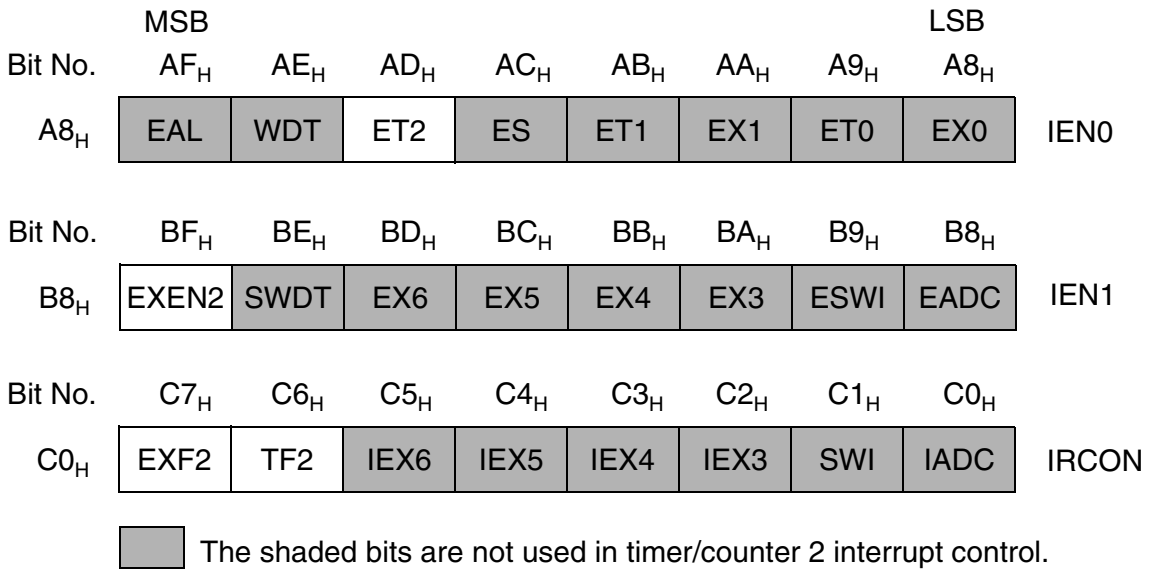
T2I1	T2I0	Function
0	0	No input selected, timer 2 stops
0	1	Timer function: input frequency = f_{osc}/6 (T2PS = 0) or f_{osc}/12 (T2PS = 1)
1	0	Counter function: external input signal at pin P1.7 / AN7 / T2
1	1	Gated timer function: input controlled by pin P1.7 / AN7 / T2

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

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

Bit	Function
TL2.7-0	Timer 2 value low byte The TL2 register holds the 8-bit low part of the 16-bit timer 2 count value.
TH2.7-0	Timer 2 value high byte The TH2 register holds the 8-bit high part of the 16-bit timer 2 count value.
CRCL.7-0	Reload register low byte CRCL is the 8-bit low byte of the 16-bit reload register of timer 2. It is also used for compare/capture functions.
CRCH.7-0	Reload register high byte CRCH is the 8-bit high byte of the 16-bit reload register of timer 2. It is also used for compare/capture functions.

Special Function Register IEN0 (Address $A8_H$) **Reset Value: 00_H**
Special Function Register IEN1 (Address $B8_H$) **Reset Value: 00_H**
Special Function Register IRCON (Address $C0_H$) **Reset Value: 00_H**



Bit	Function
ET2	Timer 2 overflow/external reload interrupt enable. If ET2 = 0, the timer 2 interrupt is disabled. If ET2 = 1, the timer 2 interrupt is enabled.
EXEN2	Timer 2 external reload interrupt enable. If EXEN2 = 0, the timer 2 external reload interrupt is disabled. If EXEN2 = 1, the timer 2 external reload interrupt is enabled. The external reload function is not affected by EXEN2.
EXF2	Timer 2 external reload flag. EXF2 is set when a reload is caused by a falling edge on pin T2EX while EXEN2 = 1. If ET2 in IEN0 is set (timer 2 interrupt enabled), EXF2 = 1 will cause an interrupt. EXF2 can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.
TF2	Timer 2 overflow flag. Set by a timer 2 overflow and must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.

Special Function Register CCEN (Address $C1_H$) **Reset Value: 00_H**

Bit No.	MSB							LSB
	7	6	5	4	3	2	1	0
$C1_H$	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0

CCEN

Bit	Function
COCAH3 COCAL3	Compare/capture mode for CC register 3

COCAH3	COCAL3	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.3 / AN3 / INT6 / CC3
1	0	Compare enabled
1	1	Capture on write operation into register CCL3

Bit	Function
COCAH2 COCAL2	Compare/capture mode for CC register 2

COCAH2	COCAL2	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.2 / AN2 / INT5 / CC2
1	0	Compare enabled
1	1	Capture on write operation into register CCL2

Bit	Function
COCAH1 COCAL1	Compare/capture mode for CC register 1

COCAH1	COCAL1	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin P1.1 / AN1 / INT4 / CC1
1	0	Compare enabled
1	1	Capture on write operation into register CCL1

Bit	Function
COCAH0 COCAL0	Compare/capture mode for CRC register

COCAH0	COCAL0	Function
0	0	Compare/capture disabled
0	1	Capture on falling/rising edge at pin P1.0 / AN0 / $\overline{\text{INT3}}$ / CC0
1	0	Compare enabled
1	1	Capture on write operation into register CRCL

6.2.2.2 Timer 2 Operation

Timer 2, which is a 16-bit-wide register, can operate as timer, event counter, or gated timer. Its operation is described in detail below.

Timer Mode

In timer function, the count rate is derived from the oscillator frequency. A prescaler offers the possibility of selecting a count rate of 1/6 or 1/12 of the oscillator frequency. Thus, the 16-bit timer register (consisting of TH2 and TL2) is incremented either in every machine cycle, or in every second machine cycle. The prescaler is selected by bit T2PS in SFR T2CON. If T2PS is cleared, the input frequency is 1/6 of the oscillator frequency. If T2PS is set, the 2:1 prescaler gates 1/12 of the oscillator frequency to the timer.

Gated Timer Mode

In gated timer function, the external input pin T2 (P1.7) functions as a gate to the input of timer 2. If T2 is high, the internal clock input is gated to the timer. T2 = 0 stops the counting procedure. This facilitates pulse width measurements. The external gate signal is sampled once every machine cycle.

Event Counter Mode

In the counter function, timer 2 is incremented in response to a 1-to-0 transition at its corresponding external input pin T2 (P1.7). In this function, the external input is sampled every machine cycle. When the sampled inputs show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the timer register in the cycle following the one in which the transition was detected. Since it takes two machine cycles (12 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/12 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but it must be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

Note: The prescaler must be off for proper counter operation of timer 2, i.e. T2PS must be 0.

In either of the cases, where timer 2 is configured as timer, event counter, or gated timer, rolling-over of the count from all 1s to all 0s sets the timer overflow flag TF2 in SFR IRCON, which can generate an interrupt.

If TF2 is used to generate a timer overflow interrupt, the request flag must be cleared by the interrupt service routine because it may be necessary to check whether the TF2 flag or the external reload request flag EXF2 requested the interrupt. Both request flags cause the program to branch to the same vector address.

Reloading of Timer 2

The reload mode for timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON. **Figure 6-16** shows the configuration of timer 2 in reload mode.

Mode 0: When timer 2 rolls over from all I's to all 0's, it not only sets TF2 but also causes the timer 2 registers to be loaded with the 16-bit value in the CRC registers, which are preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000_H.

Mode 1: A 16-bit reload from the CRC register is caused by a negative transition at the corresponding input pin P1.5/AN5/T2EX. In addition, this transition will set flag EXF2, if bit EXEN2 in SFR IEN1 is set. If the timer 2 interrupt is enabled, setting EXF2 will generate an interrupt. The external input pin T2EX is sampled in every machine cycle. When the sampling shows a high in one cycle and a low in the next cycle, a transition will be recognized. Timer 2 registers will then be reloaded in the cycle following the one in which the transition was detected.

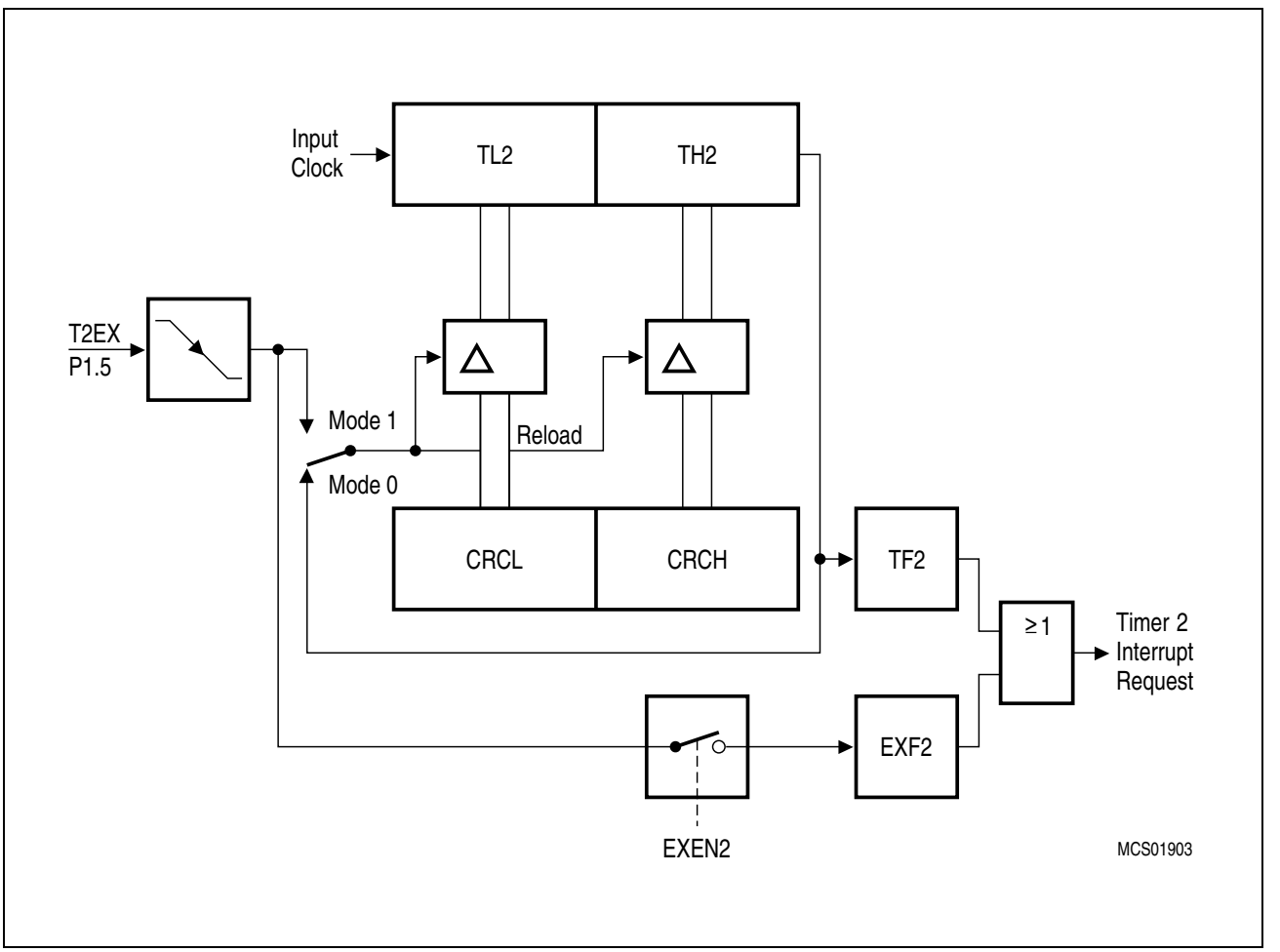


Figure 6-16
Timer 2 in Reload Mode

6.2.2.3 Compare Function of Registers CRC, CC1 to CC3

The compare function of a timer/register combination is described below.

The 16-bit value stored in a compare/capture register is compared to the contents of the timer register. If the count value in the timer register matches the stored value, an appropriate output signal is generated at a corresponding port pin, and an interrupt is requested.

The contents of a compare register can be regarded as ‘time stamp’ at which a dedicated output reacts in a predefined way (either with a positive or negative transition). Variation in this ‘time stamp’ somehow changes the wave of a rectangular output signal at a port pin. As a variation of the duty cycle of a periodic signal, this may be used for PWM as well as for a continually-controlled generation of any kind of square wave form. Two compare modes are implemented to cover a wide range of possible applications.

Compare modes 0 and 1 are selected by bit T2CM in SFR T2CON. In both compare modes, the new value arrives at the port pin 1 within the same machine cycle in which the internal compare signal is activated.

The four registers CRC, CC1 to CC3 are multifunctional as they provide a capture, compare or reload capability for the timer (CRC register only). The function is selected in register CCEN. Please note that the compare interrupt register CC0 can be programmed to be activated by either negative or positive transition. The internal compare signal (not the output signal at the port pin!) is active as long as the timer 2 contents equal the contents of one of the appropriate compare registers. Thus, when using the CRC register, it is possible to determine whether an interrupt is caused when the compare signal goes active or inactive, depending on bit I3FR in T2CON. For the CC registers 1 to 3 an interrupt is always requested when the compare signal goes active (see **Figure 6-18**).

6.2.2.3.1 Compare Mode 0

In mode 0, upon matching the timer and compare register contents, the output signal changes from low to high. It goes back to a low level on timer overflow. As long as compare mode 0 is enabled, the appropriate output pin is controlled by the timer circuit only, and not by the user. Writing to the port will have no effect. **Figure 6-17** is a functional diagram of a port latch in compare mode 0. The port latch is directly controlled by the timer overflow and compare signals. The input line from the internal bus and the write-to-latch line are disconnected when compare mode 0 is enabled.

Compare mode 0 is ideal for generating pulse width modulated output signals, which in turn can be used for digital-to-analog conversion via a filter network or by the controlled device itself (e.g., the inductance of a DC or AC motor). Mode 0 may also be used for providing output clocks with an initially-defined period and duty cycle. This is the mode that needs the least CPU time. Once set up, the output continues to oscillate without any CPU intervention. **Figures 6-18** and **6-19** illustrate the function of compare mode 0.

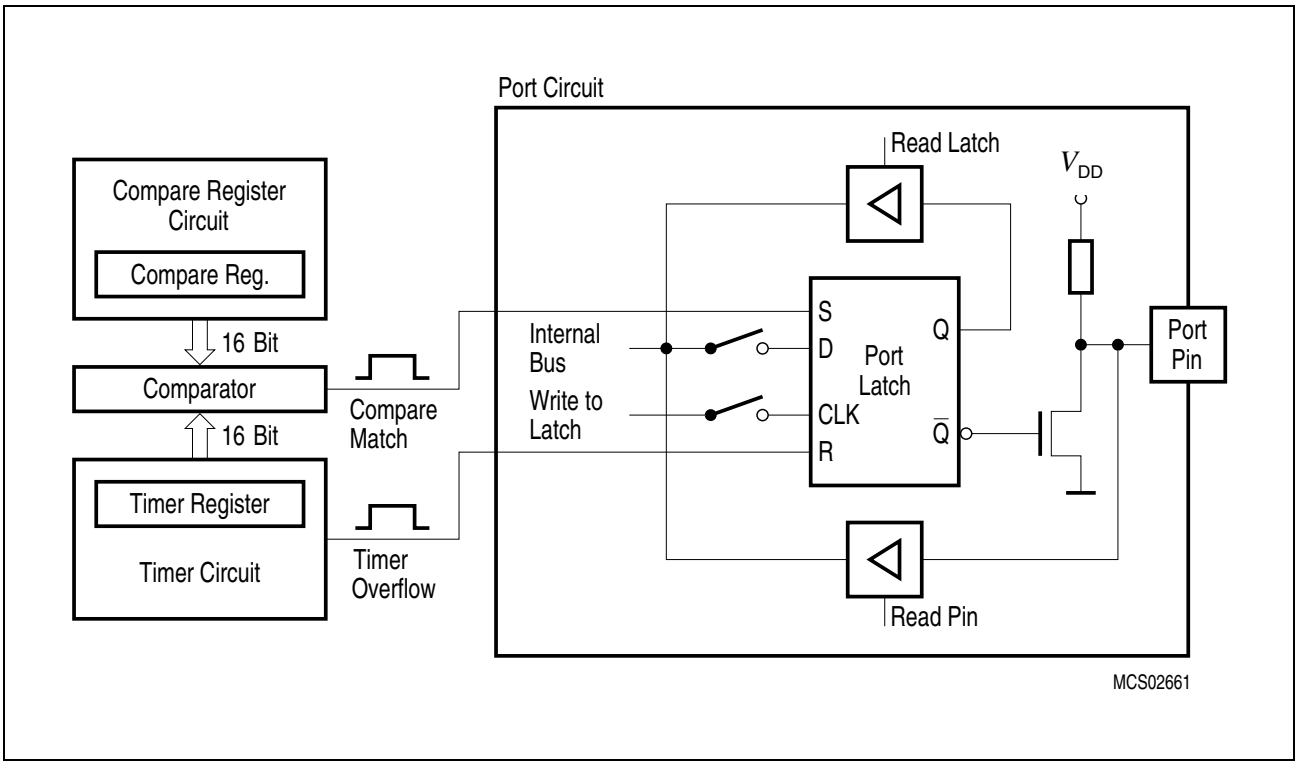


Figure 6-17
Port Latch in Compare Mode 0

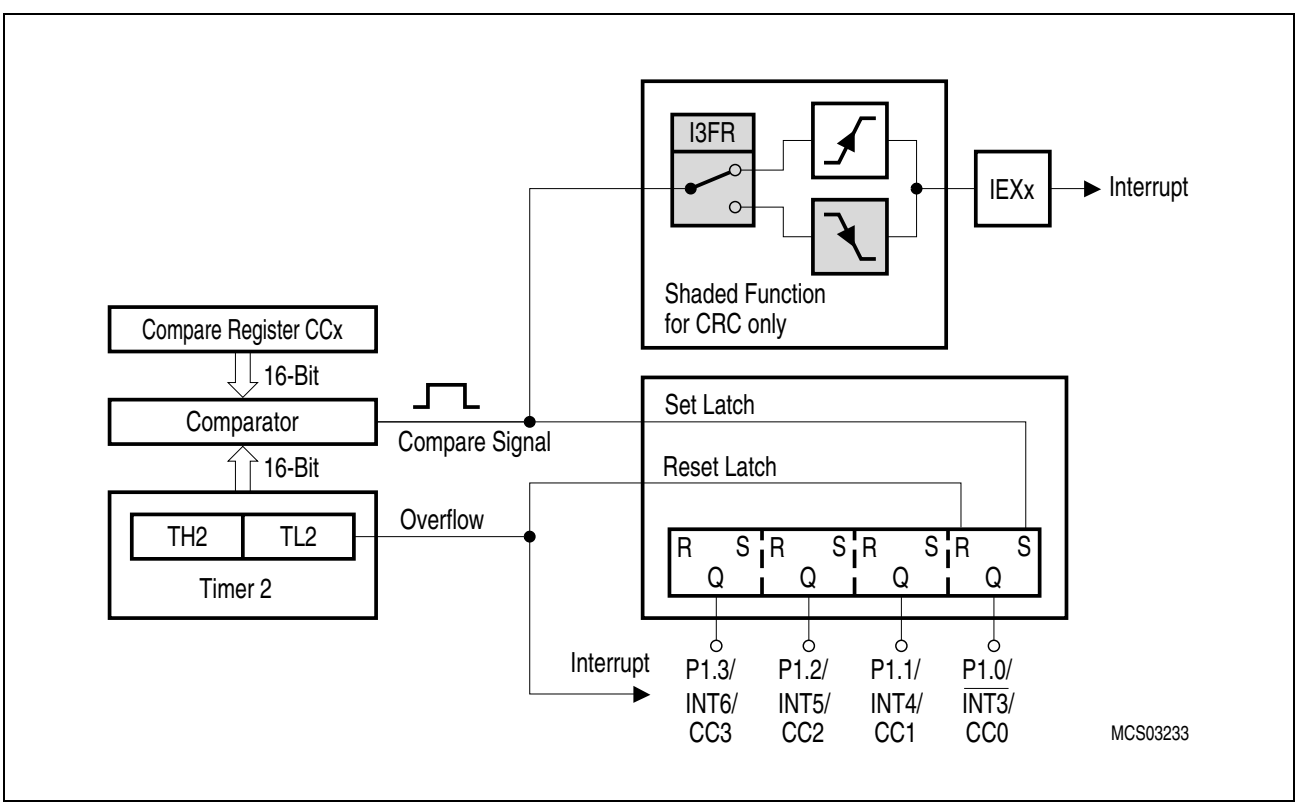


Figure 6-18
Timer 2 with Registers CCx in Compare Mode 0

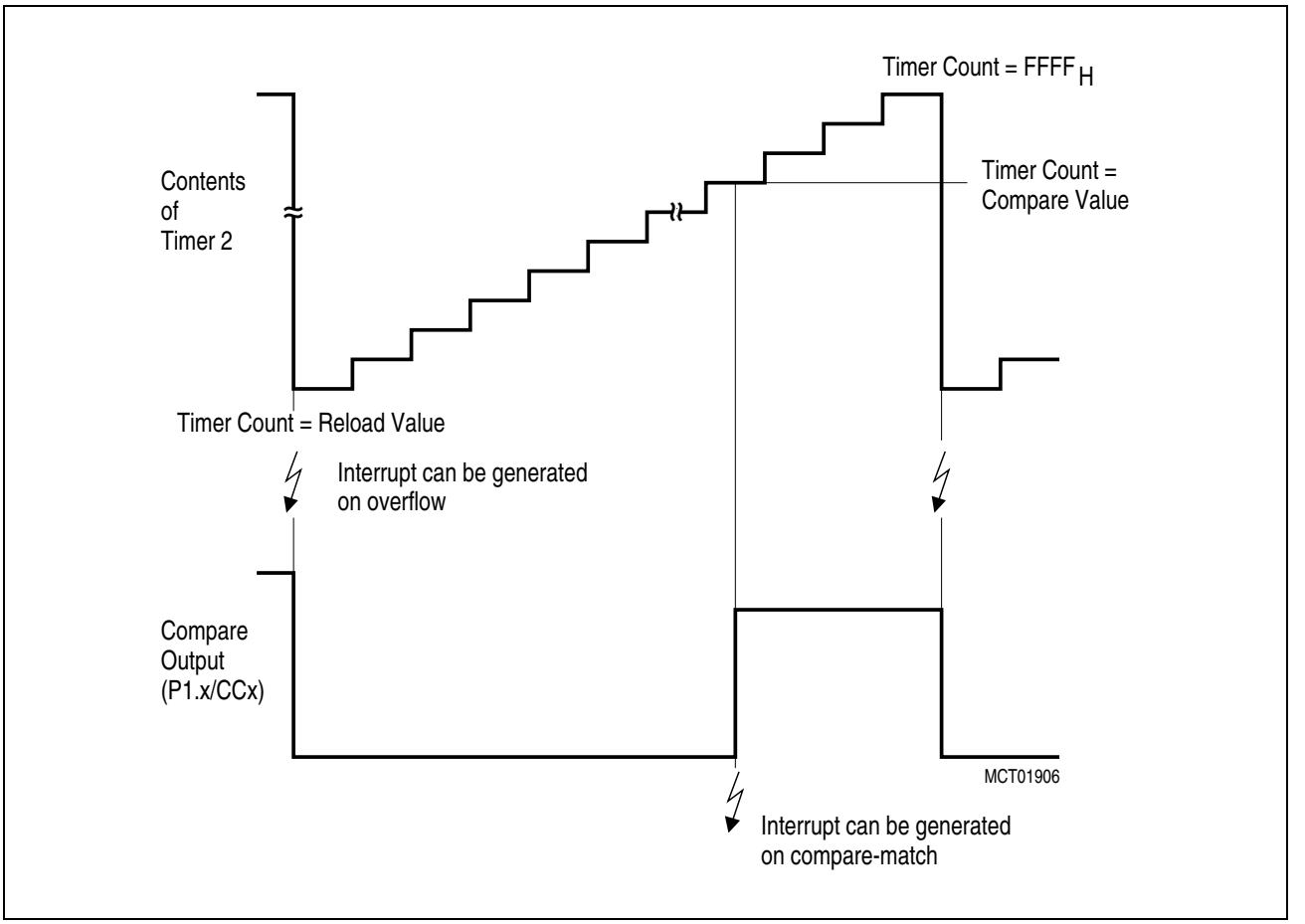


Figure 6-19
Function of Compare Mode 0

6.2.2.3.2 Modulation Range in Compare Mode 0

Generally it can be said that for every PWM generation in compare mode 0 with n-bit wide compare registers, there are 2^n different settings for the duty cycle. Starting with a constant low level (0% duty cycle) as the first setting, the maximum possible duty cycle would then be:

$$(1 - 1/2^n) \times 100\%$$

This means that a variation of the duty cycle from 0% to real 100% can never be reached if the compare register and timer register have the same length. There is always a spike that is as long as the timer clock period.

This spike may appear either when the compare register is set to the reload value (limiting the lower end of the modulation range), or it may occur at the end of a timer period. In a timer 2/CCx register configuration in compare mode 0, this spike is divided into two halves: one at the beginning when the contents of the compare register equal the reload value of the timer; the other half when the compare register contents equal the maximum value of the timer register (here: $FFFF_H$). Please refer to **Figure 6-20**, which illustrates the maximum and minimum duty cycle of a compare output signal. Timer 2 is incremented with the machine clock ($f_{OSC}/6$). Thus at 20-MHz operational frequency, these spikes are both approx. 150-ns long.

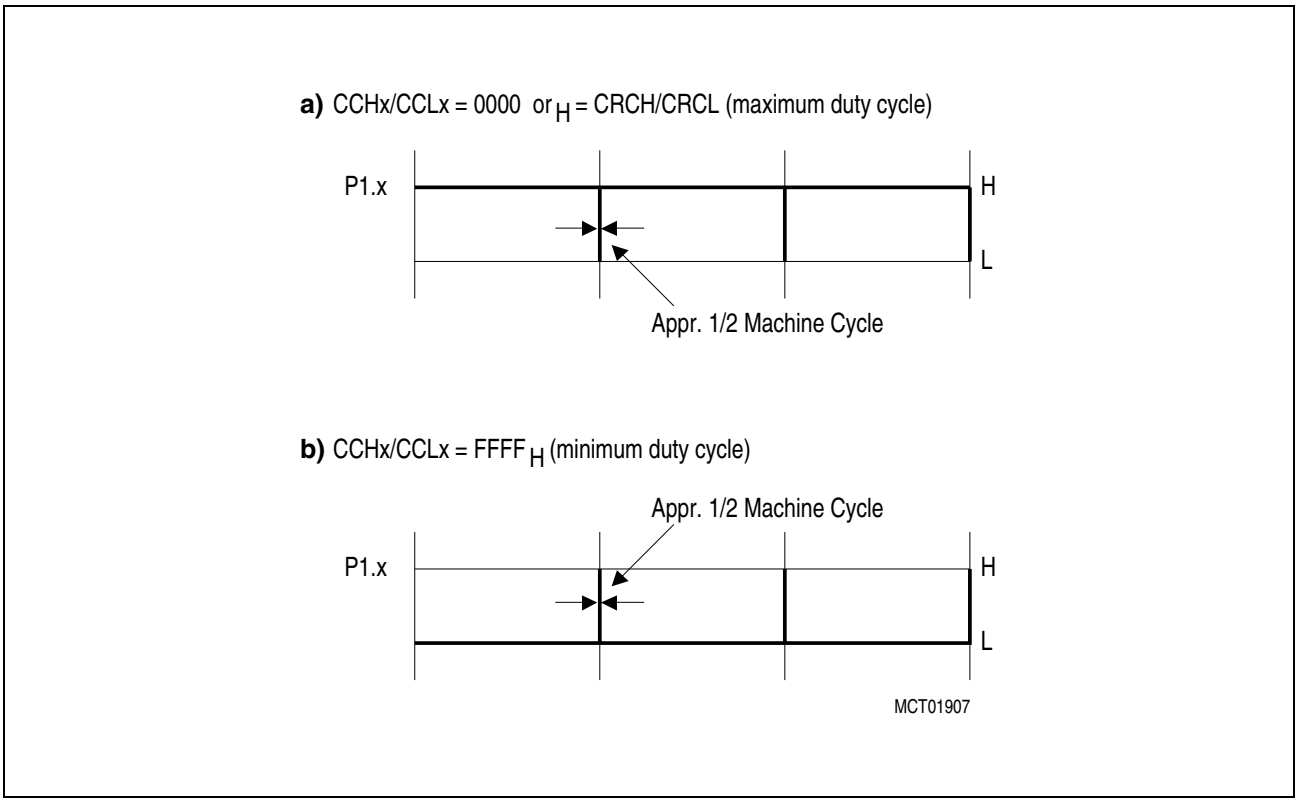


Figure 6-20
Modulation Range of a PWM Signal, generated with a Timer 2/CCx Register Combination in Compare Mode 0*

The following example shows how to calculate the modulation range for a PWM signal. To calculate with reasonable numbers, a reduction of the resolution to 8-bit is used. Otherwise for the maximum resolution of 16-bits, the modulation range would be so severely limited that it would be negligible.

Example:

Timer 2 in auto-reload mode; contents of reload register CRC = $FF00_H$

Restriction of modulation range = $1/(256 \times 2) \times 100\% = 0.195\%$

This leads to a variation of the duty cycle from 0.195% to 99.805% for a timer 2/CCx register configuration when 8 of 16 bits are used.

6.2.2.3.3 Compare Mode 1

In compare mode 1, the software adaptively determines the transition of the output signal. It is commonly used when output signals are not related to a constant signal period (as in a standard PWM Generation) but must be controlled very precisely with high resolution and without jitter. In compare mode 1, both transitions of a signal can be controlled. Compare outputs in this mode can be regarded as high-speed outputs that are independent of the CPU activity.

If compare mode 1 is enabled and the software writes to the appropriate output latch at the port, the new value will not appear at the output pin until the next compare match occurs. Thus, one can choose whether the output signal is to make a new transition (1-to-0 or 0-to-1, depending on the actual pinlevel) or keep its old value when the timer 2 count matches the stored compare value.

Figure 6-21 and **Figure 6-22** are functional diagrams of the timer/compare register/port latch configuration in compare mode 1. In this function, the port latch consists of two separate latches. The upper latch (which acts as a “shadow latch”) can be written under software control, but its value will be transferred to the output latch (and thus to the port pin) only in response to a compare match.

Note that the double latch structure is transparent as long as the internal compare signal is active. While the compare signal is active, a write operation to the port will then change both latches. This may become important when driving timer 2 with a slow external clock. In this case, the compare signal could be active for many machine cycles during which the CPU could inadvertently change the contents of the port latch.

A read-modify-write instruction will read the user-controlled shadow-latch and write the modified value back to this shadow-latch. A standard read instruction will read the pin of the corresponding compare output, as usual.

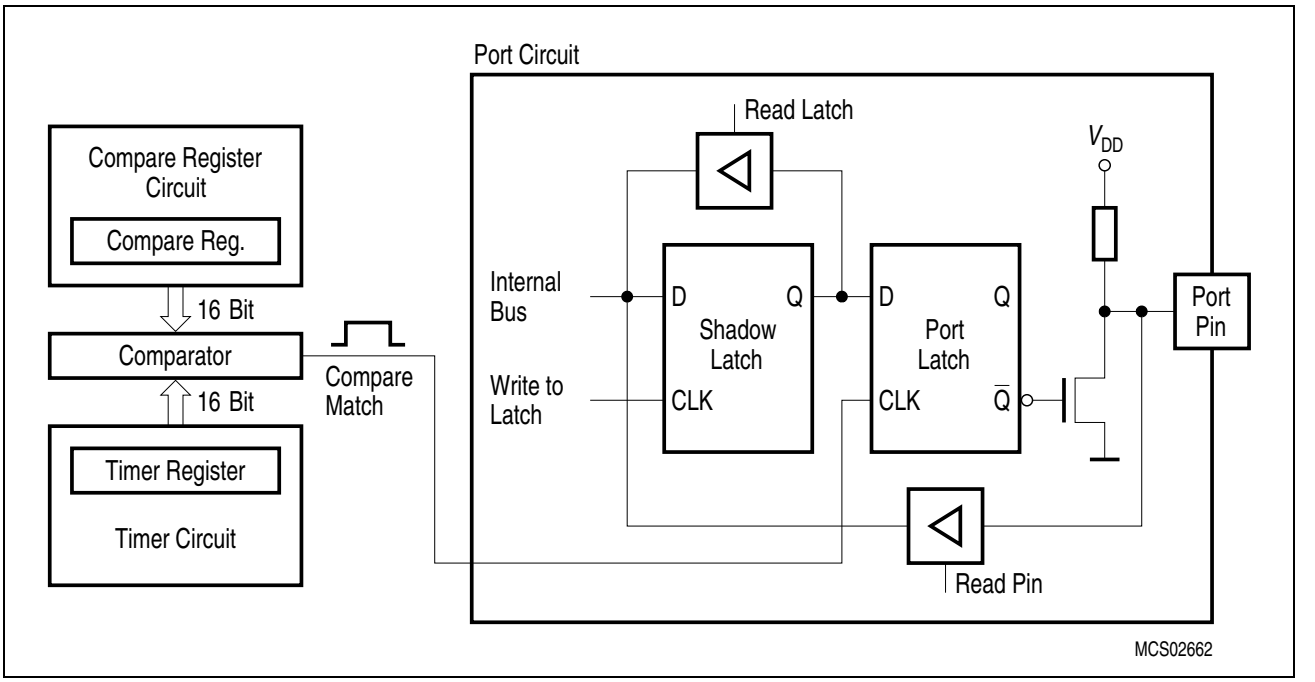


Figure 6-21
Port Latch in Compare Mode 1

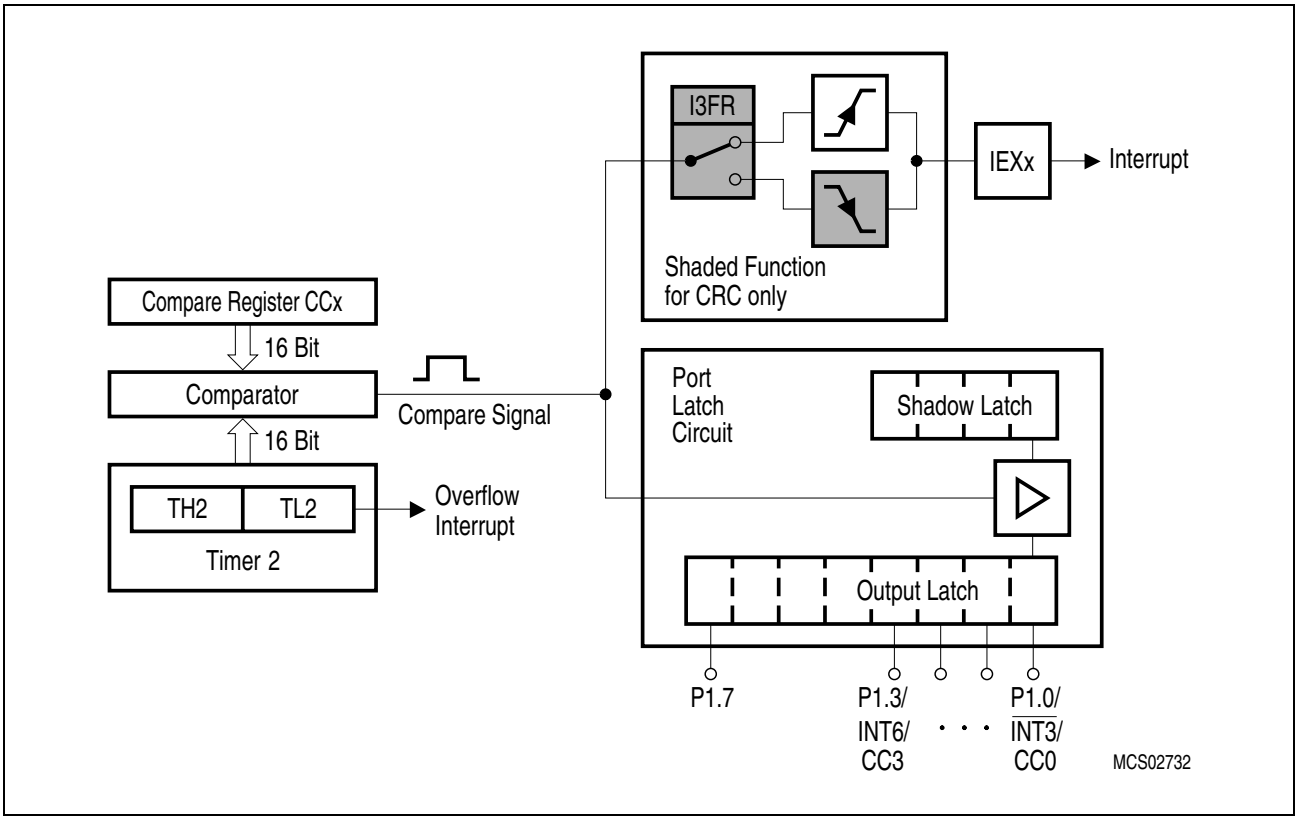


Figure 6-22
Timer 2 with Registers CCx in Compare Mode 1

(CCx stands for CRC, CC1 to CC3, IEXx stands for IEX3 to IEX6)

6.2.2.4 Using Interrupts in Combination with the Compare Function

The compare service of registers CRC, CC1, CC2 and CC3 are assigned to alternate output functions at port pins P1.0 to P1.3. Another option of these pins is that they can be used as external interrupt inputs. However, when using the port lines as compare outputs, the input line from the port pin to the interrupt system is disconnected (but the pin's level can still be read under software control). Thus, a change of the pin's level will not cause the corresponding interrupt flag to be set. In this case, the interrupt input is directly connected to the (internal) compare signal, thus providing a compare interrupt.

The compare interrupt can be used very effectively to change the contents of the compare registers or to determine the level of the port outputs for the next "compare match". The principle is that the internal compare signal (generated by a matching timer count and register contents) not only manipulates the compare output, but also sets the corresponding interrupt request flag. The current task of the CPU is interrupted if the priority of the compare interrupt is higher than the present task's priority, and the corresponding interrupt service routine is called. This service routine then sets up all the necessary parameters for the next compare event.

Advantages of Compare interrupts

There is no danger of unintentionally overwriting a compare register before a match has been reached. This could happen if the CPU wrote to the compare register without having information about the actual timer 2 count.

The most interesting advantage of the compare feature is that the output pin is exclusively controlled by hardware. Therefore, it is completely independent of any service delay, which could be disastrous in real-time applications. The compare interrupt, in turn, is not sensitive to such delays, since it loads the parameters for the next event. This, in turn, is supposed to happen after a sufficient amount of time.

In some special cases, however, a program using compare interrupts may behave surprisingly. One configuration for such a case is that described for compare mode 1. The fact that the compare interrupts are transition-activated becomes important when driving timer 2 with a slow external clock. In this case it should be carefully considered that the compare signal is active as long as the timer 2 count is equal to the contents of the corresponding compare register, and that the compare signal has a rising and a falling edge. Furthermore, the shadow latch used in compare mode 1 is transparent while the compare signal is active. With a slow input clock for timer 2, the comparator signal is active for a long time (= high number of machine cycles). Therefore, a fast interrupt-controlled reload of the compare register could change not only the shadow latch (probably intended), but also the output buffer.

When using the CRC, the programmer can select whether an interrupt should be generated when the compare signal goes active or inactive, depending on the status of bit I3FR in T2CON. Initializing the interrupt to be triggered by a negative transition is advisable in the case described above. Then the compare signal is already inactive, and any write access to the port latch just changes the contents of the shadow-latch.

Please note that for CC1 to CC3 registers, an interrupt is always requested when the compare signal goes active.

The second configuration which should be noted is when the compare function is combined with interrupts activated by negative transitions. If the port latch of port P1.0 contains a 1, the interrupt request flags IEX3 will be set immediately after enabling the compare mode for the CRC register. The reason is that first the external interrupt input is controlled by the pin's level. When the compare option is enabled, the interrupt logic input is switched to the internal compare signal, which carries a low level when no true comparison is detected. The interrupt logic sees a 1-to-0 edge and sets the interrupt request flag.

An unintentional generation of an interrupt during compare initialization can be prevented. If the request flag is cleared by software after the compare is activated, and before the external interrupt is enabled.

6.2.2.5 Capture Function

Each of the compare/capture registers CC1 to CC3 and the CRC register can be used to latch the current 16-bit value of the timer 2 registers TL2 and TH2. Two different modes are provided for this function. In mode 0, an external event latches the timer 2 contents to a dedicated capture register. In mode 1, a capture will occur upon writing to the low-order byte of the dedicated 16-bit capture register. This mode is provided to allow the software to read the timer 2 contents “on-the-fly”.

In mode 0, the external event causing a capture is:

- For CC registers 1 to 3: A positive transition at pins CC1 to CC3 of port 1
- For the CRC register: A positive or negative transition at the corresponding pin, depending on the status of the bit I3FR in SFR T2CON. If the edge flag is cleared, a capture occurs in response to a negative transition; if the edge flag is set, a capture occurs in response to a positive transition at pin P1.0 / $\overline{\text{INT3}}$ / CC0.

In both cases, the appropriate port 1 pin is used as input, and the port latch must be programmed to contain a 1. The external input is sampled in every machine cycle. When the sampled input shows a low (high) level in one cycle and a high (low) in the next cycle, a transition is recognized. The timer 2 contents is latched to the appropriate capture register in the cycle following the one in which the transition was identified.

In mode 0, a transition at the external capture inputs of registers CC1 to CC3 will also set the corresponding external interrupt request flags IEX3 to IEX6. If the interrupts are enabled, an external capture signal will cause the CPU to vector to the appropriate interrupt service routine.

In mode 1, a capture occurs in response to a write instruction to the low order byte of a capture register. The write-to-register signal (e.g. write-to-CRCL) is used to initiate a capture. The value written to the dedicated capture register is irrelevant for this function. The timer 2 contents will be latched into the appropriate capture register in the cycle following the write instruction. In this mode, no interrupt request will be generated.

Figure 6-23 illustrates the operation of the CRC register, while **Figure 6-23a** shows the operation of the compare/capture registers 1 (similarly applicable to registers 2 & 3).

The two capture modes can be established individually for each capture register by bits in SFR CCEN (compare/capture enable register). In contrast to the compare modes, it is possible to select simultaneously mode 0 for one capture register and mode 1 for another register.

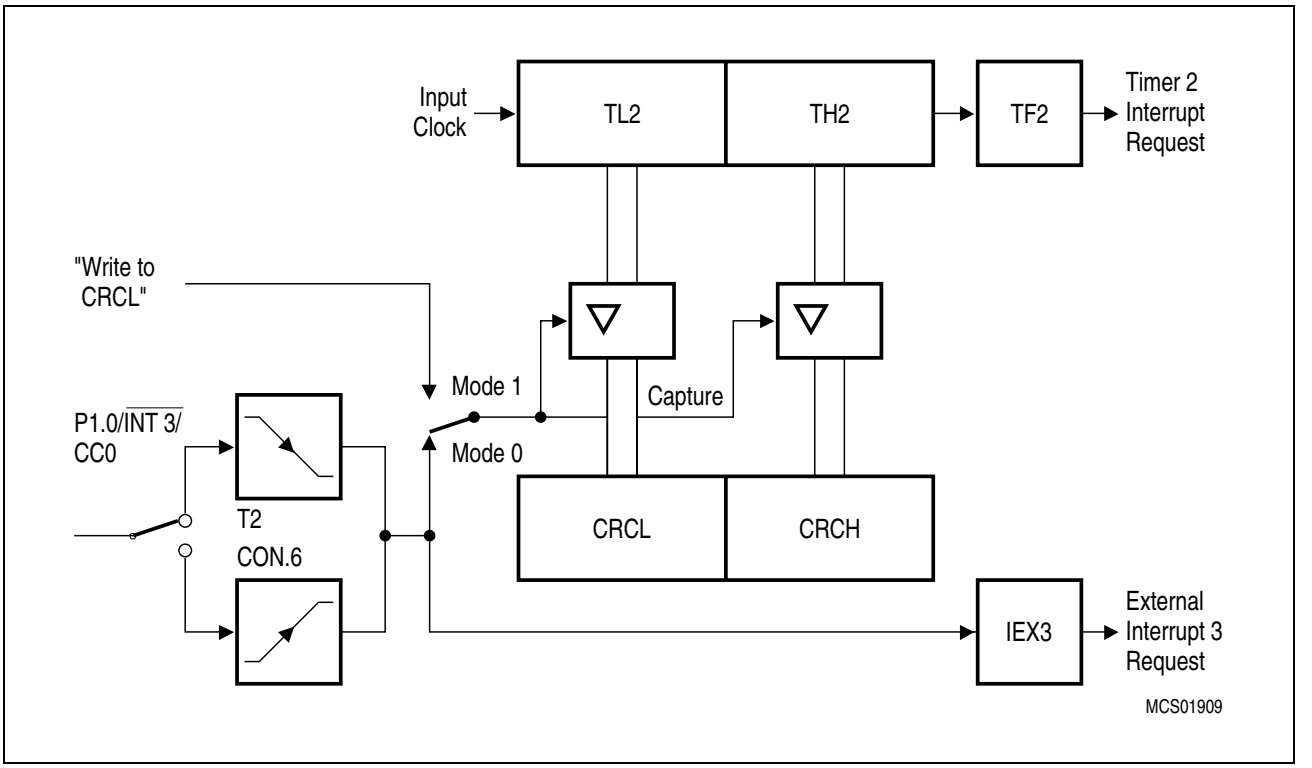


Figure 6-23
Timer 2 - Capture with Register CRC

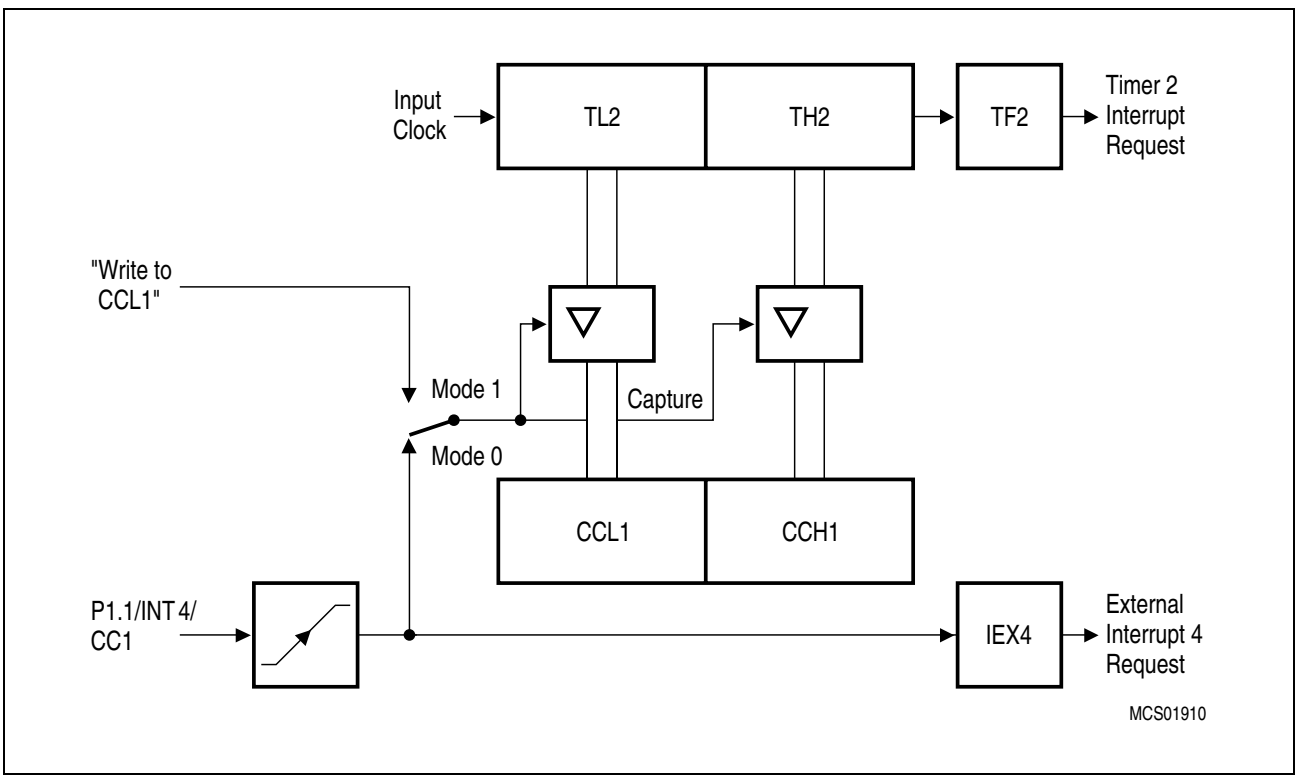


Figure 6-23a
Timer 2 - Capture with Registers CC1 to CC3

6.3 Serial Interface

The serial port of the C505L 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 via Special Function Register (SFR) 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). The baudrate clock for the serial port is derived from the oscillator frequency (mode 0, 2), or generated either by timer 1 or by a dedicated baudrate generator (mode 1, 3).

Mode 0, Shift Register (Synchronous) Mode:

Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted/received, LSB first. The baudrate is fixed at 1/6 of the oscillator frequency (see **Section 6.3.4** for more detailed information).

Mode 1, 8-Bit USART, Variable Baudrate:

Ten bits are transmitted (through TxD) or received (through RxD): A start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SFR SCON. The baudrate is variable (see **Section 6.3.5** for more detailed information).

Mode 2, 9-Bit USART, Fixed Baudrate:

Eleven bits are transmitted (through TxD) or received (through RxD): A start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned to the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in special function register SCON, while the stop bit is ignored. The baudrate is programmable to either 1/16 or 1/32 of the oscillator frequency (see **Section 6.3.6** for more detailed information).

Mode 3, 9-Bit USART, Variable Baudrate:

Eleven bits are transmitted (through TxD) or received (through RxD): A start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Mode 3 is the same as mode 2 in all respects except the baudrate. The baudrate in mode 3 is variable (see **Section 6.3.6** for more detailed information).

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

The serial interface also provides interrupt requests when transmission or reception of frames has been completed. The corresponding interrupt request flags are TI or RI, resp. See **Chapter 7** of this users manual for more details about the interrupt structure. If the serial interrupt is not enabled, the interrupt request flags TI and RI can also be used for polling the serial interface.

6.3.1 Multiprocessor Communication

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received, followed by a stop bit. The 9th data bit goes into RB8. 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 described below.

When the master processor transmits 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 being addressed. The addressed slave will clear its SM2 bit and prepare to receive the incoming data bytes. The slaves that weren't being addressed leave their SM2s set and do not receive the incoming data bytes.

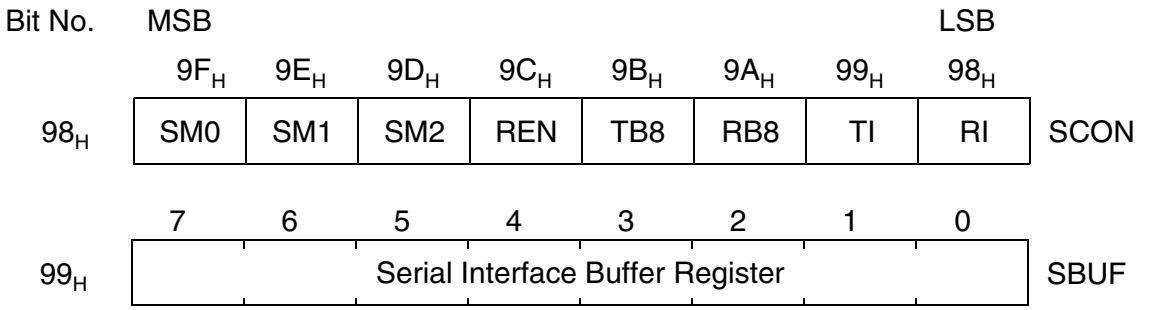
SM2 has no effect in mode 0. SM2 can be used in mode 1 to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

6.3.2 Serial Port Registers

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

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

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



Bit	Function
SM0 SM1	Serial port 0 operating mode selection bits **SM0** / **SM1** / **Selected operating mode** 0 / 0 / Serial mode 0: Shift register, fixed baudrate (f_{OSC}/6) 0 / 1 / Serial mode 1: 8-bit UART, variable baudrate 1 / 0 / Serial mode 2: 9-bit UART, fixed baudrate (f_{OSC}/16 or f_{OSC}/32) 1 / 1 / Serial mode 3: 9-bit UART, variable baudrate
SM2	Enable serial port multiprocessor communication in modes 2 and 3 In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the 9th data bit (RB8) received is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	Enable receiver of serial port Enables serial reception. Set by software to enable serial reception. Cleared by software to disable serial reception.
TB8	Serial port transmitter bit 9 TB8 is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	Serial port receiver bit 9 In modes 2 and 3, RB8 is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	Serial port transmitter interrupt flag TI is set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. TI must be cleared by software.
RI	Serial port receiver interrupt flag RI is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (exception see SM2). RI must be cleared by software.

6.3.3 Baudrate Generation

There are several possibilities to generate the baud-rate clock for the serial port depending on the mode of operation.

For clarification the difference between “baud-rate clock” and “baudrate” must be defined. The serial interface requires a clock rate which is 16 times the baudrate for internal synchronization. Therefore, the baudrate generators have to provide a “baudrate clock” to the serial interface which - divided by 16 - results in the actual “baudrate”. However, all formulas given in the following section already include the factor and calculate the final baudrate. Further, the abbreviation f_{OSC} refers to the external clock frequency (oscillator or external input clock operation).

The baudrate of the serial port is controlled by two bits that are located in the SFRs shown below.

Special Function Register ADCON0 (Address $D8_H$) **Reset Value: $00X00000_B$**
Special Function Register PCON (Address 87_H) **Reset Value: 00_H**

Bit No.	MSB							LSB	
	DF_H	DE_H	DD_H	DC_H	DB_H	DA_H	$D9_H$	$D8_H$	
$D8_H$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	ADCON0
	7	6	5	4	3	2	1	0	
87_H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

The shaded bits are not used for controlling the baudrate.

Bit	Function
BD	Baud-rate generator enable When set, the baudrate of serial interface is derived from the dedicated baudrate generator. When cleared (default after reset), baudrate is derived from the timer 1 overflow rate.
SMOD	Double baudrate When set, the baudrate of serial interface in modes 1, 2, 3 is doubled. After reset this bit is cleared.
–	Reserved bits for future use. Read by CPU returns undefined values.

Figure 6-24 shows the configuration for the baudrate generation of the serial port.

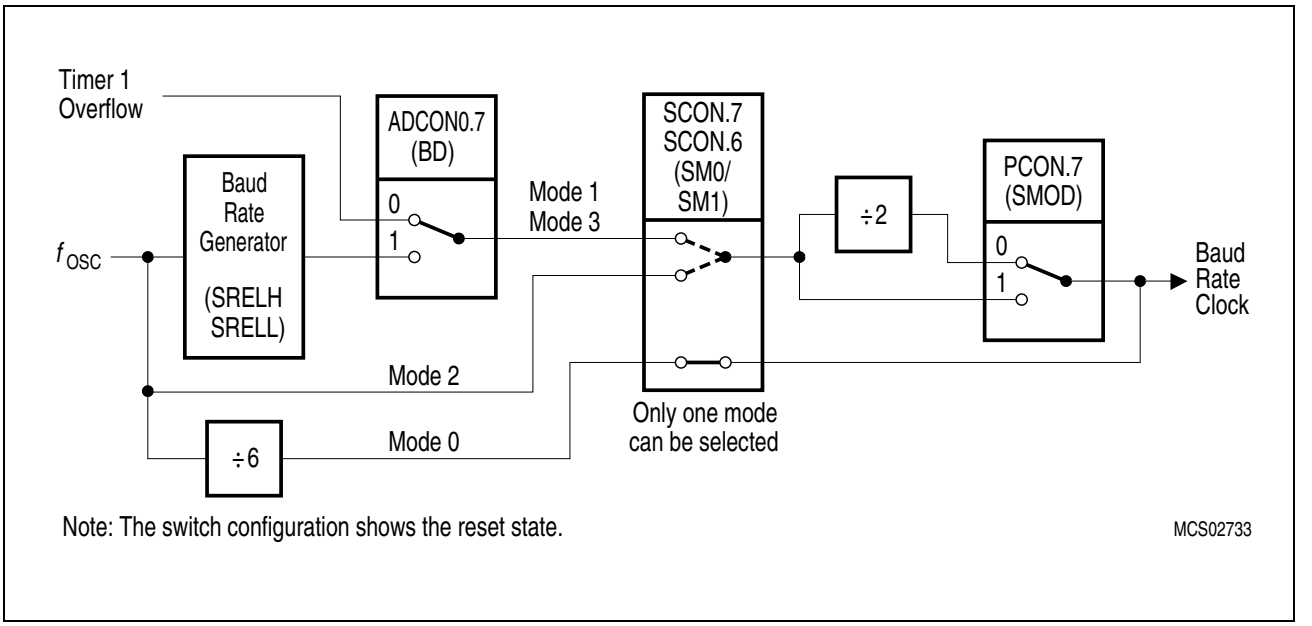


Figure 6-24
Baud-rate Generation for the Serial Port

Depending on the programmed operating mode, different paths are selected for the baud-rate clock generation. **Figure 6-24** shows the dependencies of the serial port baud-rate clock generation on the two control bits, and on the mode selected in the SFR SCON.

6.3.3.1 Baudrate in Mode 0

The baudrate in mode 0 is fixed to:

$$\text{Mode 0 baudrate} = \frac{\text{oscillator frequency}}{6}$$

6.3.3.2 Baudrate in Mode 2

The baudrate in mode 2 depends on the value of bit SMOD in SFR PCON. If SMOD = 0 (which is the value after reset), the baudrate is 1/32 of the oscillator frequency. If SMOD = 1, the baudrate is 1/16 of the oscillator frequency.

$$\text{Mode 2 baudrate} = \frac{2^{\text{SMOD}}}{32} \times \text{oscillator frequency}$$

6.3.3.3 Baudrate in Mode 1 and 3

In these modes the baudrate is variable and can be generated alternatively by a baudrate generator or by timer 1.

6.3.3.3.1 Using the Internal Baudrate Generator

In modes 1 and 3, the C505L can use an internal baudrate generator for the serial port. To enable this feature, bit BD (bit 7 of SFR ADCON0) must be set. Bit SMOD (PCON.7) controls a divide-by-2 circuit which affect the input and output clock signal of the baudrate generator. After reset the divide-by-2 circuit is active and the resulting overflow output clock will be divided by 2. The input clock of the baudrate generator is f_{OSC}.

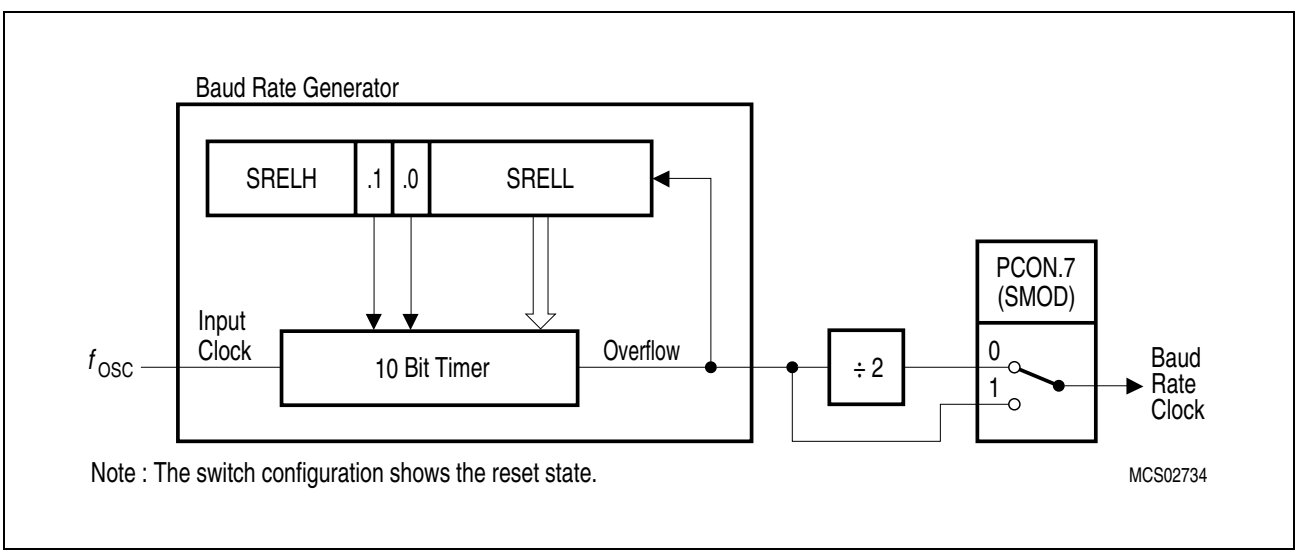
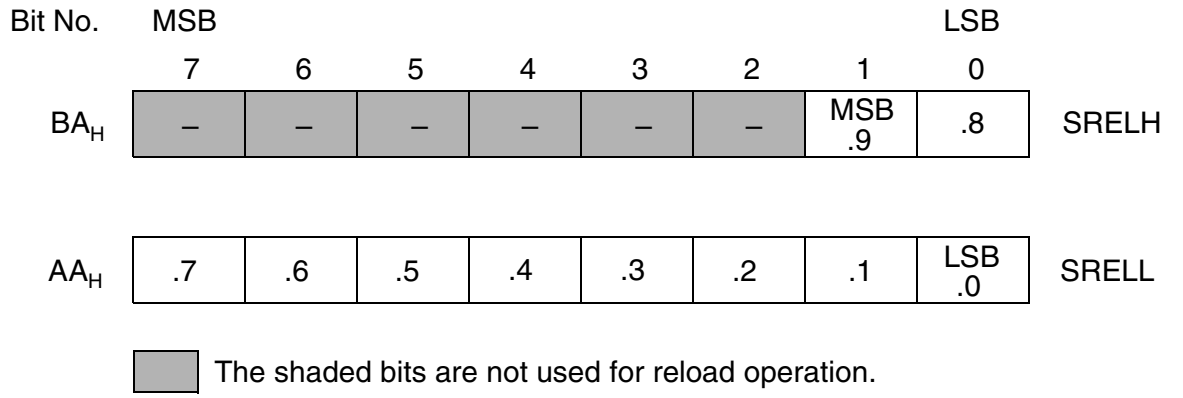


Figure 6-25
Serial Port Input Clock when using the Baudrate Generator

The baudrate generator consists of a free-running upward counting 10-bit timer. On overflow of this timer (next count step after counter value $3FF_H$), there is an automatic 10-bit reload from the registers SRELL and SRELH. 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 baudrate timer is reloaded by writing to SRELL.

Special Function Register SRELH (Address BA_H) **Reset Value: $XXXXXX11_B$**
Special Function Register SRELL (Address AA_H) **Reset Value: $D9_H$**



Bit	Function
SRELH.0-1	Baudrate generator reload high value Upper two bits of the baudrate timer reload value.
SRELL.0-7	Baudrate generator reload low value Lower 8 bits of the baudrate timer reload value.
–	Reserved bits for future use. Read by CPU returns undefined values.

After reset, SRELH and SRELL have a reload value of $3D9_H$. With this reload value, the baudrate generator has an overflow rate of input clock/39. With a 6 MHz oscillator frequency, the commonly used baudrates 4800 (SMOD = 0) and 9600 (SMOD = 1) are available (with 0.16% deviation).

With the baudrate generator as clock source for the serial port in mode 1 and 3, the baudrate of the serial port can be determined as follows:

$$\text{Mode 1, 3 baudrate} = \frac{2^{SMOD} \times \text{oscillator frequency}}{32 \times (\text{baudrate generator overflow rate})}$$

$$\text{Baudrate generator overflow rate} = 2^{10} - \text{SREL}$$

with SREL = SRELH.1 – 0, SRELL.7 – 0

6.3.3.3.2 Using Timer 1 to Generate Baudrates

In mode 1 and 3 of the serial port, timer 1 can also be used for generating baudrates. Then the baudrate is determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Mode 1, 3 baudrate} = \frac{2^{SMOD}}{32} \times (\text{timer 1 overflow rate})$$

The timer 1 interrupt is usually disabled in this application. Timer 1 itself can be configured for either “timer” or “counter” operation in any of its operating modes. In most typical applications, it is configured for “timer” operation in the auto-reload mode (high nibble of TMOD = 0010_B). In this case the baudrate is given by the formula:

$$\text{Mode 1, 3 baudrate} = \frac{2^{SMOD} \times \text{oscillator frequency}}{32 \times 6 \times (256 - (TH1))}$$

Very low baudrates can be achieved with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001_B), and using the timer 1 interrupt for a 16-bit software reload.

6.3.4 Details about Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted/received, LSB first. The baudrate is fixed at $f_{OSC}/6$.

Figure 6-26 is a simplified functional diagram of the serial port in mode 0. The associated timing is illustrated in **Figure 6-27**.

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 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 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 RI = 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 bits come 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.

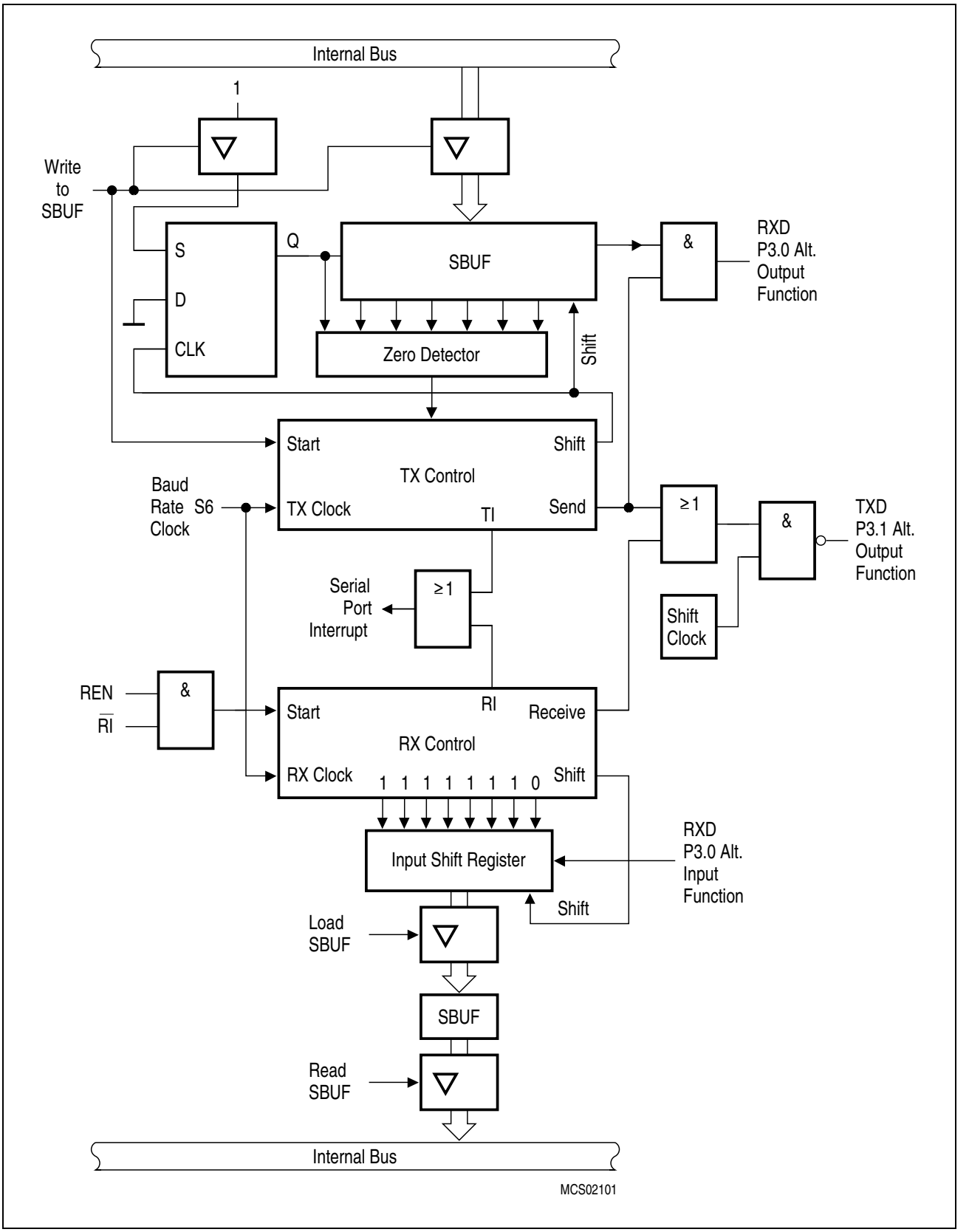


Figure 6-26
Serial Interface, Mode 0, Functional Diagram

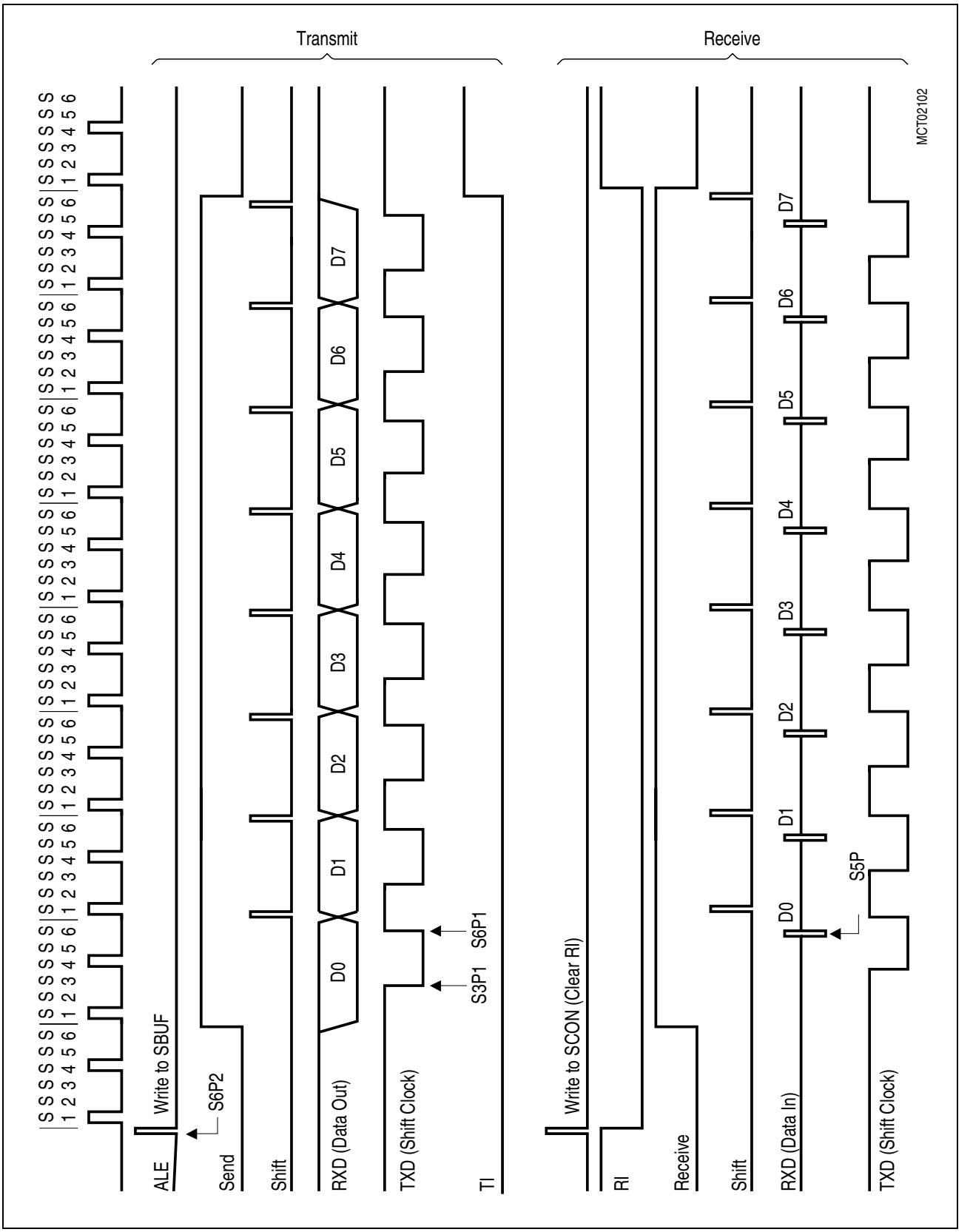


Figure 6-27
Serial Interface, Mode 0, Timing Diagram

6.3.5 Details about Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into RB8 in SCON. The baudrate is determined either by the timer 1 overflow rate or by the internal baudrate generator.

Figure 6-28 is a simplified functional diagram of the serial port in mode 1. The associated timings for transmit/receive are illustrated in **Figure 6-29**.

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 detecting a 1-to-0 transition at RxD. For this purpose, RxD is sampled at a rate of 16 times the baudrate that has been established. When a transition is detected, the divide-by-16 counter is reset immediately, 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 least 2 of the 3 samples. This kind of sampling is done for 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 look for another 1-to-0 transition, thereby providing rejection of false start bits. If the start bit is 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 is a 9-bit register in mode 1), 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 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 one of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bit goes into SBUF, and RI is activated. At this time, whether the conditions are met or not, the unit goes back to look for a 1-to-0 transition in RxD.

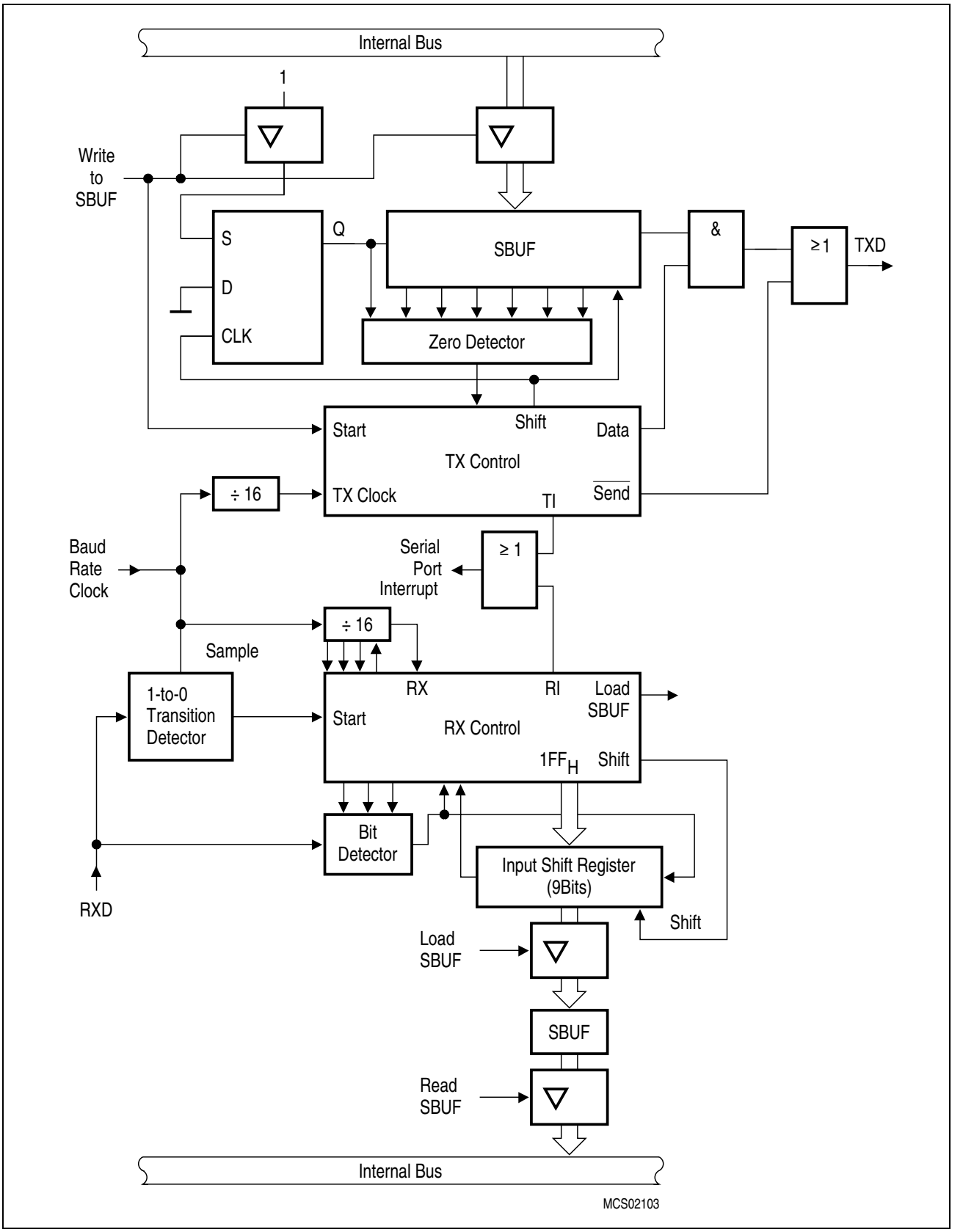


Figure 6-28
Serial Interface, Mode 1, Functional Diagram

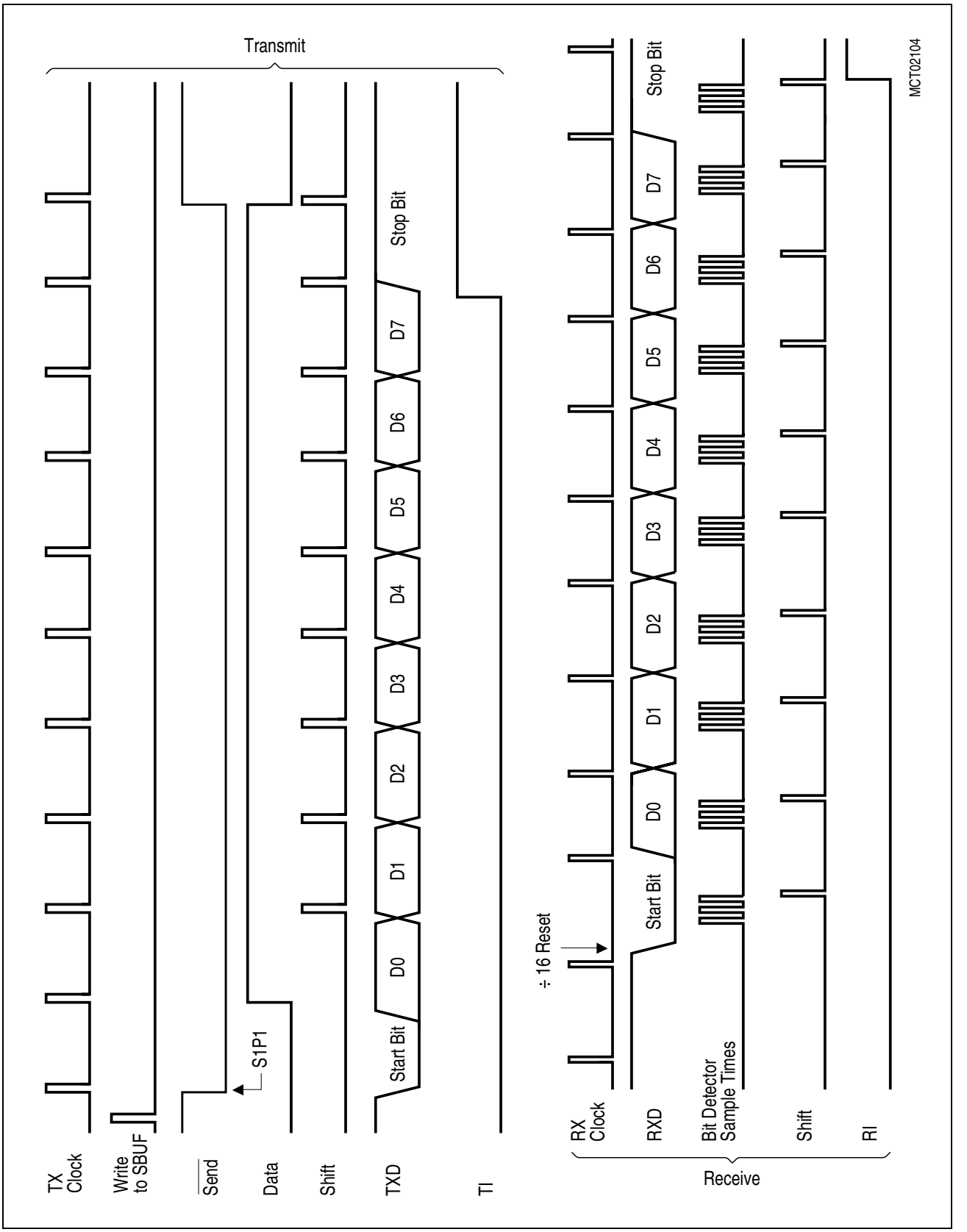


Figure 6-29
Serial Interface, Mode 1, Timing Diagram

6.3.6 Details about Modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmission, the 9th data bit (TB8) can be assigned the value of 0 or 1. On reception, the 9th data bit goes into RB8 in SCON. The baudrate is programmable to either 1/16 or 1/32 the oscillator frequency in mode 2. When bit SMOD in SFR PCON (87_H) is set, the baudrate is $f_{OSC}/16$. In mode 3, the baudrate clock is generated by timer 1, which is incremented by a rate of $f_{OSC}/6$ or by the internal baudrate generator.

Figure 6-30 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 timing for transmit/receive are illustrated in **Figure 6-31**.

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 transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only 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 condition flags the TX control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after “Write-to-SBUF”.

Reception is initiated when a 1-to-0 transition is detected at RxD. For this purpose, RxD is sampled at a rate of 16 times the baudrate that has been established. When a transition is detected, the divide-by-16 counter is reset immediately, 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 bits come from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which is a 9-bit register in modes 2 and 3), 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 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.

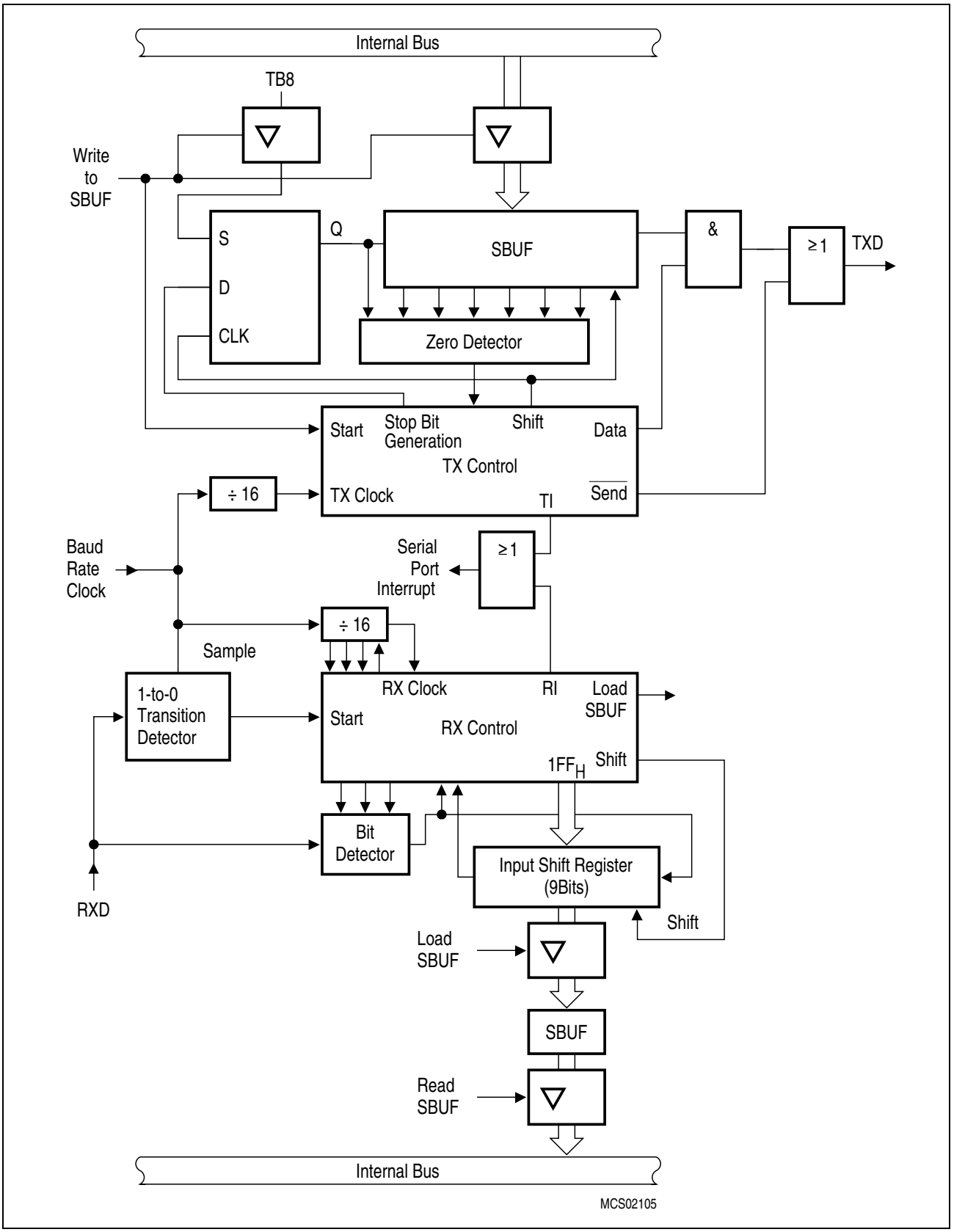


Figure 6-30
Serial Interface, Mode 2 and 3, Functional Diagram

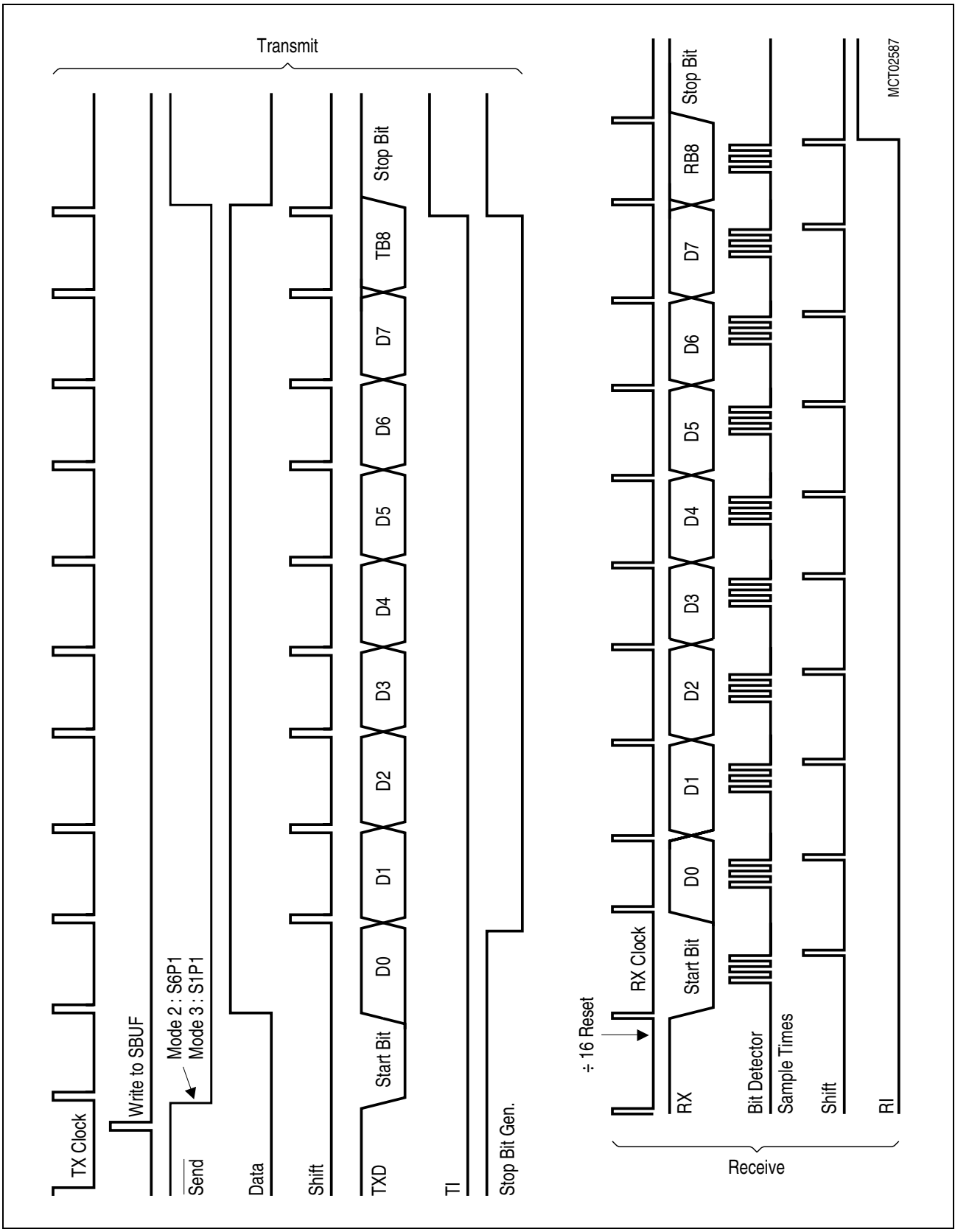


Figure 6-31
Serial Interface, Mode 2 and 3, Timing Diagram

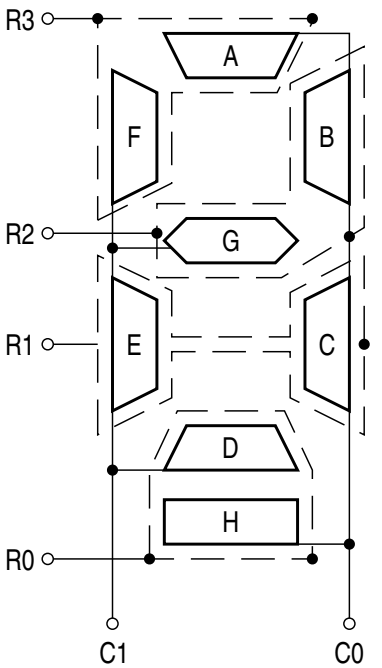
6.4 LCD Controller Unit

The Liquid Crystal Display (LCD) controller unit in the C505L is designed for the control of an LCD display module of 128 display segments (4 rows and 32 columns) using the 1/4 duty-cycle driving method. The C505L can be programmed to generate reference voltages for adjusting the contrast of the display.

6.4.1 Functionality

6.4.1.1 Display Module Organization

An example of a typical LCD module is shown in **Figure 6-32**. The table describes the different combinations of the row and column signals required to activate a particular segment. The signals R0-R3 and C0-C31 are the row and column signals, respectively, connected to the display module.



Rows	Columns	
	C1	C0
R3	F	A
R2	G	B
R1	E	C
R0	D	H

MCD03858

Figure 6-32
Organization of a Typical LCD Display Module

6.4.2 LCD Registers

The memory required by the LCD controller includes a control register and 16 individual digit registers. These registers are implemented in the on-chip external data memory area. Accesses to these registers are similar to on-chip XRAM accesses (MOVX instructions) and therefore must be preceded by an enable operation on the on-chip XRAM. The registers are described below.

6.4.2.1 Control Register

In order to display a character, LCON (the control register for the LCD display) must be configured first.

LCD Control Register, LCON (Address $F3DD_H$) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$F3DD_H$	DSB1	DSB0	0	0	0	0	CSEL	LCEN	LCON

Bit	Function
DSB1	Columns C24 to C31 disable DSB1 = 0: Disable output column lines C24 to C31. DSB1 = 1: Enable output column lines C24 to C31. Columns C24 to C31 are disabled by default and are used as digital I/O
DSB0	Columns C16 to C23 disable DSB0 = 0: Disable output column lines C16 to C23. DSB0 = 1: Enable output column lines C16 to C23. Segments C16 to C23 are disabled by default and are used for digital I/O.
CSEL	LCD controller input clock selection CSEL = 1: Use RTC Clock input (32.768 KHz) for f_{LCDIN} CSEL = 0: Use system clock (f_{OSC}) for f_{LCDIN}
LCEN	LCD controller enable LCEN = 0: LCD controller is disabled (default after reset). LCEN = 1: LCD controller is enabled.
0	These bits are reserved and should be always written with 0. Writing a 1 into these bits will give undefined results.

6.4.2.2 Digit Registers

The LCD controller has 16 registers that are used to control the 32 column outputs. Each digit register contains eight bits that control the individual segments in that digit. For example, the bit SEGA in register DIG0 controls the segment A in digit 0 of the display. A value of ‘1’ in this bit would represent an active segment, whereas a ‘0’ would represent an inactive segment. This information is used together with the row/backplane voltage to drive the LCD display.

LCD Digit Register DIGx (x = 0 to F) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$F3Ex_H$	SEGF	SEGA	SEGG	SEGB	SEGE	SEGC	SEGH	SEGD	DIGx

6.4.3 LCD Input Clock

Most of the LCD display modules require row and column signals at a maximum frequency of 360 Hz in order to activate a display segment. In order to achieve this 360-Hz frequency limit, the LCD controller uses a 15-bit timer. This timer has a programmable 15-bit reload register as shown below.

LCD Timer Reload Low Register, LCRL (address $F3DE_H$) **Reset Value: 00_H**
LCD Timer Reload High Register, LCRH (address $F3DF_H$) **Reset Value: 00_H**

Bit No.	7	6	5	4	3	2	1	0	
$F3DF_H$	SLT	MSB	–	–	–	–	–	–	LCRH
$F3DE_H$	–	–	–	–	–	–	–	LSB	LCRL

Bit	Function
SLT	LCD timer control SLT = 1: Start LCD Clock generator SLT = 0: Stop LCD Clock generator
LCRH.7-LCRL.0	LCD timer reload value (15-bit) This register should be written with a suitable 15-bit value to control the LCD signal frequency (please see table in **Figure 6-33**).

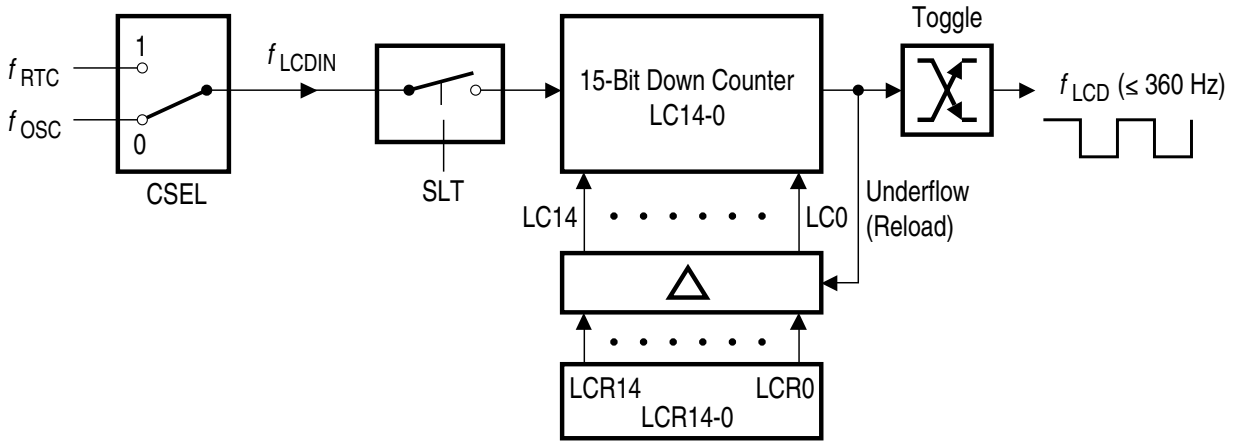
The bit SLT in the LCRH register should be set to start the 15-bit timer. The timer register is then loaded with the contents of the reload register and the timer begins counting down. When the SLT bit is cleared, the counting is stopped and the timer register is reset to 0000_H. The timer cannot continue from its previous count once it is stopped.

The LCD clock generator starts counting with a value in the reload register. It is a down counter that decrements once every input clock, f_{LCDIN}, up to 0000_H. When an underflow occurs, the LCD clock is toggled and the timer is reloaded with value in the 15-bit reload register LCR (LCRH.6-LCRL.0) and continues counting. Initially the LCD clock has a value of '1' as shown in **Figure 6-33**.

The frequency of the LCD clock is:

$$f_{LCD} = \frac{f_{LCDIN}}{2 \times \text{(15-bit reload value)}} \text{ Hz}$$

The generated LCD clock has a duty-cycle of 50%, since two count cycles (from the reload value to 0000_H) will be used to generate one LCD clock cycle. The table in **Figure 6-33** shows the recommended reload values at different input frequencies (f_{LCDIN}) to generate LCD clocks of frequencies less than 360 Hz.



f_{LCDIN}	LCR (15-Bit reload)	f_{LCD} (in Hz)
32.768 kHz	$002E_H$	356.17
2 MHz	$0ADA_H$	359.97
4 MHz	$15B4_H$	359.97
6 MHz	$208E_H$	359.97
10 MHz	3642_H	359.97
12 MHz	$411C_H$	359.97
16 MHz	$56D0_H$	359.97
20 MHz	$6C84_H$	359.97

MCD03859

Figure 6-33
LCD Clock Generation

6.4.3.1 LCD Input Clock Selection

The input clock, f_{LCDIN}, into the LCD clock generation logic can be selected with the bit CSEL in the LCON register. This bit makes it possible to chose one of the clock sources for the LCD timer: The system clock (f_{OSC}) or the 32.768 KHz clock input into the real-time clock circuit, f_{RTC}. In either case the LCR register can be programmed to derive the f_{LCD} clock frequency.

The main purpose of the CSEL bit is to ensure that the LCD controller can still function when the C505L enters power down mode 2. In this mode, the system clock (f_{OSC}) is not available and only the real-time clock is running.

6.4.4 Row Signals

The LCD controller generates four row/backplane signals. These four signals have fixed timing and levels, without any dependency on the individual digit register values. The row signals in combination with the column signals, which depend on the respective digit register values, determine the segments are to be activated or deactivated.

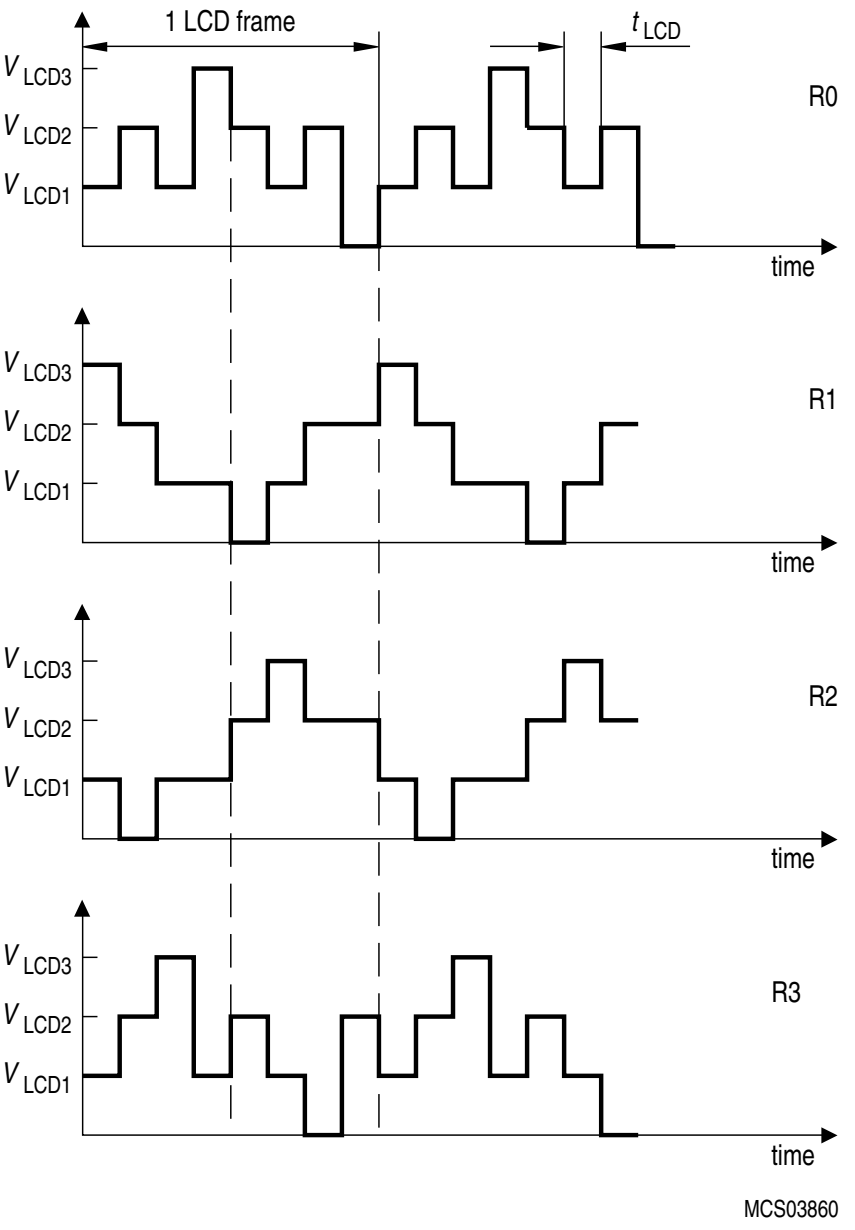


Figure 6-34
Row/Backplane Signals

The row signals, R0-R3, are as shown in **Figure 6-34**. Here t_{LCD} is the clock period of the LCD clock (f_{LCD} < 360 Hz). An LCD frame consists 8 periods of the LCD input clock.

6.4.5 Column Signals

The column signals determine the difference voltages across each segment with respect to the row signal, depending on the digit register values. Column outputs are organized in groups of two for the 1/4 duty-cycle driving supported by the C505L.

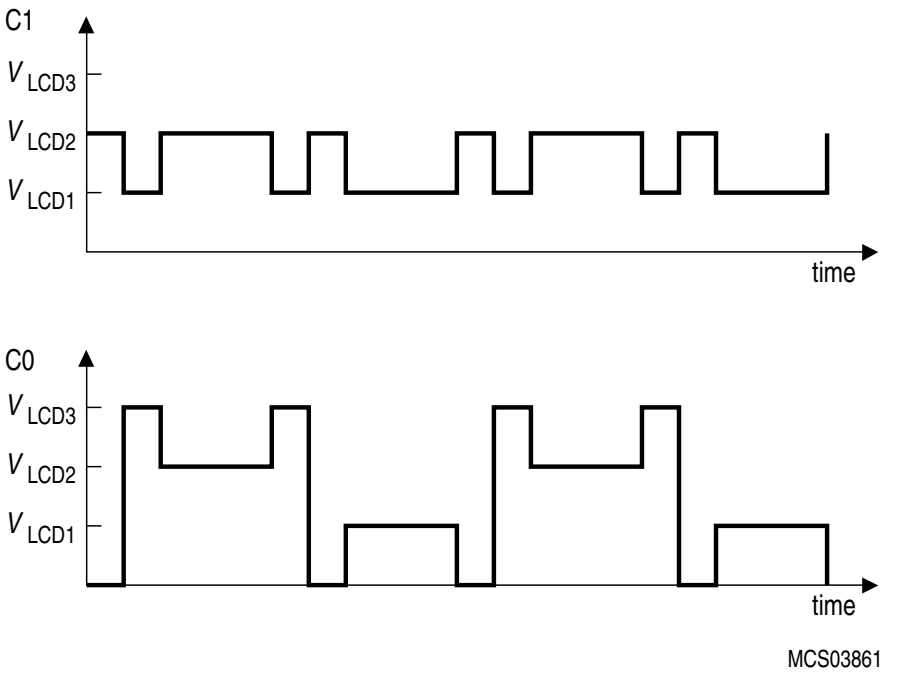


Figure 6-35
Column Signals to display ‘1’

Figure 6-35 shows the segment signals generated by the LCD controller when a digit register is configured to display the number ‘1’. **Figures 6-36**, **6-37** and **6-38** show the difference signals while using the column signals as shown above.

The segments that are not activated in a digit have difference voltages across them that are not recognized by the LCD display module. The active segments, however, have fluctuating difference voltages during an LCD frame. These difference voltages are recognized by the LCD display module. In the example above, the difference signal has an amplitude of $\pm V_{LCD3}$ for the segments B and C (twice within an LCD frame).

Note: The actual segment organization within the display unit could be different from the example considered here. In such cases, the segment names/positions may vary. User should consult the manufacturer of the LCD display unit regarding the display unit’s segment organization.

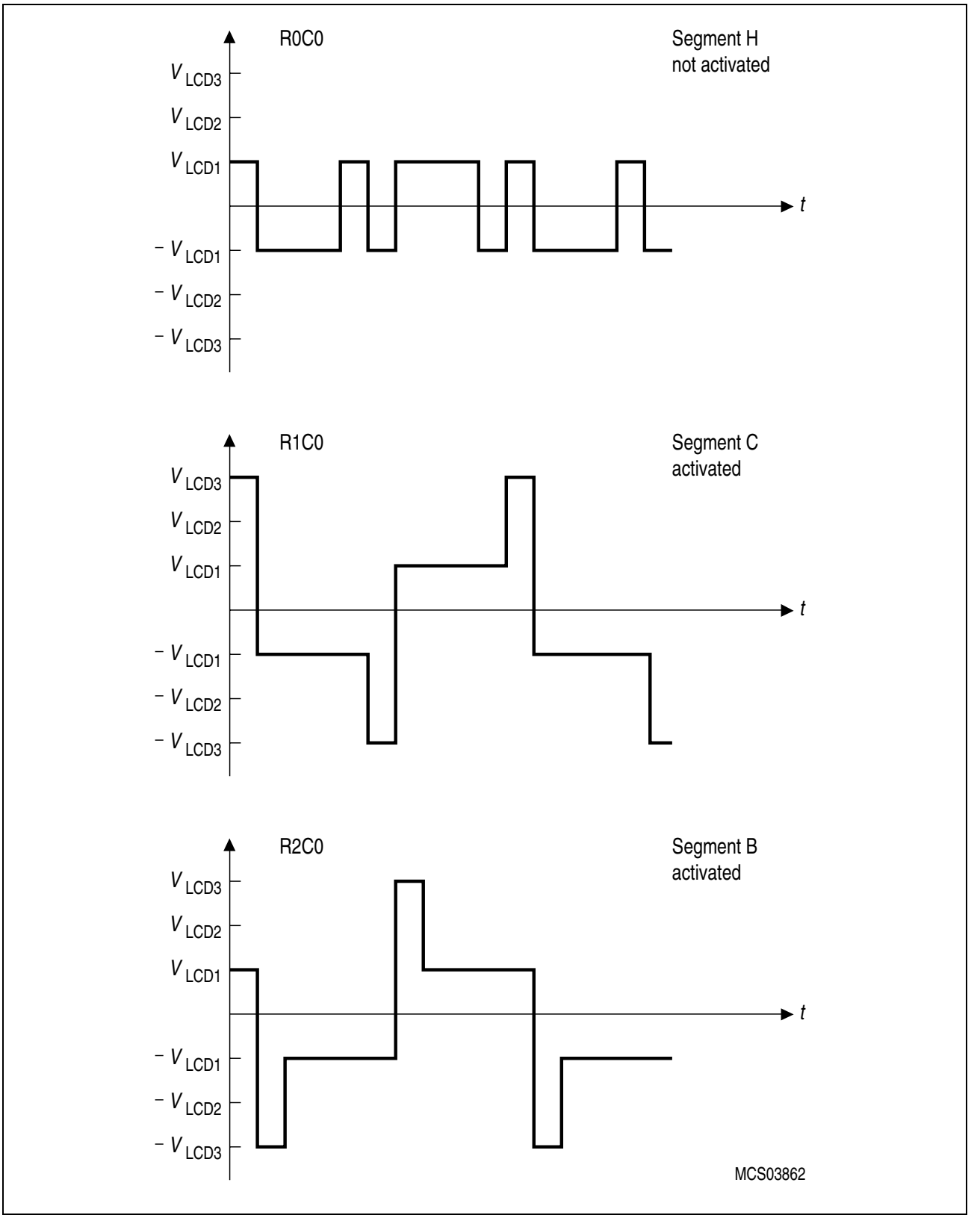


Figure 6-36
Difference Signals to display ‘1’ - Part 1

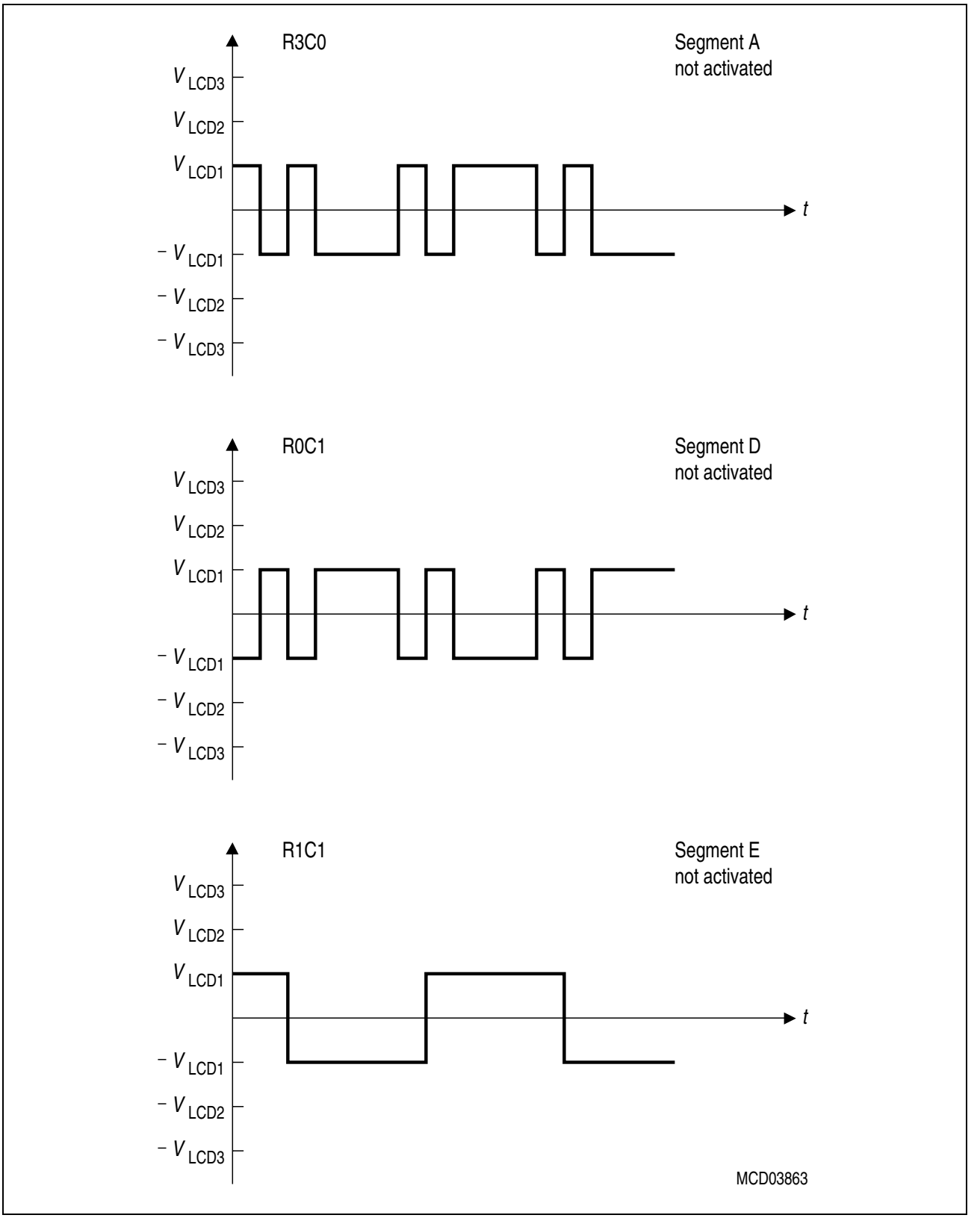


Figure 6-37
Difference Signals to display ‘1’ - Part 2

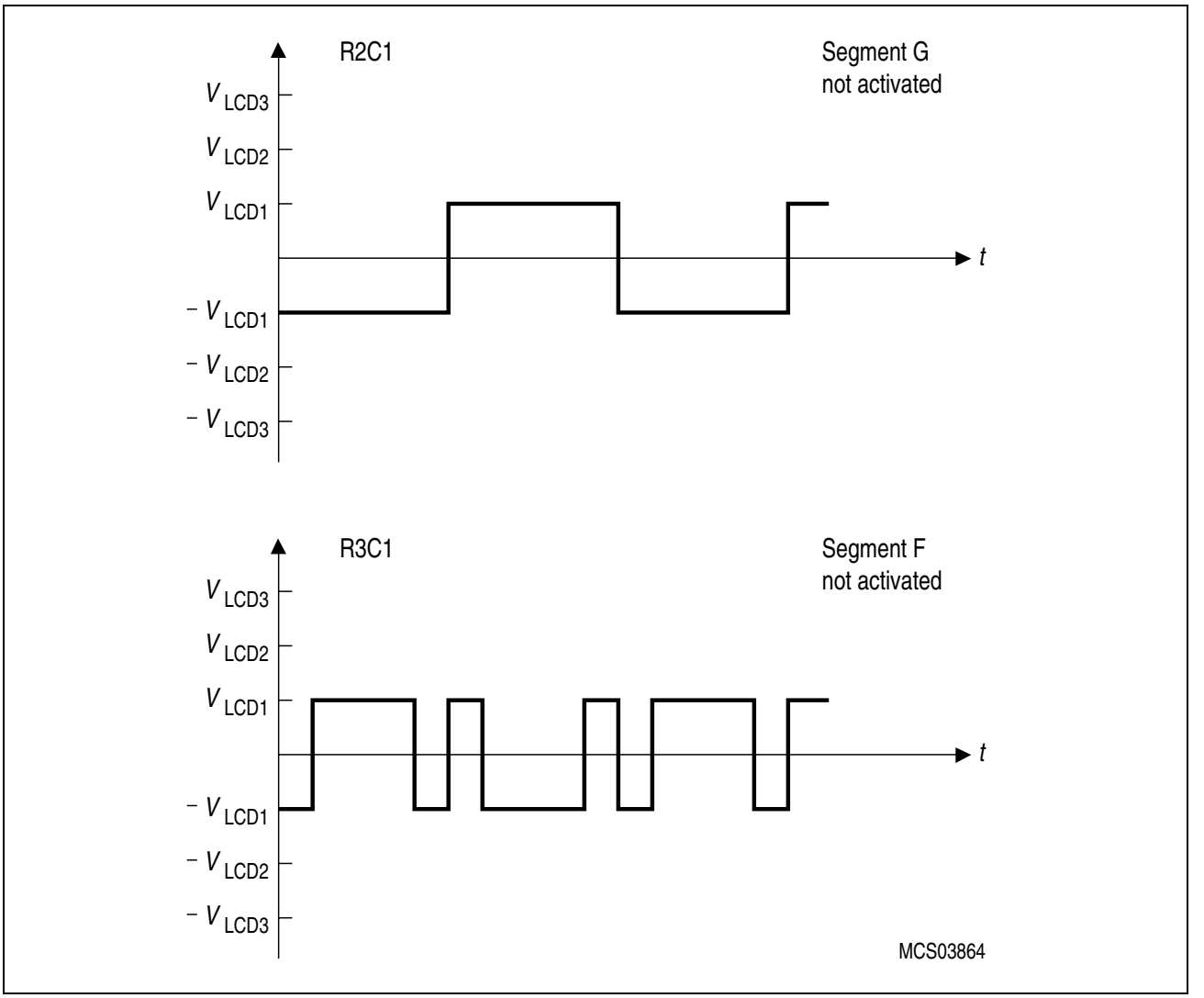


Figure 6-38
Difference Signals to display ‘1’ - Part 3

6.4.6 Voltage Levels

The LCD controller outputs three voltage levels required for driving the LCD display module, namely one-third of V_{LCD}, two-thirds of V_{LCD}, and V_{LCD}. Of these voltage levels, V_{LCD} is generated by the 8-bit D/A converter. The other two voltage levels are derived from V_{LCD} through a resistive divider network.

6.4.6.1 D/A Converter (Reference Voltage Generator)

The D/A converter is enabled by the LCD controller enable bit LCEN (LCON.0). It contains an 8-bit register, DAC0, mapped to the on-chip XRAM area at address $F3DC_H$.

D/A Conversion register, DAC0 (Address $F3DC_H$) **Reset Value: 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$F3DC_H$	S7	S6	S5	S4	S3	S2	S1	S0	DAC0

Any write operation to this register with the LCD controller enabled, starts the D/A conversion and thereby the display outputs. Therefore, the C505L can be used with a wide range of LCD display modules.
It is important to note that the output of D/A converter is, in addition to the DAC0 register value, also affected by the operating voltage range of the device (refer to Data Sheet). Due to the settling time required by the D/A converter, it is recommended to start a D/A conversion prior to writing to the display registers.

6.4.7 Power Saving Mode Options

In order to reduce power consumption, the C505L can be put into the software power down mode 2. In this mode, the LCD controller and the D/A converter do not lose their register contents and remain in operation, provided the following conditions are satisfied:

- The input clock to the LCD is the 32.768-KHz real-time clock input, and
- The real-time clock input at XTAL3 and XTAL4 pins is still valid.

Please refer to **Chapter 9** for further details about the power-down modes.

6.5 Real-Time Clock

One of the C505L's peripherals is the real-time clock that, once started, can work independently of the state of the rest of the microcontroller. This real-time clock contains a 32.768-kHz oscillator to count time elapsed with respect to an initial time. The C505L real-time clock does not provide for any error correction. Any such corrections can be done by software only.

6.5.1 Oscillator

The real-time clock contains an oscillator which can receive an input from an external 32.768 kHz crystal connected to the XTAL3 and XTAL4 pins. Once started, the oscillator can operate irrespective of the state of the microcontroller. That is, it keeps running even when the device has entered idle, slow-down or certain power-down modes. However, the real-time clock can also be powered down by software if necessary.

The oscillator, as well as the whole real-time clock, remains in operation during certain power-down modes with a supply of **3 V** (except at start-up when a 5 V range supply is required).

6.5.2 Real-Time Clock Registers

The register memory for the real-time clock is implemented in the on-chip external data memory area. Accesses to these registers are similar to on-chip XRAM accesses (MOVX instructions) and therefore must be preceded by an enable operation on the on-chip XRAM. The registers are described below:

Real-Time Clock Control Register, RTCON (Address $F3F0_H$) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$F3F0_H$	0	0	0	0	RTPD	IRTC	ERTC	RTCS	RTCON

Bit	Function
RTPD	Real-Time clock Power-Down enable RTPD = 0: Real-time clock is enabled RTPD = 1: Real-time clock is powered down Real-time Clock is enabled by default after reset.
IRTC	Real-Time Clock Interrupt request flag If ERTC bit is set, this bit is set by hardware when the contents of the CLREG and RTINT registers are equal. This bit has to be cleared by software. A wake-up request is generated only if the C505L is either in software power-down modes 2 or 3.
ERTC	Real-Time Clock interrupt Enable ERTC = 0: Disable real-time clock interrupt ERTC = 1: Enable real-time clock interrupt
RTCS	Real-Time Clock Start/Stop bit RTCS = 1: Start Real-Time Clock operation RTCS = 0: Stop Real-Time Clock operation
0	These bits are reserved and should be always written with 0. Writing a 1 into these bits will give undefined results.

Real-Time Clock Initialization Register, RTCRx (x = 0 to 4) **Reset Values: 00_H**

Bit No.	7	6	5	4	3	2	1	0	
$F3F5_H$	MSB	–	–	–	–	–	–	–	RTCR4
$F3F4_H$	–	–	–	–	–	–	–	–	RTCR3
$F3F3_H$	–	–	–	–	–	–	–	–	RTCR2
$F3F2_H$	–	–	–	–	–	–	–	–	RTCR1
$F3F1_H$	–	–	–	–	–	–	–	LSB	RTCR0

Registers RTCR4-RTCR0 form the initial value of the upper 40-bits of the real-time clock counter. These bits are collectively referred to as the RTCR register. The contents of the RTCR register are transferred to the CLREG register under either of the following conditions:

- Once bit RTCS (RTCON.0)is set to start the real-time clock counting, or
- After every subsequent overflow of the real-time clock counter

Therefore, any write operation on this register while the clock is still running does not affect the clock count.

Clock Count Register, CLREGx (x = 0 to 4) **Reset Values: 00_H**

Bit No.	7	6	5	4	3	2	1	0	
$F3FA_H$	MSB	–	–	–	–	–	–	–	CLREG4
$F3F9_H$	–	–	–	–	–	–	–	–	CLREG3
$F3F8_H$	–	–	–	–	–	–	–	–	CLREG2
$F3F7_H$	–	–	–	–	–	–	–	–	CLREG1
$F3F6_H$	–	–	–	–	–	–	–	LSB	CLREG0

Registers CLREG4-CLREG0, collectively known as the CLREG register, hold the current upper 40-bit value of the real-time clock count. The CLREG register can be read at anytime, just like on-chip XRAM locations. The CLREG register can, however, never be written by the user directly. This is to protect the clock count from being manipulated accidentally by any user software.

The CLREG register covers the upper 40-bits of the real-time clock counter. The lower 7 bits of the counter are never accessible by the user and merely act as prescalers that are initialized to 0000000_B after a start operation on the real-time clock.

6.5.3 Functionality

The real-time clock initialization register RTCR, 40 bits, can be written with any value desired by the user. The value written is used as an initial value for the upper 40 bits of the real-time clock timer, when it is started.

The real-time clock is started by setting bit RTCS in the RTCON register to ‘1’. This enables the input clock into the 47-bit timer. The contents of the 40-bit RTCR register are transferred to the CLREG register. The real-time clock’s lower 7 bits, which serve as a prescaler into the 40-bit timer, are set to an initial value of 0000000_B. One increment of the clock register is then made for every cycle of the input clock. With an input clock frequency of 32.768 kHz, one second in real-time will be equivalent to an overflow of the lower 15-bits of the 47-bit counter. Under this condition, the registers CLREG4-CLREG1 actually hold the real-time value in “seconds”.

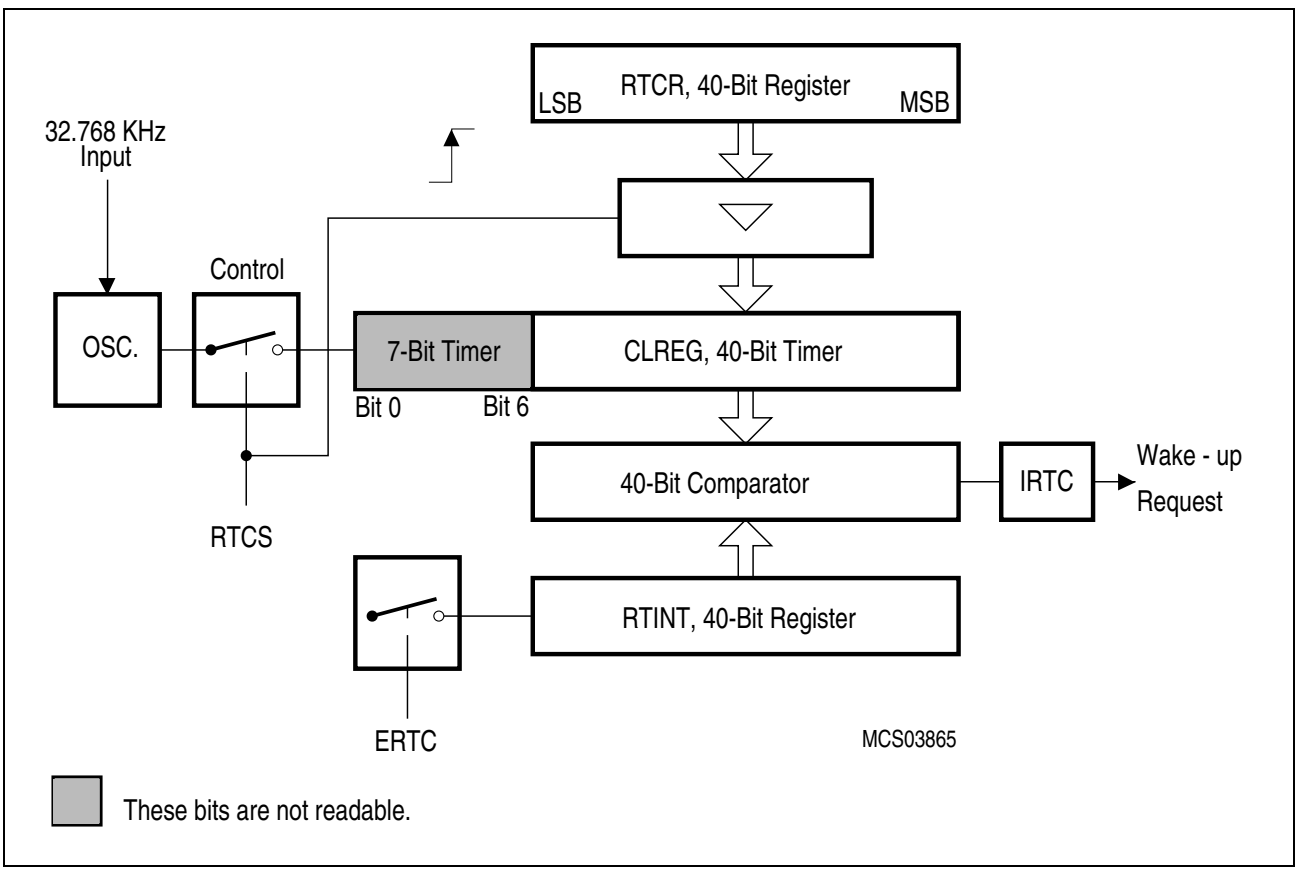


Figure 6-39
Real-Time Clock

Upon an overflow condition of the real-time clock timer, the contents of the RTCR register are reloaded into the CLREG for a fresh count sequence.

The real-time clock stops counting when the RTCS bit is written with a ‘0’. Setting the RTCS bit subsequently does not resume the count because a new counting sequence is started.

When the real-time clock is in operation, the upper 40-bits of the CLREG register can be read on-the-fly just like external data memory.

6.5.4 Real-Time Clock Wake-up Interrupt

The upper 40-bit content of the real-time clock counter can be compared with the content of the RTINT register in order to generate an interrupt request while the C505L is in one of software power-down modes 2 or 3. The 40-bit RTINT register consists of 5 individual 8-bit registers as described below.

Real-Time Clock Interrupt Register, RTINTx (x = 0 to 4) **Reset Value: 00_H**

Bit No.	7	6	5	4	3	2	1	0	
$F3FF_H$	MSB	–	–	–	–	–	–	–	RTINT4
$F3FE_H$	–	–	–	–	–	–	–	–	RTINT3
$F3FD_H$	–	–	–	–	–	–	–	–	RTINT2
$F3FC_H$	–	–	–	–	–	–	–	–	RTINT1
$F3FB_H$	–	–	–	–	–	–	–	LSB	RTINT0

The RTINT register is programmable by the user, and any 40-bit value can be programmed into it. While the real-time clock is in operation, the contents of this register will be compared to the upper 40-bits of the clock counter (CLREG). Bit IRTC (RTCON.2) will be set when the contents are equal; this will generate a wake-up from software power-down interrupt. The IRTC flag can be monitored for the real-time clock interrupt wake-up request, but the flag has to be cleared by user software.

The real-time clock can generate a wake-up request to the C505L provided all the following conditions are fulfilled:

- The C505L is in one of the software power-down modes 2 or 3,
- Wake-up from software power-down is enabled (bit EWPD = 1 in SFR PCON1)
- Real-time clock wake-up source is selected (bit WS = 1 in SFR PCON1),
- The real-time clock interrupt is enabled (bit ERTC = 1 of RTCON), and
- Normally operating V_{DD} levels are maintained

In this case, the handling is similar to the wake-up from power-down through P3.2/$\overline{INT0}$ (please see **Chapter 9** for further details).

6.5.5 Power-saving Mode Options

Once started, the real-time clock continues counting until the bit RTCS (RTCR0.0) is cleared. The real-time clock is not affected by any of the idle and slow-down modes of the C505L, and continues counting in these modes. In the software power-down mode 3 the real-time clock continues to run provided:

- The externally applied input clock for the real-time clock is available, and
- $V_{DD\ (min)} \geq 3.0$ Volts

In software power-down mode 2, however, the supply voltage should remain at normal levels as specified in the Device Specifications (refer to Data Sheet).

The real-time clock stops operation in software power-down mode 1 (bit RTPD in RTCON.3 is set). Please refer to **Chapter 9** for detailed information regarding the power-down modes.

6.6 A/D Converter

The C505L includes a high performance, high-speed, 10-bit A/D Converter (ADC) with 8 analog input channels. It operates with a successive approximation technique, and uses self calibration mechanisms for reduction and compensation of offset and linearity errors. The ADC has the following features:

- 8 multiplexed input channels (port 1) that can also be used as digital inputs/outputs
- 10-bit resolution
- Single or continuous conversion mode
- Internal start-of-conversion trigger capability
- Interrupt request generation after each conversion
- Using successive approximation conversion technique via a capacitor array
- Built-in hidden calibration of offset and linearity errors

The externally-applied reference voltages have to be held at fixed values within the specifications (please see Data Sheet). The main functional blocks of the ADC are shown in **Figure 6-40**.

6.6.1 A/D Converter Operation

An internal start of a single A/D conversion is triggered by a write-to-ADDATL instruction. The start procedure itself is independent of the value that is written to ADDATL. When single conversion mode is selected (bit ADM = 0), only one A/D conversion is performed. In continuous mode (bit ADM = 1), after completion of an A/D conversion, a new A/D conversion is triggered automatically until bit ADM is reset.

The busy flag BSY (ADCON0.4) is set automatically when an A/D conversion is in progress. After completion of the conversion it is reset by hardware. This flag can only be read; a write has no effect. The interrupt request flag IADC (IRCON.0) is set when an A/D conversion is completed.

The bits MX0 to MX2 in Special Function Registers (SFRs) ADCON0 and ADCON1 are used for selection of the analog input channel. The bits MX0 to MX2 are represented in both registers ADCON0 and ADCON1; however, these bits are present only once. Therefore, there are two methods of selecting an analog input channel: If a new channel is selected in ADCON1 the change is automatically done in the corresponding bits MX0 to MX2 in ADCON0, and vice versa.

Port 1 is a dual purpose input/output port. These pins can be used either for digital I/O functions or as the analog inputs. If fewer than 8 analog inputs are required, the unused analog inputs at port 1 are free for digital I/O functions.

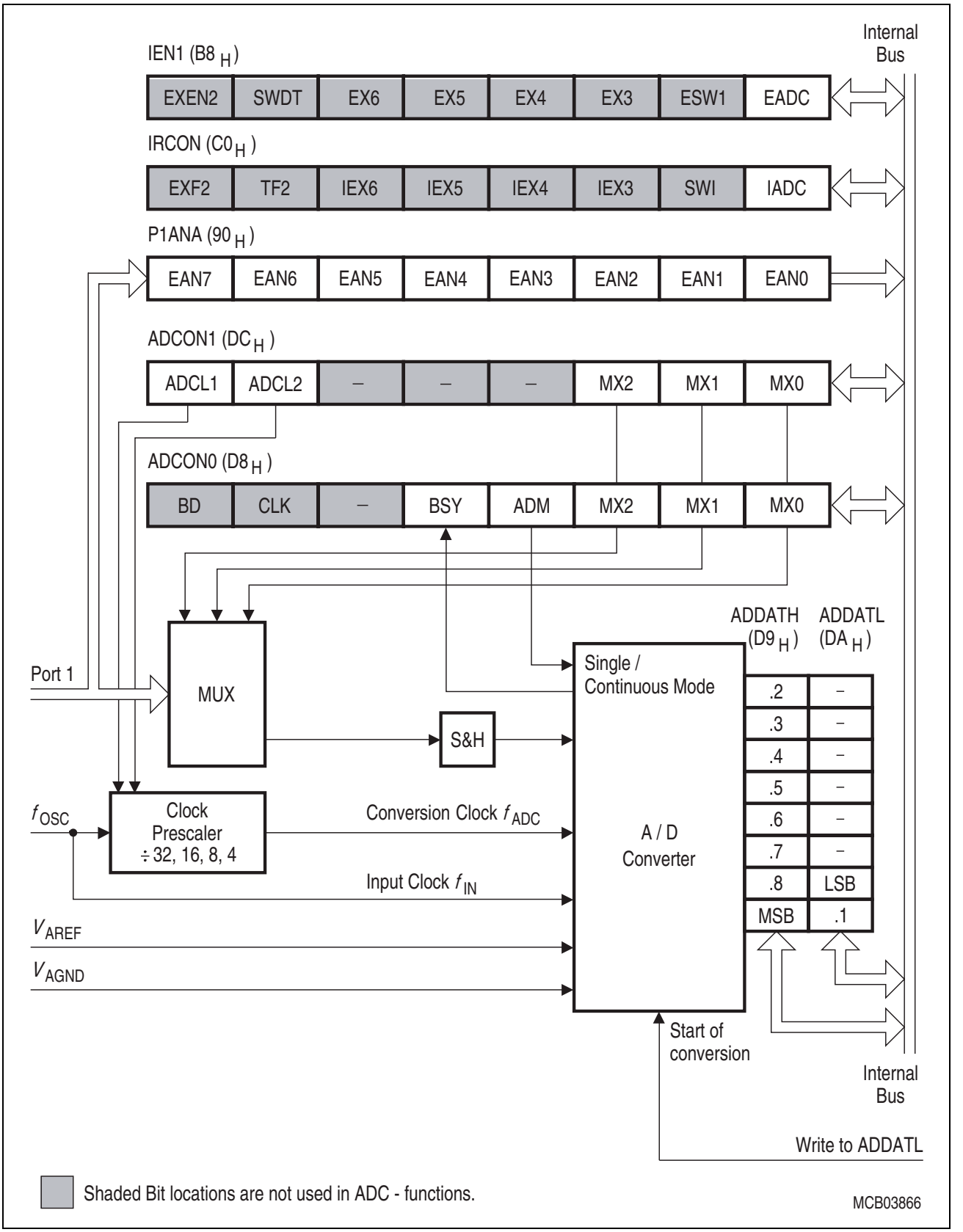


Figure 6-40
Block Diagram of the A/D Converter

6.6.2 A/D Converter Registers

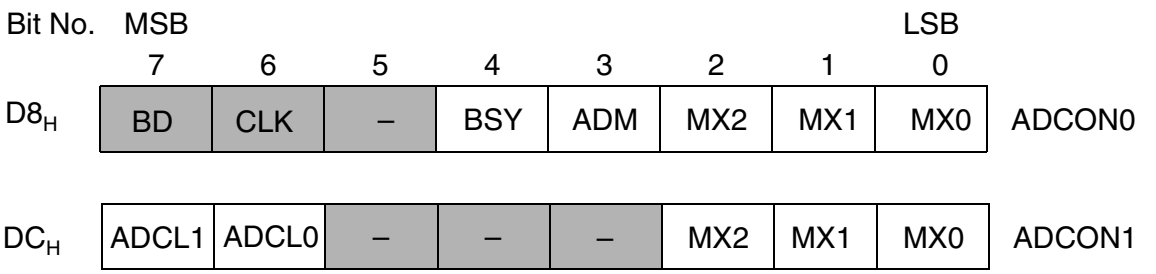
This section describes the bits/functions of all registers which are used by the ADC.

Special Function Register ADDATH (Address $D9_H$) **Reset Value: 00_H**
Special Function Register ADDATL (Address DA_H) **Reset Value: $00XXXXXX_B$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$D9_H$	MSB .9	.8	.7	.6	.5	.4	.3	.2	ADDATH
DA_H	.1	LSB .0	–	–	–	–	–	–	ADDATL

The registers ADDATH and ADDATL hold the 10-bit conversion result in left justified data format. The MSB of the 10-bit conversion result is bit 7 of ADDATH. The LSB of the 10-bit conversion result is bit 6 of ADDATL. To get a 10-bit conversion result, both ADDAT registers must be read. If an 8-bit conversion result is required, only the reading of ADDATH is necessary. The data remains in ADDAT until it is overwritten by the next converted data. ADDAT can be read or written under software control. If the ADC of the C505L is not used, register ADDATH can be used as an additional general purpose register.

Special Function Register ADCON0 (Address $D8_H$) **Reset Value: $00X00000_B$**
Special Function Register ADCON1 (Address DC_H) **Reset Value: $01XXX000_B$**



The shaded bits are not used for ADC control.

Bit	Function
–	Reserved bits for future use
BSY	Busy flag This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is finished.
ADM	A/D conversion Mode When set, a continuous A/D conversion is selected. If cleared during a running A/D conversion, the conversion is stopped at its end.
MX2 - MX0	ADC input channel select bits Bits MX2-0 can be written or read either in ADCON0 or ADCON1. The channel selection done by writing to ADCON 1(0) overwrites the selection in ADCON 0(1) when ADCON 1(0) is written after ADCON 0(1). The analog inputs are selected according to the following table:

MX2	MX1	MX0	Selected Analog Input
0	0	0	P1.0 / AN0 / $\overline{\text{INT3}}$ / CC0
0	0	1	P1.1 / AN1 / INT4 / CC1
0	1	0	P1.2 / AN2 / INT5 / CC2
0	1	1	P1.3 / AN3 / INT6 / CC3
1	0	0	P1.4 / AN4
1	0	1	P1.5 / AN5 / T2EX
1	1	0	P1.6 / AN6 / CLKOUT
1	1	1	P1.7 / AN7 / T2

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>ADCL1
ADCL0</td><td>ADC clock prescaler selection
ADCL1 and ADCL0 select the prescaler ratio for the A/D conversion clock f_{ADC}. Depending on the clock rate f_{OSC} of the C505L, f_{ADC} must be adjusted in a way that the resulting conversion clock f_{ADC} is less than or equal to 2 MHz (see Section 6.6.3).
The prescaler ratio is selected according to the following table:
<table>
<tr><th>ADCL1</th><th>ADCL0</th><th>Prescaler Ratio</th></tr>
<tr><td>0</td><td>0</td><td>divide by 4</td></tr>
<tr><td>0</td><td>1</td><td>divide by 8 (default after reset)</td></tr>
<tr><td>1</td><td>0</td><td>divide by 16</td></tr>
<tr><td>1</td><td>1</td><td>divide by 32</td></tr>
</table>
</td></tr>
</table>

Note: Generally, before entering the power-down mode, an A/D conversion in progress must be stopped. If a single A/D conversion is running, it must be terminated by polling the BSY bit or waiting for the A/D conversion interrupt. In continuous conversion mode, bit ADM must be cleared and the last A/D conversion must be terminated before entering the power-down mode.

A single A/D conversion is started by writing to SFR ADDATL with dummy data. A continuous conversion is started under the following conditions:

- By setting bit ADM during a running single A/D conversion
- By setting bit ADM when at least one A/D conversion has occurred after the last reset operation.
- By writing ADDATL with dummy data after bit ADM has been set before (if no A/D conversion has occurred after the last reset operation).

When bit ADM is reset by software in continuous conversion mode, the just running A/D conversion is stopped after its end.

The ADC interrupt is controlled by bits which are located in the SFRs IEN1 and IRCON.

Special Function Register IEN1 (Address $B8_H$) **Reset Value: 00_H**
Special Function Register IRCON (Address $C0_H$) **Reset Value: 00_H**

Bit No.	MSB							LSB	
	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	ESWI	EADC	IEN1
	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	SWI	IADC	IRCON

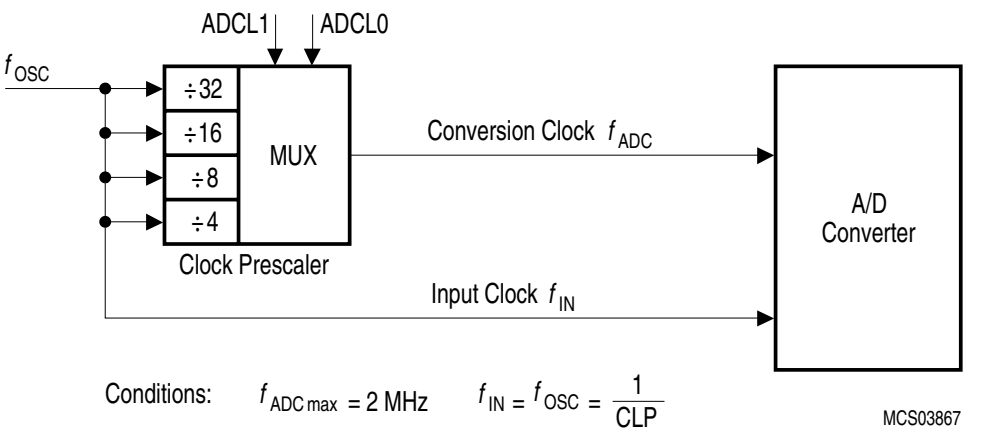
The shaded bits are not used for ADC control.

Bit	Function
EADC	Enable ADC interrupt If EADC = 0, the ADC interrupt is disabled.
IADC	ADC Interrupt request flag Set by hardware at the end of an A/D conversion. Must be cleared by software.

6.6.3 A/D Converter Clock Selection

The ADC uses two clock signals for operation: The conversion clock f_{ADC} (= $1/t_{ADC}$) and the input clock f_{IN} (= $1/t_{IN}$). f_{ADC} is derived from the C505L system clock f_{OSC} which is applied at the XTAL pins via the ADC clock prescaler as shown in **Figure 6-41**. The input clock f_{IN} is equal to f_{OSC}. The conversion clock f_{ADC} is limited to a maximum frequency of 2 MHz. Therefore, the ADC clock prescaler must be programmed to a value that assures that the conversion clock does not exceed 2 MHz. The prescaler ratio is selected by the bits ADCL1 and ADCL0 of SFR ADCON1.

The table in **Figure 6-41** shows the prescaler ratio that must be selected by ADCL1 and ADCL0 for typical system clock rates. A prescaler ratio equal to 4 is selected for system clocks up to 8 MHz. A prescaler ratio of at least 8 must be selected when using a system clock greater than 8 MHz and less than 16 MHz. A prescaler ratio of at least 16 must be selected when using a system clock greater than 16 MHz. A prescaler ratio of 32 can used for any of the above frequency ranges.



MCU System Clock Rate (f_{OSC})	f_{IN} [MHz]	Prescaler Ratio	f_{ADC} [MHz]	ADCL1	ADCL0
2 MHz	2	÷ 4	0.5	0	0
6 MHz	6	÷ 4	1.5	0	0
8 MHz	8	÷ 4	2	0	0
12 MHz	12	÷ 8	1.5	0	1
16 MHz	16	÷ 8	2	0	1
20 MHz	20	÷ 16	1.25	1	0

Figure 6-41
ADC Clock Selection

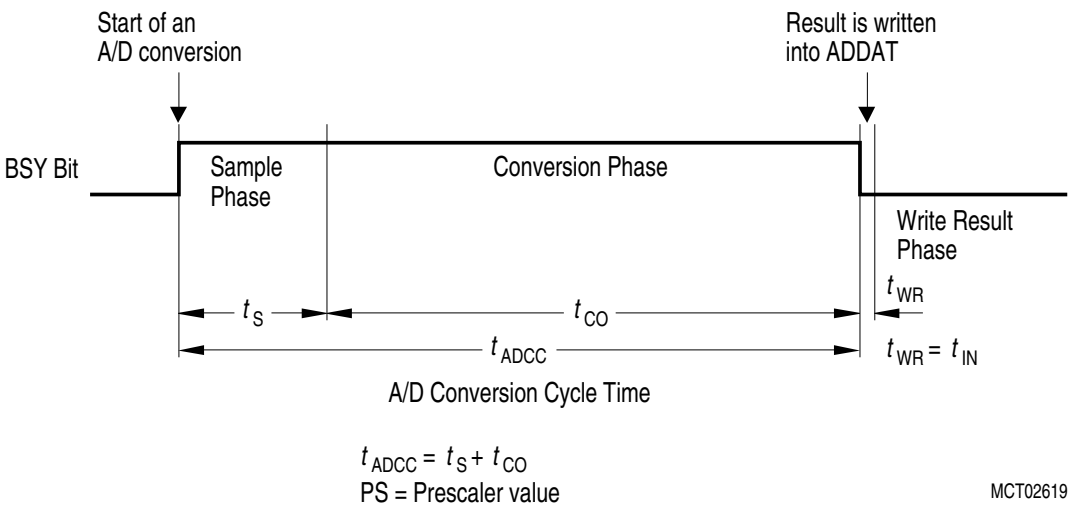
The duration of an A/D conversion is a multiple of the period of the f_{IN} clock signal. The calculation of the A/D conversion time is shown in the next section.

6.6.4 A/D Conversion Timing

An A/D conversion is started by writing into the SFR ADDATL with dummy data. A write to SFR ADDATL will start a new conversion, even if another conversion is currently in progress. The conversion begins with the next machine cycle, and the BSY flag in SFR ADCON0 will be set.
The A/D conversion procedure is divided into three parts:

- Sample phase (t_S), used for sampling the analog input voltage.
- Conversion phase (t_{CO}), used for the real A/D conversion (includes calibration).
- Write result phase (t_{WR}), used for writing the conversion result into the ADDAT registers.

The total A/D conversion time is defined by t_{ADCC}, which is the sum of the two phase times, t_S and t_{CO}. The duration of the phases of an A/D conversion is specified by their corresponding timing parameters as shown in **Figure 6-42**.



Prescaler Ratio (= PS)	t_S	t_{CO}	t_{ADCC}
32	$64 \times t_{IN}$	$320 \times t_{IN}$	$384 \times t_{IN}$
16	$32 \times t_{IN}$	$160 \times t_{IN}$	$192 \times t_{IN}$
8	$16 \times t_{IN}$	$80 \times t_{IN}$	$96 \times t_{IN}$
4	$8 \times t_{IN}$	$40 \times t_{IN}$	$48 \times t_{IN}$

Figure 6-42
A/D Conversion Timing

Sample Time t_S:
During this time, the internal capacitor array is connected to the selected analog input channel and is loaded with the analog voltage to be converted. The analog voltage is fed internally to a voltage comparator. At the beginning of the sample phase, the BSY bit in SFR ADCON0 is set.

Conversion Time t_{CO}:
During the conversion time, the analog voltage is converted into a 10-bit digital value using the

successive approximation technique with a binary-weighted capacitor network. During an A/D conversion, a calibration also takes place. During this calibration, alternating offset and linearity calibration cycles are executed (see also **Section 6.6.5**). At the end of the calibration time, the BSY bit is reset and the IADC bit in SFR IRCON is set indicating an ADC interrupt condition.

Write Result Time t_{WR}:
During the result phase, the conversion result is written into the ADDAT registers.

Figure 6-43 shows how an A/D conversion is embedded into the microcontroller cycle scheme using the relation $6 \times t_{IN}$ = 1 instruction cycle. It also shows the behavior of the busy flag (BSY) and the interrupt flag (IADC) during an A/D conversion.

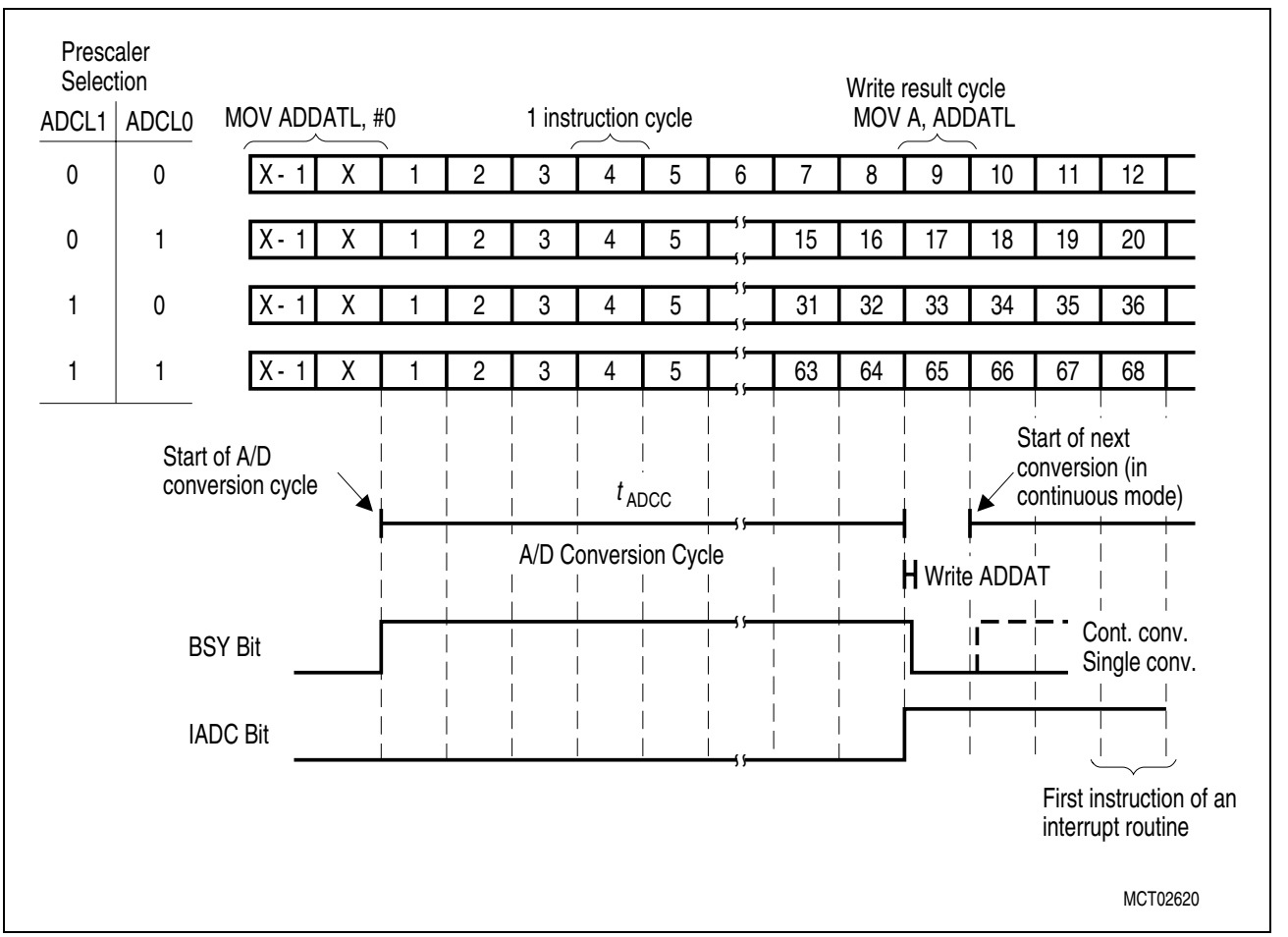


Figure 6-43
A/D Conversion Timing in Relation to Processor Cycles

Depending on the selected prescaler ratio (see **Figure 6-41**), four different relationships between machine cycles and A/D conversion are possible. The A/D conversion is started when SFR ADDATL is written with dummy data. This write operation may take one or two machine cycles. In **Figure 6-43**, the instruction MOV ADDATL,#0 starts the A/D conversion (machine cycle X-1 and X). The total A/D conversion (sample, conversion, and calibration phases) is finished with the end of the 8th, 16th, 32nd, or 64th machine cycle after the A/D conversion start. In the next machine cycle, the conversion result is written into the ADDAT registers, and can be read in the same cycle by an instruction (e.g. MOV A,ADDATL). If continuous conversion is selected (bit ADM set), the next conversion is started with the beginning of the machine cycle that follows the write result cycle.

The BSY bit is set at the beginning of the first A/D conversion machine cycle, and reset at the beginning of the write result cycle. If continuous conversion is selected, BSY is again set with the beginning of the machine cycle that follows the write result cycle.

The interrupt flag IADC is set at the end of the A/D conversion. If the ADC interrupt is enabled and prioritized to be serviced immediately, the first instruction of the interrupt service routine will be executed in the third machine cycle which follows the write result cycle. IADC must be reset by software.

Depending on the application, there are three typical methods to handle the A/D conversion in the C505L.

- Software delay
  The machine cycles of the A/D conversion are counted, and the program executes a software delay (e.g. NOPs) before reading the A/D conversion result in the write result cycle. This is the fastest method to get the result of an A/D conversion.
- Polling BSY bit
  The BSY bit is polled, and the program waits until BSY = 0. Note: a polling JB instruction that is two machine cycles long, may not recognize the BSY = 0 condition during the write result cycle in the continuous conversion mode.
- A/D conversion interrupt
  After the start of an A/D conversion, the ADC interrupt is enabled. The result of the A/D conversion is read in the interrupt service routine. If other C505L interrupts are enabled, the interrupt latency must taken into account. Therefore, this software method is the slowest way to get the result of an A/D conversion.

Depending on the oscillator frequency of the C505L and the selected divider ratio of the conversion clock prescaler, the total time of an A/D conversion is calculated according to **Figure 6-42** and **Table 6-6. Figure 6-44** on the next page shows the minimum A/D conversion time in relation to the oscillator frequency f_{OSC}. The minimum conversion time, 6 µs, can be achieved at f_{OSC} of 8 or 16 MHz (or whenever f_{ADC} = 2 MHz).

Table 6-6
A/D Conversion Time for Dedicated System Clock Rates

f_{OSC} [MHz]	Prescaler Ratio PS	f_{ADC} [MHz]	Sample Time t_S [µs]	Total Conversion Time t_{ADCC} [µs]
2 MHz	÷ 4	0.5	4	24
6 MHz	÷ 4	1.5	1.33	8
8 MHz	÷ 4	2	1	6
12 MHz	÷ 8	1.5	1.33	8
16 MHz	÷ 8	2	1	6
20 MHz	÷ 16	1.25	1.6	9.6

*Note: The prescaler ratios in **Table 6-6** are minimum values.*

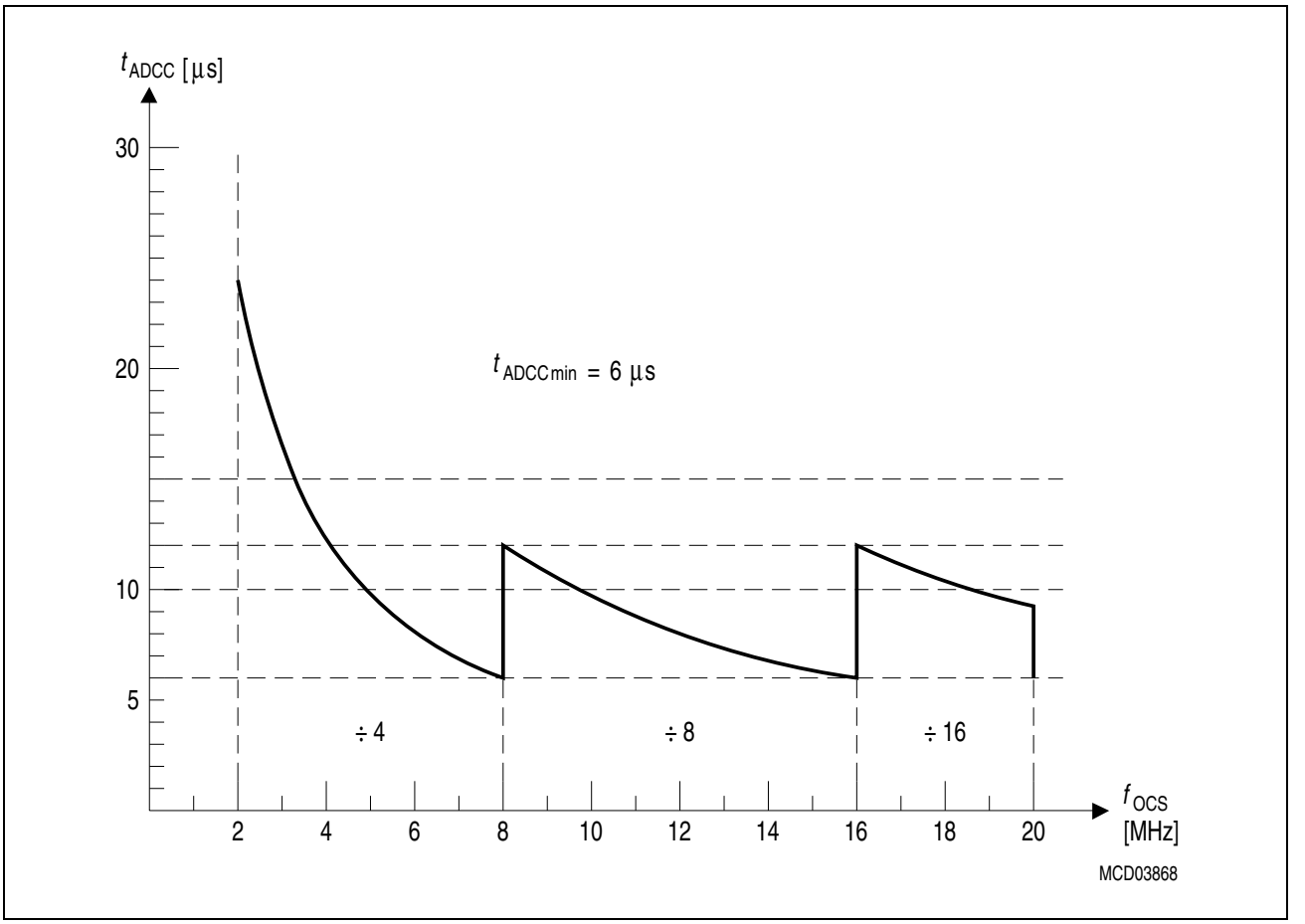


Figure 6-44
Minimum A/D Conversion Time in Relation to System Clock

6.6.5 A/D Converter Calibration

The C505L ADC includes hidden internal calibration mechanisms to ensure that the ADC will function safely according to the DC characteristics. A user program that executes A/D conversions will not affect the ADC's operation. Furthermore, a user program has no control over the calibration mechanism. The calibration itself executes two basic functions:

- Offset calibration: Correction of offset errors of comparator and the capacitor network
- Linearity calibration: Correction of the binary-weighted capacitor network

The ADC calibration operates in two phases: Calibration after a reset operation and calibration at each A/D conversion. The calibration phases are controlled by a state machine in the ADC. This state machine executes the calibration phases, and stores the calibration results dynamically in a small calibration RAM.

After a reset operation, the ADC calibration is automatically started. This reset calibration phase takes 3328 f_{ADC} clocks, alternating offset and linearity calibration. Therefore, at 16-MHz oscillator frequency and with the default-after-reset prescaler value of 8, the reset calibration time will be approximately 1.66 ms. For achieving a proper reset calibration, the f_{ADC} prescaler value must satisfy the condition $f_{ADC\ max} \leq 2$ MHz. If this condition is not met at a specific oscillator frequency with the default prescaler value after reset, the f_{ADC} prescaler must be adjusted immediately after reset by setting bits ADCL1 and ADCL0 in SFR ADCON1 to a suitable value. Proper voltages (see the DC specifications of the Data Sheet) should be applied at the V_{AREF} and V_{AGND} pins before the reset calibration has started.

After the reset calibration phase, the ADC is calibrated according to its DC characteristics (refer to Data Sheet). Nevertheless, single or continuous A/D conversion can be executed during the reset calibration phase. In this case, the reset calibration is interrupted and continued after the end of the A/D conversion. Therefore, interrupting the reset calibration phase by A/D conversions extends the total reset calibration time. If the specified Total Unadjusted Error (TUE) is to be valid for an A/D conversion, the first A/D conversion should start after reset when the reset calibration phase is finished. Depending on the oscillator frequency used, the reset calibration phase can be shortened by setting ADCL1 and ADCL0 (prescaler value) to its final value immediately after reset.

After the reset calibration, a second calibration mechanism is initiated. This calibration is coupled to each A/D conversion**.** With this second calibration mechanism, offset and linearity calibration values stored in the calibration RAM, are always checked when an A/D conversion is executed. These values are corrected, if required.

6.6.6 A/D Converter Analog Input Selection

The analog inputs are located at port 1. The corresponding pins have a port structure, which allows them to be used as either digital I/O pins or as analog inputs (see **Section 6.1.3.2**). The analog input function of these digital/analog port lines are selected via the register P1ANA. This register lies in the mapped SFR area, and can be accessed if bit RMAP in SFR SYSCON is set when writing to its address (90_H). If a specific bit location of P1ANA is set, the corresponding port line is configured as a digital I/O. With a 0 in the bit location, the port line operates as an analog port.

Special Function Registers P1ANA (Address 90_H) **Reset Value: FF_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
90_H	EAN7	EAN6	EAN5	EAN4	EAN3	EAN2	EAN1	EAN0	P1ANA

Bit	Function
EAN7 - EAN0	Enable ANalog port 1 inputs If EANx (x = 7-0) is cleared, port pin P1.x is enabled for operation as an analog input. If EANx is set, port pin P1.x is enabled for digital I/O function (default after reset).

7 Interrupt System

The C505L provides 12 interrupt vectors with four priority levels. Five interrupt requests can be generated by the on-chip peripherals (timer 0, timer 1, timer 2, serial interface, A/D converter) and six interrupts may be triggered externally (P3.2/$\overline{\text{INT0}}$, P3.3/$\overline{\text{INT1}}$, P1.0/AN0/$\overline{\text{INT3}}$/CC0, P1.1/AN1/INT4/CC1, P1.2/AN2/INT5/CC2, P1.3/AN3/INT6/CC3). Additionally, the P1.5/AN5/T2EX can trigger an interrupt. There is one software-generated interrupt (bit SWI in SFR IEN1) in addition to the above interrupts. The wake-up from power-down mode interrupt has a special functionality which allows an exit from the software power-down mode by a short low pulse at either pin P3.2/$\overline{\text{INT0}}$ or by the real-time clock interrupt (please refer to **Chapter 9** for further details).

The four external interrupts ($\overline{\text{INT3}}$, INT4, INT5 and INT6) can also be generated by the timer 2 in capture/compare mode.

This chapter shows the interrupt structure, the interrupt vectors and the interrupt related Special Function Registers (SFRs). **Figure 7-1** to **7-3** give a general overview of the interrupt sources and illustrate the request and the control flags that are described in the next sections.

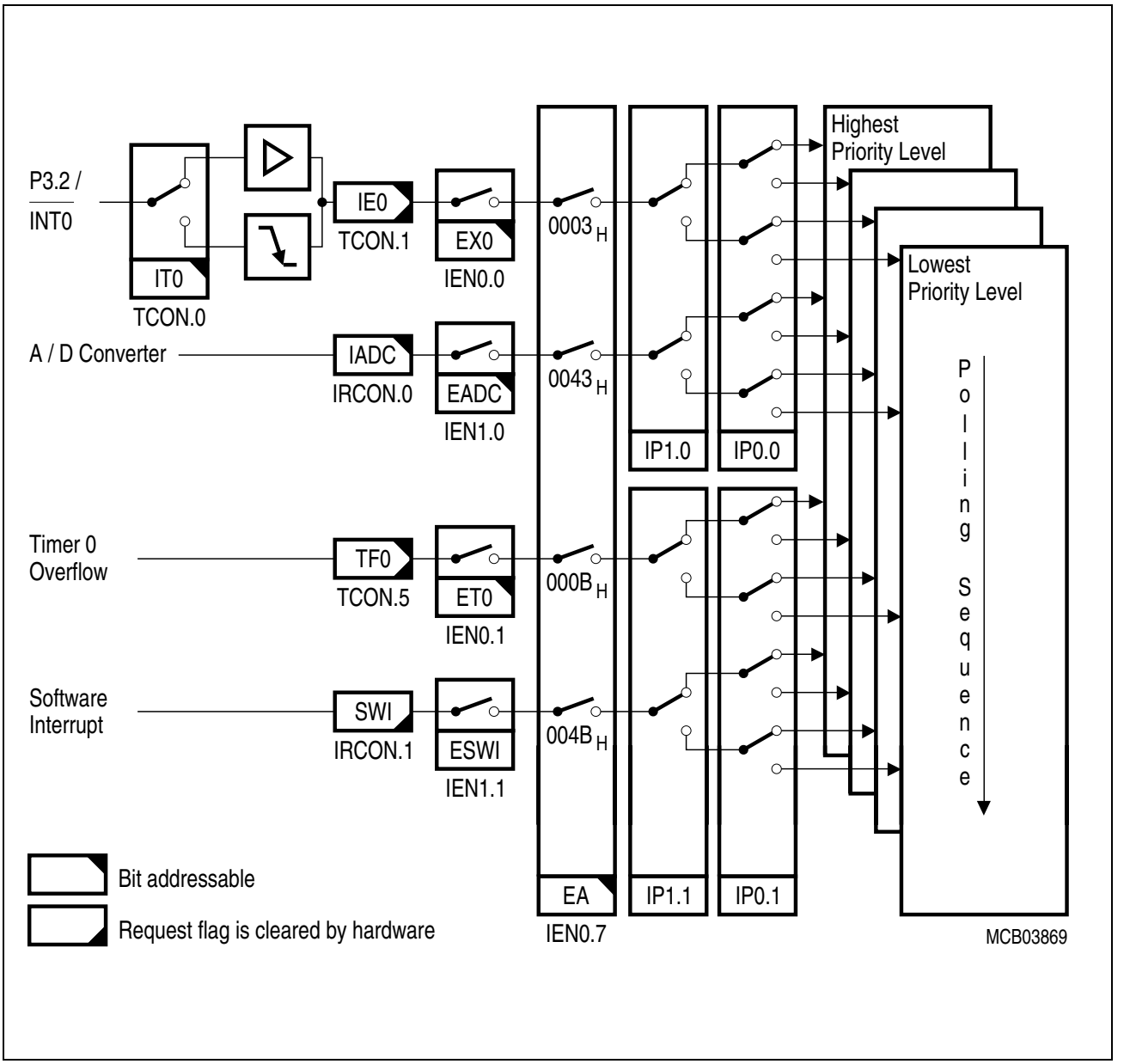


Figure 7-1
Interrupt Structure, Overview Part 1

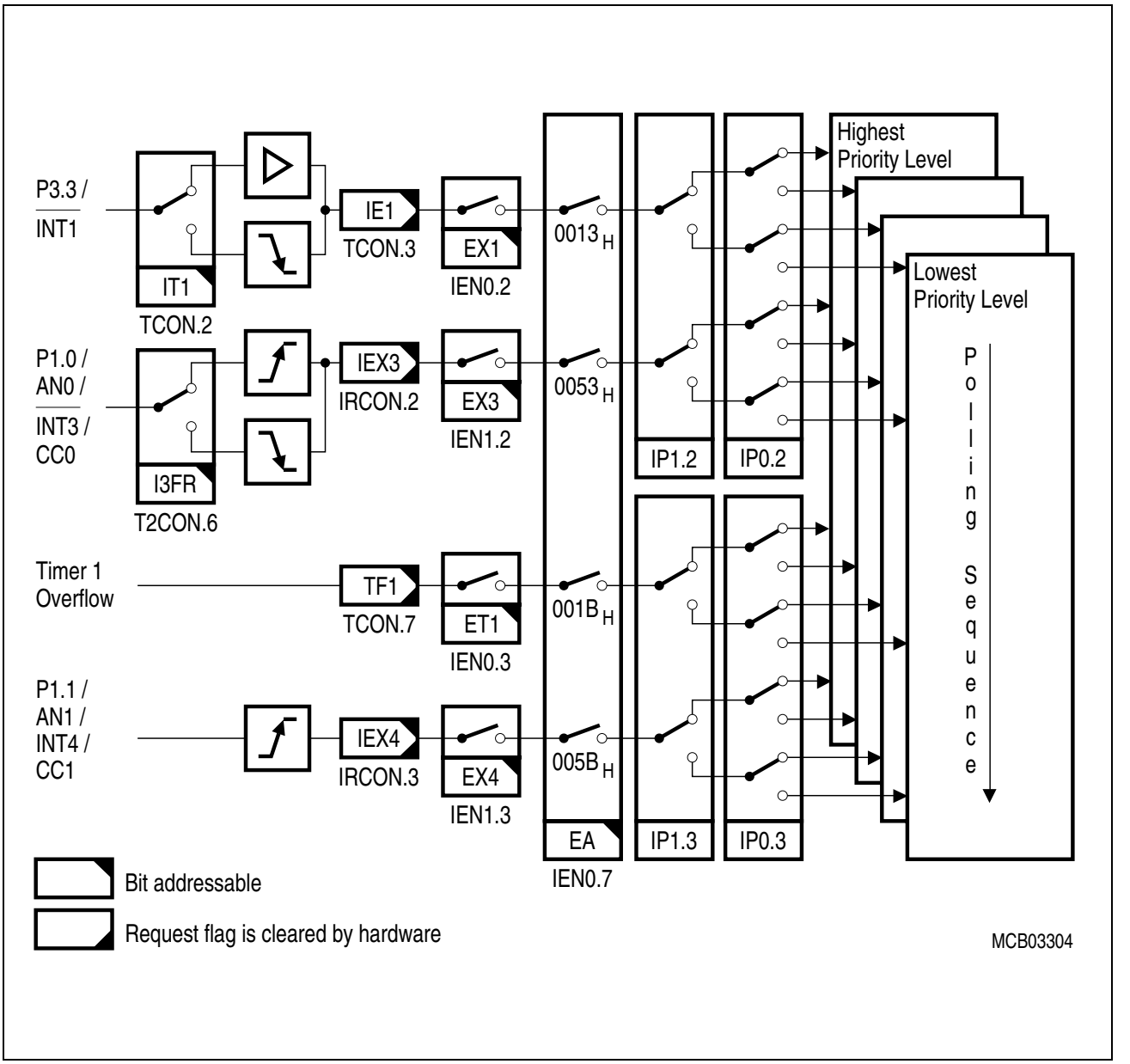


Figure 7-2
Interrupt Structure, Overview Part 2

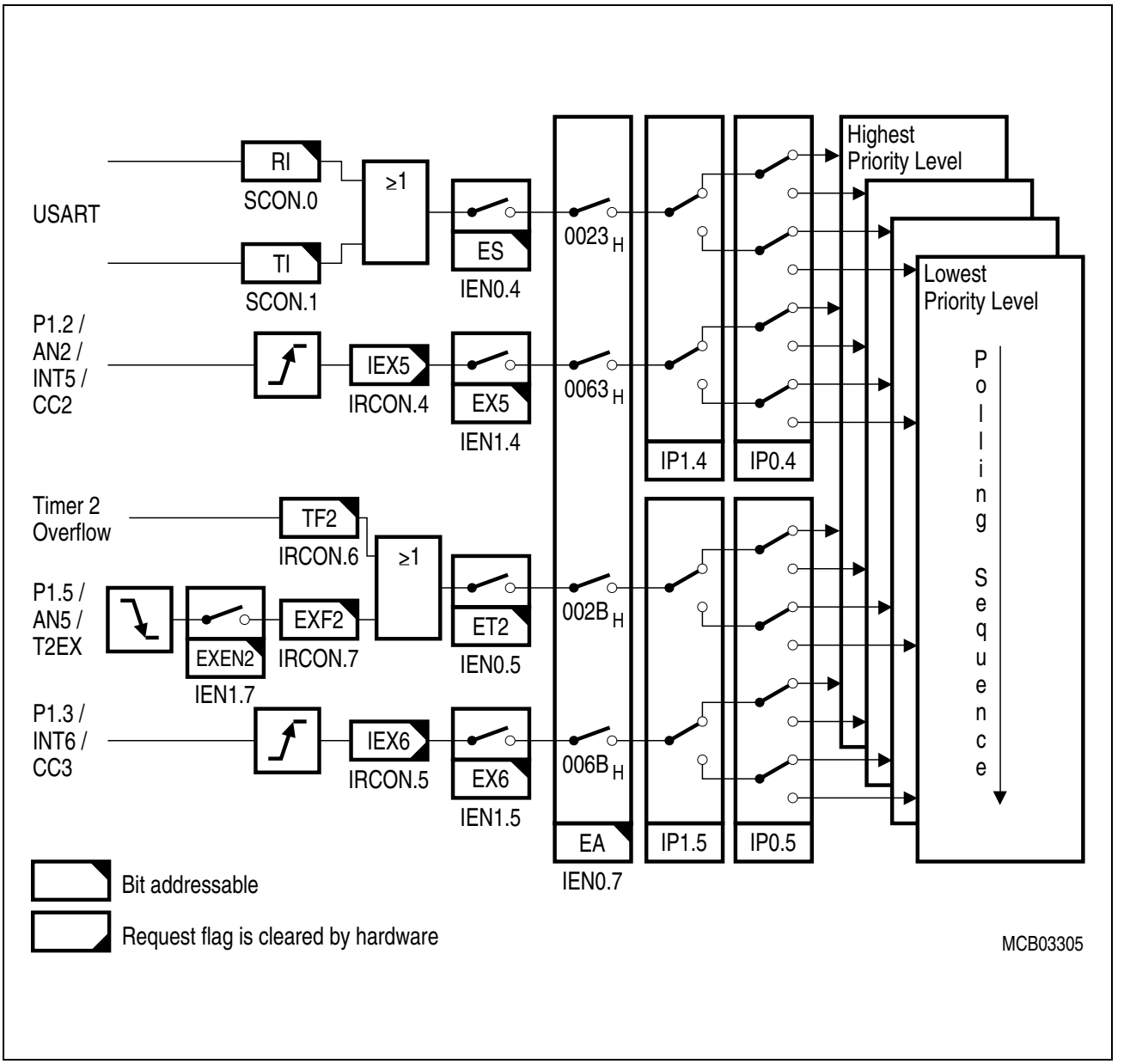


Figure 7-3
Interrupt Structure, Overview Part 3

7.1 Interrupt Registers

7.1.1 Interrupt Enable Registers

Each interrupt vector can be individually enabled or disabled by setting or clearing the corresponding bit in the interrupt enable registers IEN0 and IEN1. Register IEN0 also contains the global disable bit (EA), which can be cleared to disable all interrupts at once. Generally, after reset all interrupt enable bits are set to 0. That means that the corresponding interrupts are disabled.

The IEN0 register contains the general enable/disable flags of the external interrupts 0 and 1, the timer interrupts, and the USART interrupt.

Special Function Register IEN0 (Address $A8_H$) **Reset Value: 00_H**

	MSB							LSB	
Bit No.	AF_H	AE_H	AD_H	AC_H	AB_H	AA_H	$A9_H$	$A8_H$	
$A8_H$	EA	WDT	ET2	ES	ET1	EX1	ET0	EX0	IEN0

The shaded bits are not used for interrupt control.

Bit	Function
EA	Enable/disable All interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
ET2	Timer 2 overflow/external reload interrupt Enable. If ET2 = 0, the timer 2 interrupt is disabled. If ET2 = 1, the timer 2 interrupt is enabled.
ES	Serial channel (USART) interrupt Enable If ES = 0, the serial channel interrupt 0 is disabled. If ES = 1, the serial channel interrupt 0 is enabled.
ET1	Timer 1 overflow interrupt Enable. If ET1 = 0, the timer 1 interrupt is disabled. If ET1 = 1, the timer 1 interrupt is enabled.
EX1	EXternal interrupt 1 Enable. If EX1 = 0, the external interrupt 1 is disabled. If EX1 = 1, the external interrupt 1 is enabled.
ET0	Timer 0 overflow interrupt Enable. If ET0 = 0, the timer 0 interrupt is disabled. If ET0 = 1, the timer 0 interrupt is enabled.
EX0	EXternal interrupt 0 Enable. If EX0 = 0, the external interrupt 0 is disabled. If EX0 = 1, the external interrupt 0 is enabled.

The IEN1 register contains enable/disable flags of the timer 2 external timer reload interrupt, the external interrupts 2 and 3, the software interrupt, and the A/D converter (ADC) interrupt.

Special Function Register IEN1 (Address $B8_H$) **Reset Value: 00_H**

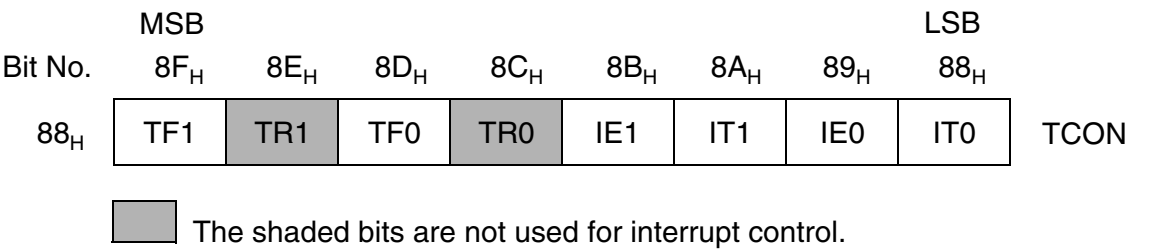
Bit No.	MSB BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	LSB $B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	ESWI	EADC	IEN1

The shaded bits are not used for interrupt control.

Bit	Function
EXEN2	Timer 2 EXternal reload interrupt ENable If EXEN2 = 0, the timer 2 external reload interrupt is disabled. If EXEN2 = 1, the timer 2 external reload interrupt is enabled. The external reload function is not affected by EXEN2.
EX6	EXternal interrupt 6/capture/compare interrupt 3 Enable If EX6 = 0, external interrupt 6 is disabled. If EX6 = 1, external interrupt 6 is enabled.
EX5	EXternal interrupt 5/capture/compare interrupt 2 Enable If EX5 = 0, external interrupt 5 is disabled. If EX5 = 1, external interrupt 5 is enabled.
EX4	EXternal interrupt 4/capture/compare interrupt 1 Enable If EX4 = 0, external interrupt 4 is disabled. If EX4 = 1, external interrupt 4 is enabled.
EX3	EXternal interrupt 3/capture/compare interrupt 0 Enable If EX3 = 0, external interrupt 3 is disabled. If EX3 = 1, external interrupt 3 is enabled.
ESWI	SoftWare Interrupt Enable If ESWI = 0, the software interrupt is disabled. If ESWI = 1, the software interrupt is enabled. This bit must be set in order to enable the software interrupt at bit SWI (IRCON.1)
EADC	ADC interrupt Enable If EADC = 0, the A/D converter interrupt is disabled. If EADC = 1, the A/D converter interrupt is enabled.

7.1.2 Interrupt Request / Control Flags

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



Bit	Function
TF1	Timer 1 overflow Flag Set by hardware on timer/counter 1 overflow. Cleared by hardware when processor vectors to interrupt routine.
TF0	Timer 0 overflow Flag Set by hardware on timer/counter 0 overflow. Cleared by hardware when processor vectors to interrupt routine.
IE1	External Interrupt 1 request flag Set by hardware when external interrupt 1 edge is detected. Cleared by hardware when processor vectors to interrupt routine.
IT1	External interrupt 1 level/edge Trigger control flag If IT1 = 0, low-level-triggered external interrupt 1 is selected. If IT1 = 1, falling-edge-triggered external interrupt 1 is selected.
IE0	External Interrupt 0 request flag Set by hardware when external interrupt 0 edge is detected. Cleared by hardware when processor vectors to interrupt routine.
IT0	External interrupt 0 level/edge Trigger control flag If IT0 = 0, low level triggered external interrupt 0 is selected. If IT0 = 1, falling edge triggered external interrupt 0 is selected.

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

The **timer 0 and timer 1 interrupts** are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective timer/counter registers. When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when vectoring to the service routine.

Special Function Register T2CON (Address $C8_H$) **Reset Value: $00X00000_B$**

Bit No.	MSB							LSB	
	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	T2PS	I3FR	–	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

The shaded bits are not used for interrupt control.

Bit	Function
I3FR	External Interrupt 3 Falling/Rising edge control flag If I3FR = 0, external interrupt 3 is activated by a falling edge at P1.0/AN0/$\overline{INT3}$/CC0. If I3FR = 1, external interrupt 3 is activated by a rising edge at P1.0/AN0/$\overline{INT3}$/CC0.
–	This bit has no effect in the C505L.

The **external interrupt 3** ($\overline{INT3}$) can be activated by either a positive or negative transition, depending on bit I3FR in register T2CON. The flag that actually generates this interrupt is bit IEX3 in register IRCON. In addition, this flag will be set if a compare event occurs at pin P1.0/AN0/$\overline{INT3}$/CC0, regardless of the compare mode established and the transition at the respective pin. The flag IEX3 is cleared by hardware when vectoring to the service routine.

Special Function Register IRCON (Address $C0_H$) **Reset Value: 00_H**

	MSB							LSB	
Bit No.	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	SWI	IADC	IRCON

Bit	Function
EXF2	Timer 2 EXternal reload Flag EXF2 is set when a reload is caused by a falling edge on pin T2EX while EXEN2 = 1. If ET2 in IEN0 is set (timer 2 interrupt enabled), EXF2 = 1 will cause an interrupt. EXF2 can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.
TF2	Timer 2 overflow Flag Set by a timer 2 overflow and must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
IEX6	EXternal Interrupt 6 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at P1.3/AN3/INT6/CC3. Cleared by hardware when processor vectors to interrupt routine.
IEX5	EXternal Interrupt 5 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at P1.2/AN2/INT5/CC2. Cleared by hardware when processor vectors to interrupt routine.
IEX4	EXternal Interrupt 4 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at P1.1/AN1/INT4/CC1. Cleared by hardware when processor vectors to interrupt routine.
IEX3	EXternal Interrupt 3 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at P1.0/AN0/$\overline{\text{INT3}}$/CC0. Cleared by hardware when processor vectors to interrupt routine.
SWI	SoftWare Interrupt flag This bit must be set by software to generate an interrupt. Cleared by hardware when processor vectors to interrupt routine.
IADC	ADC Interrupt request flag Set by hardware at the end of an A/D conversion. Must be cleared by software.

The **timer 2 interrupt** is generated by the logical OR of bits TF2 and EXF2 in register IRCON. Neither of these flags is cleared by hardware when vectoring to the service routine. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared by software.

The **ADC interrupt** is generated by IADC bit in register IRCON. If an interrupt is generated, in any case the converted result in ADDAT is valid on the first instruction of the interrupt service routine. If continuous conversion is established, IADC is set once during each conversion. If an ADC interrupt is generated, flag IADC will have to be cleared by software.

The **external interrupts 4 to 6** (INT4, INT5 and INT6) are activated by a positive transition. The flags that actually generate these interrupts are bits IEX4, IEX5 and IEX6 in register IRCON. In addition, these flags will be set if a compare event occurs at the corresponding output pin P1.3/AN3/INT6/CC3, P1.2/AN2/INT5/CC2, and P1.1/AN1/INT4/CC1, regardless of the compare mode established and the transition at the respective pin. When an interrupt is generated, the flag that generated it is cleared by the on-chip hardware when vectoring to the service routine.

All of these interrupt request bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The only exceptions are the request flags IE0 and IE1. If the external interrupts 0 and 1 are programmed to be level-activated, IE0 and IE1 are controlled by the external source via pin $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, respectively. Thus, writing a one to these bits will not set the request flag IE0 and/or IE1. In this mode, interrupts 0 and 1 can only be generated by software and by writing a 0 to the corresponding pins $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3), provided that this will not affect any peripheral circuit connected to the pins.

The bit SWI (IRCON.1) can be set by software to vector to location $\mathbf{004B_H}$. Prior to setting this bit, the bit IEN1.1 should be set to enable this software interrupt. Care should be taken to avoid any erroneous interrupt generation while manipulating this bit.

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

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

The shaded bits are not used for interrupt control.

Bit	Function
TI	Serial interface Transmitter Interrupt flag Set by hardware at the end of a serial data transmission. Must be cleared by software.
RI	Serial interface Receiver Interrupt flag Set by hardware if a serial data byte has been received. Must be cleared by software.

The **serial port interrupt** is generated by a logical OR of flag RI and TI in SFR SCON. Neither of these flags is cleared by hardware when vectoring to the service routine. In fact, the service routine will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, and the corresponding bit will have to be cleared by software.

7.1.3 Interrupt Priority Registers

The lower six bits of these two registers are used to define the interrupt priority level of the interrupt groups as they are defined in **Table 7-1** in the next section.

Special Function Register IP0 (Address $A9_H$) **Reset Value: 00_H**
Special Function Register IP1 (Address $B9_H$) **Reset Value: $XX000000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$A9_H$	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0
Bit No.	7	6	5	4	3	2	1	0	
$B9_H$	–	–	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	IP1

The shaded bits are not used for interrupt control.

Bit	Function
IP1.x IP0.x	Interrupt group priority level bits (x = 0 - 5, see **Table 7-1**)
–	Reserved bits for future use. Read by CPU returns undefined values.

IP1.x	IP0.x	Function
0	0	Interrupt group x is set to priority level 0 (lowest)
0	1	Interrupt group x is set to priority level 1
1	0	Interrupt group x is set to priority level 2
1	1	Interrupt group x is set to priority level 3 (highest)

7.2 Interrupt Priority Level Structure

The following table shows the interrupt grouping of the C505L interrupt sources.

Table 7-1
Interrupt Source Structure

Interrupt Group	Associated Interrupts High priority ⟶ Low priority		Priority
1	External interrupt 0	ADC interrupt	High
2	Timer 0 overflow	Software Interrupt (SWI)	↓
3	External interrupt 1	External interrupt 3	
4	Timer 1 overflow	External interrupt 4	
5	Serial channel interrupt	External interrupt 5	
6	Timer 2 interrupt	External interrupt 6	Low

Each pair of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the SFR IP0 and one in IP1. A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another interrupt of the same or a lower priority. An interrupt of the highest priority level cannot be interrupted by another interrupt source.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is to be serviced first. Thus, within each priority level, there is a second priority structure determined by the following polling sequence:

- Within one interrupt group the “left” interrupt is serviced first, and
- The interrupt groups are serviced from top to bottom of the table.

7.3 How Interrupts Are Handled

The interrupt flags are sampled at S5P2 in each machine cycle. The sampled flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the 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 IEN0/IEN1 or IP0/IP1.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress is completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write access to registers IEN0/IEN1 or IP0/IP1, then at least one more instruction will be executed before vectoring to any interrupt; this delay guarantees that changes in 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 7-4**.

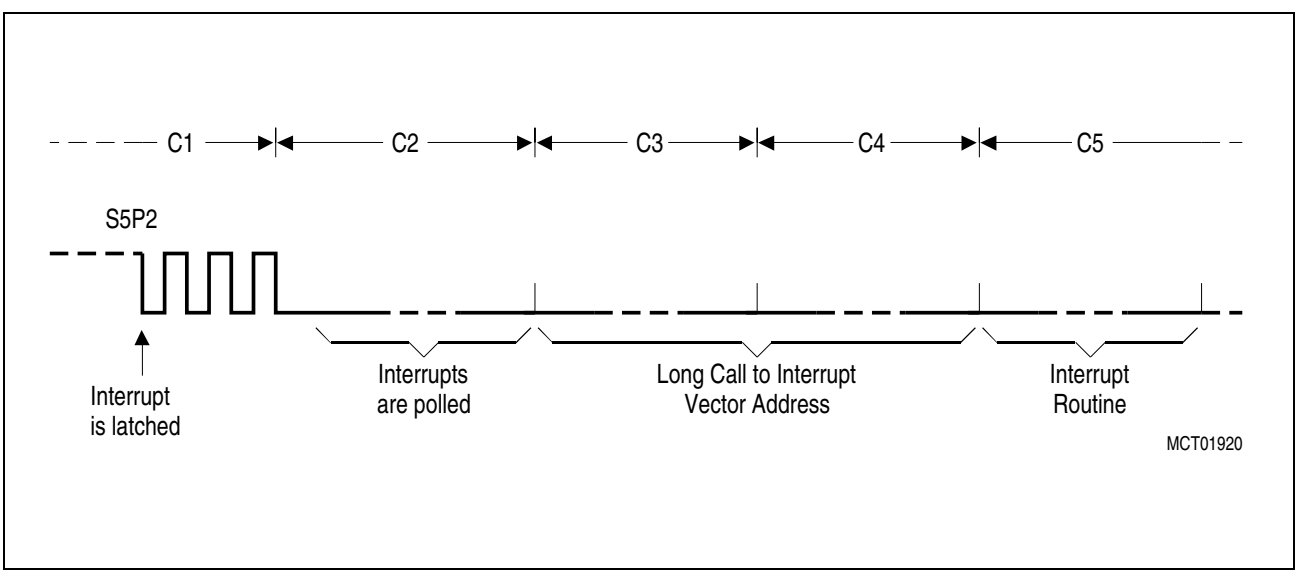


Figure 7-4
Interrupt Response Timing Diagram

Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in **Figure 7-4**, 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. If the latter occurs, the flag must be cleared by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the Program Status Word, PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored to, as shown in the following **Table 7-2**.

Table 7-2
Interrupt Source and Vectors

Interrupt Source	Interrupt Vector Address	Interrupt Request Flags
External Interrupt 0	0003_H	IE0
Timer 0 Overflow	$000B_H$	TF0
External Interrupt 1	0013_H	IE1
Timer 1 Overflow	$001B_H$	TF1
Serial Channel	0023_H	RI/TI
Timer 2 Overflow/Ext. Reload	$002B_H$	TF2/EXF2
A/D Converter	0043_H	IADC
Software Interrupt	$004B_H$	SWI
External interrupt 3	0053_H	IEX3
External Interrupt 4	$005B_H$	IEX4
External Interrupt 5	0063_H	IEX5
External interrupt 6	$006B_H$	IEX6
Wake-up from power-down mode	$007B_H$	IRTC (real-time clock wake-up only)

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 has left the current interrupt priority level. A simple RET instruction would also have returned execution to the interrupted program, but without clearing the internal interrupt status of that particular interrupt. In this case, no interrupt of the same or lower priority level would be acknowledged.

7.4 External Interrupts

External interrupts 0 and 1 can be programmed to be level-activated or negative-transition activated by setting or clearing bit ITx (x = 0 or 1), respectively, in register TCON. If ITx = 0, external interrupt x is triggered by a low level detected 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 = 1 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 interrupts 4, 5 and 6 are activated only by a positive transition. The external timer 2 reload trigger interrupt request flag EXF2 is activated by a negative transition at pin P1.5/AN5/T2EX only if bit EXEN2 is set.

Since the external interrupt pins (INT4, INT5 and INT6) are sampled once in each machine cycle, an input high or low should be held for at least 6 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin low for at least one cycle, and then hold it high for at least one cycle. This will ensure that the transition is recognized, so that the corresponding interrupt request flag will be set (see **Figure 7-5**). The external interrupt request flags will be cleared automatically by the CPU when the service routine is called.

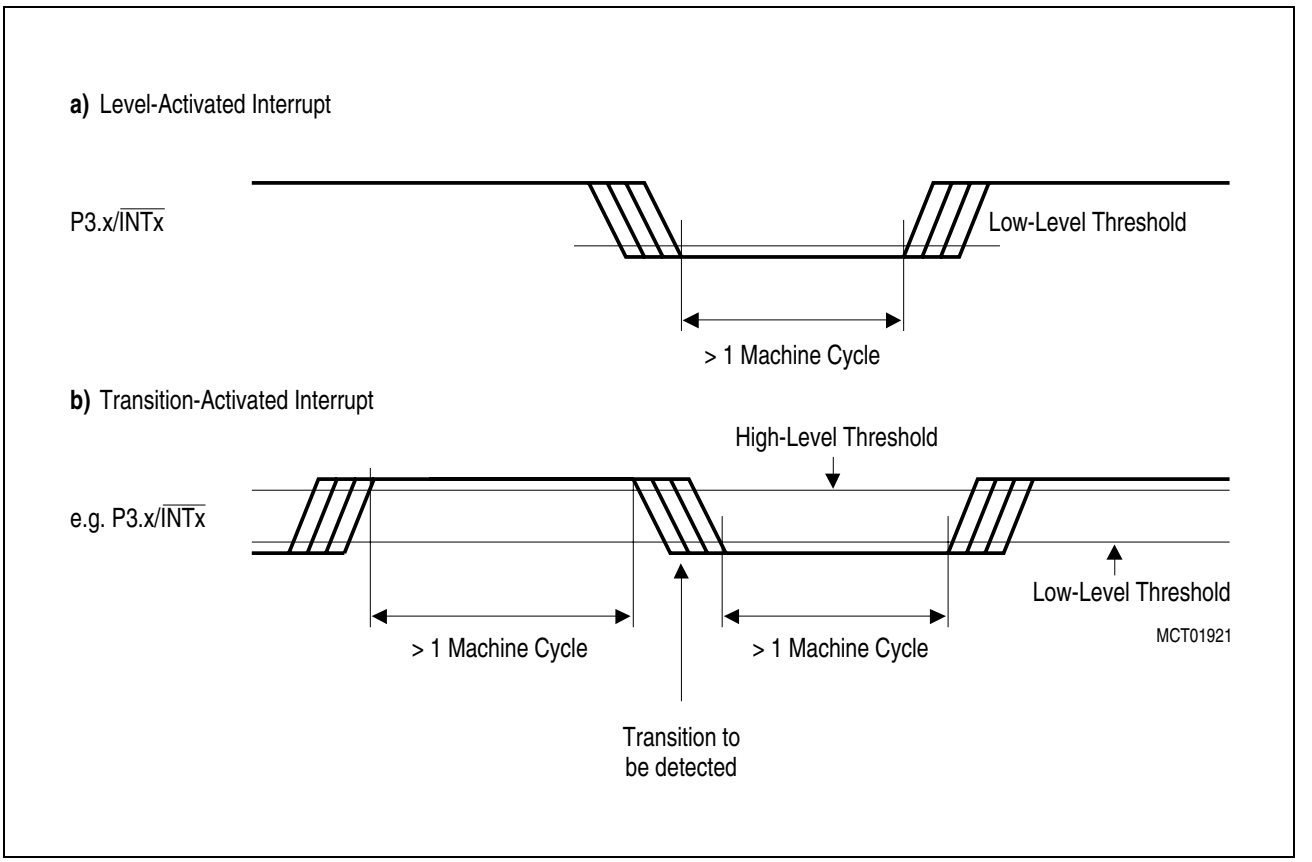


Figure 7-5
External Interrupt Detection

7.5 Interrupt Response Time

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

A longer response time would be obtained if the request was blocked by one of the three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles because the longest instructions (MUL and DIV) are only 4 cycles long. If the instruction in progress is RETI or a write access to registers IEN0, IEN1 or IP0, IP1, 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 in a single interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

8 Fail Safe Mechanisms

The C505L offers enhanced fail save mechanisms, which allow an automatic recovery from software upset or hardware failure:

- A programmable WatchDog Timer, with variable time-out period from 192 µs up to approx. 393.2 ms at 16 MHz (314.5 ms at 20 MHz).
- An Oscillator WatchDog (OWD) unit that monitors the on-chip oscillator and forces the microcontroller into reset state in case the on-chip oscillator fails. The OWD also provides the clock for a fast internal reset after power-on.

8.1 Programmable WatchDog Timer

To protect the system against software failure, the user's program has to clear the watchdog timer within a previously programmed time period. If the software fails to refresh the watchdog timer periodically, an internal hardware reset will be initiated. The software can be designed so 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 C505L is a 15-bit timer that is incremented by a count rate of $f_{OSC}/192$ up to $f_{OSC}/12$. The machine clock of the C505L is divided by two prescalers: A divide-by-2, and a divide-by-16 prescaler. For programming of the watchdog timer overflow rate, the upper 7 bits of the watchdog timer can be written. **Figure 8-1** shows the block diagram of the watchdog timer unit.

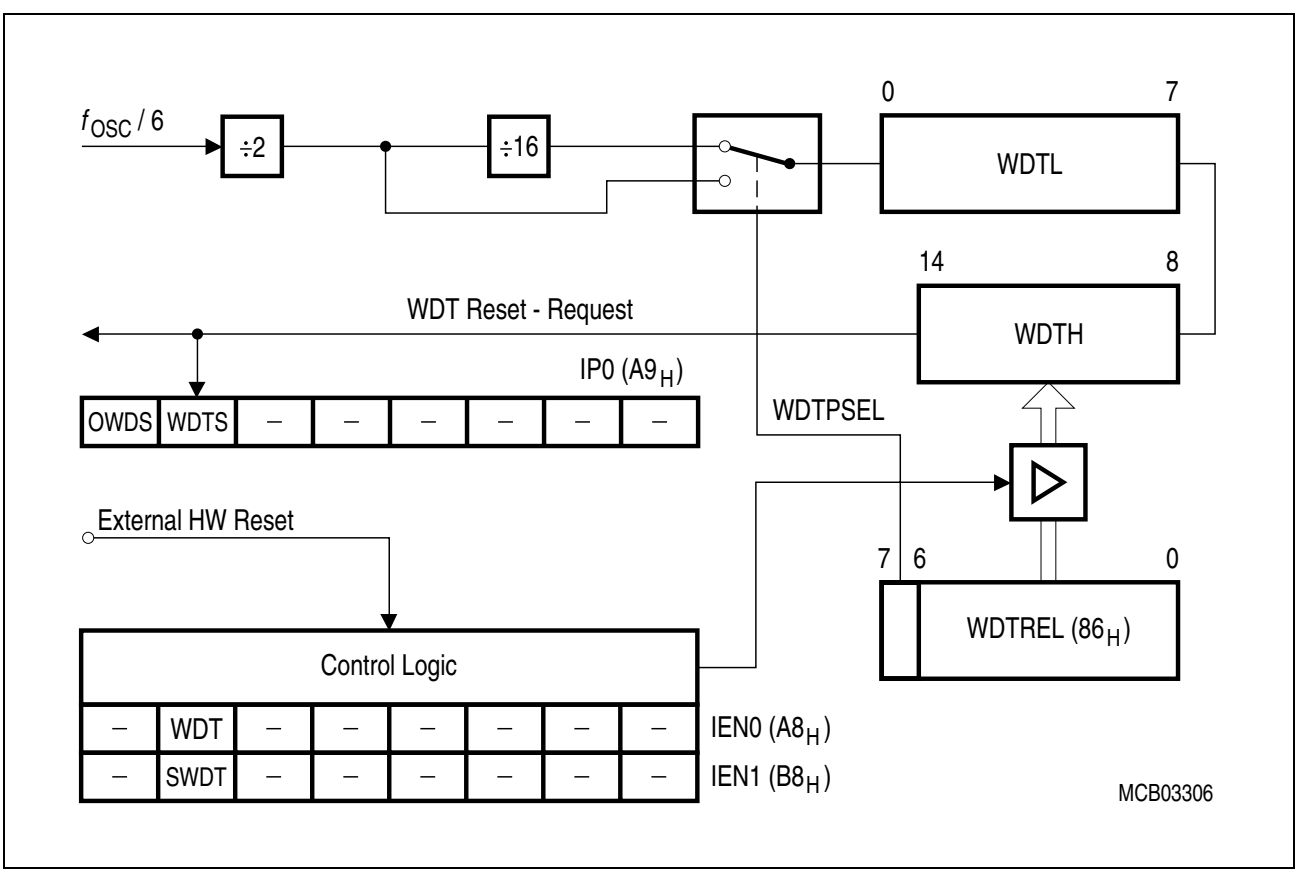
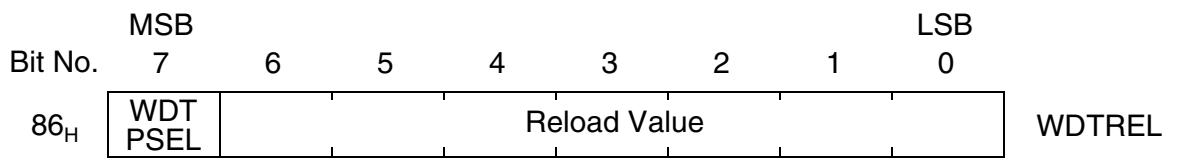


Figure 8-1
Block Diagram of the Programmable Watchdog Timer

8.1.1 Input Clock Selection

The input clock rate of the watchdog timer is derived from the system clock of the C505L. There is a prescaler available, which is software selectable and defines the input clock rate. This prescaler is controlled by bit WDTPSEL in the SFR WDTREL. **Table 8-1** shows resulting time-out periods at f_{OSC} = 12 and 16 MHz.

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



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

Table 8-1
Watchdog Timer Time-out Periods

WDTREL	Time-out Period		Comments
	f_{osc} = 12 MHz	f_{osc} = 16 MHz	
00_H	32.768 ms	24.576 ms	This is the default value
80_H	524.2 ms	393.2 ms	Maximum time period
$7F_H$	256 µs	192 µs	Minimum time period

8.1.2 Watchdog Timer Control / Status Flags

The watchdog timer is controlled by two control flags, WDT and SWDT (located in SFR IEN0 and IEN1), and one status flag, WDTS (located in SFR IP0).

Special Function Register IEN0 (Address $A8_H$) **Reset Value: 00_H**
Special Function Register IEN1 (Address $B8_H$) **Reset Value: 00_H**
Special Function Register IP0 (Address $A9_H$) **Reset Value: 00_H**

	MSB							LSB	
	AF_H	AE_H	AD_H	AC_H	AB_H	AA_H	$A9_H$	$A8_H$	
$A8_H$	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	IEN0
	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	ECAN	EADC	IEN1
Bit No.	7	6	5	4	3	2	1	0	
$A9_H$	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0

The shaded bits are not used for watchdog timer control.

Bit	Function
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.
WDTS	WatchDog Timer Status flag. Set by hardware when a watchdog timer reset occurred. Can be cleared and set by software.

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

8.1.3 Starting the Watchdog Timer

The watchdog timer can be started by software (bit SWDT in SFR IEN1), 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 IP0 is set). A refresh of the watchdog timer is done by setting bits WDT (SFR IEN0) 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 **Chapter 9**). Therefore, the watchdog timer cannot reset the device when one of the power saving modes has been entered.

8.1.4 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 cannot be stopped by software but can only be refreshed to the reload value by first setting bit WDT (IEN0.6) and by the next instruction setting SWDT (IEN1.6). Bit WDT will be cleared automatically during the second machine cycle after having been set. For this reason, setting SWDT bit has to be a one-cycle instruction (e.g. SETB SWDT). This double-instruction refresh of the watchdog timer is implemented to minimize the chance of an unintentional reset of the watchdog.

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

The reload register WDTREL can be written to at any time. Therefore, a periodical refresh of WDTREL can be added to the starting procedure mentioned above for the watchdog timer. Thus, a wrong reload value caused by an error during the write operation to the WDTREL can be corrected by software.

8.1.5 Watchdog Reset and Watchdog Status Flag

If the software fails to refresh 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 cycles or 128 cycles). This internal reset differs from an external one only in so far as the watchdog timer is not disabled, and bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. **Figure 8-2** shows a block diagram of all reset requests in the C505L, and the function of the watchdog status flags. The WDTS flag is a flip-flop that is set by a watchdog timer reset and cleared by an external hardware reset. Bit WDTS allows the software to examine which source activated the reset. The watchdog timer status flag can also be cleared by software.

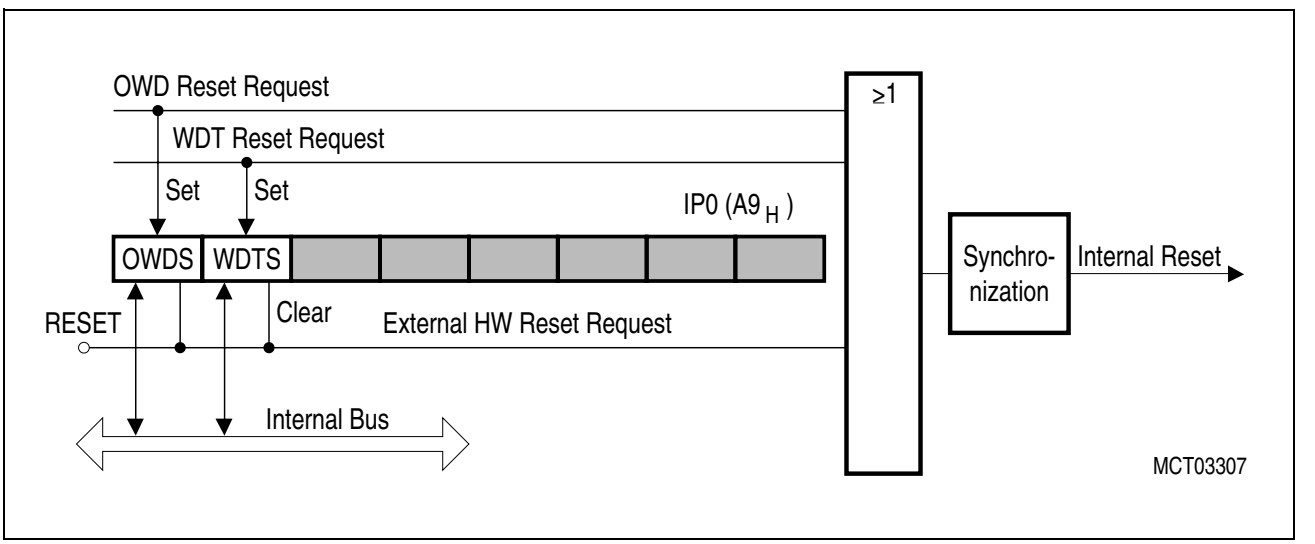


Figure 8-2
Watchdog Timer Status Flags and Reset Requests

8.2 Oscillator Watchdog Unit

The oscillator watchdog serves three functions:

- **Monitoring 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 device 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 device starts program execution from address 0000_H 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 and the monitoring function also work identically.
- **Control of wake-up from software power-down mode**
  When the power-down mode is left by a low level signal at the P3.2/$\overline{INT0}$ pin or an active Real Time Clock Interrupt Request flag IRTC, the oscillator watchdog unit ensures that the microcontroller resumes operation (execution of the power-down wake-up interrupt) with the nominal clock rate. In the power-down mode, the RC oscillator and the on-chip oscillator are stopped. Both oscillators are started again when power-down mode is released. When the on-chip oscillator has a higher frequency than the RC oscillator, the microcontroller starts program execution by processing a power down interrupt after a final delay (typ. 1 ms) in order to allow the on-chip oscillator to stabilize.

Note: The oscillator watchdog unit is always enabled.

Special Function Register IP0 (Address $A9_H$) **Reset Value: 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$A9_H$	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0

The shaded bits are not used for fail-safe control.

Bit	Function
OWDS	Oscillator WatchDog Status Flag. Set by hardware when an oscillator watchdog reset occurred. Can be set and cleared by software.

8.2.1 Detailed Description of the Oscillator Watchdog Unit

Figure 8-2 is a block diagram of the oscillator watchdog unit. The oscillator watchdog consists of an internal RC oscillator that provides the reference frequency for the comparison with the frequency of the on-chip oscillator. **Figure 8-3** also shows the additional provisions for integration of the wake-up from power-down mode.

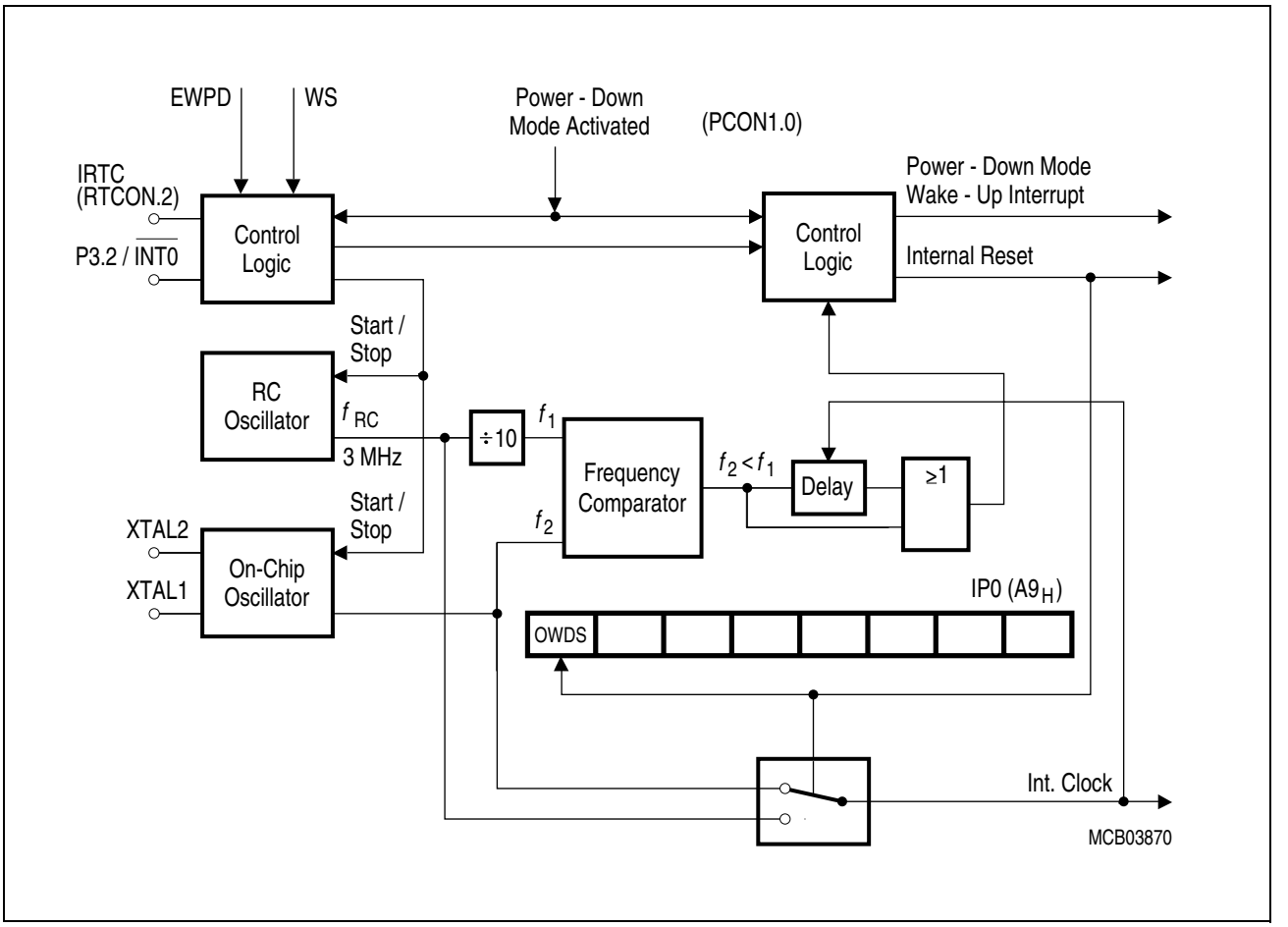


Figure 8-3
Functional Block Diagram of the Oscillator Watchdog

The frequency of the RC oscillator is divided by 10 and compared with the on-chip oscillator's frequency. If the frequency of the on-chip oscillator is lower than the frequency derived from the RC oscillator the watchdog detects a failure condition. For example, 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 device is being clocked even if the on-chip oscillator has stopped or has not yet started. At the same time, the watchdog activates the internal reset in order to bring the device to its defined reset state. The reset is performed because a clock signal 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 is, however, stopped); and bit OWDS is set. This allows the software to examine error conditions detected by the watchdog unit even if meanwhile an oscillator failure occurred.

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 1 ms. Within that time, the clock is still supplied by the RC oscillator and the device is held in reset. This allows a reliable stabilization of the on-chip oscillator. After that, the watchdog switches the clock supply back to the on-chip oscillator and releases the oscillator watchdog reset. If no other reset is applied at this time, the device will start program execution. If an external reset or a watchdog timer reset is active, however, the device will retain the reset state until the other 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).

If software power-down mode is activated, the RC oscillator and the on-chip oscillator are stopped. Both oscillators are again started in power-down mode when a low level signal is detected at either the P3.2/$\overline{\text{INT0}}$ input pin or the real-time clock interrupt flag (IRTC), when bit EWPD in SFR PCON1 is set (wake-up from power-down mode enabled). The wake-up source is chosen from one of P3.2/$\overline{\text{INT0}}$ and IRTC (RTCON.3) by bit WS in SFR PCON1. In this case, the oscillator watchdog does not execute an internal reset during start-up of the on-chip oscillator. After the start-up phase of the on-chip oscillator, the watchdog generates a power-down mode wake-up interrupt. Detailed description of the wake-up from software power-down mode is given in **Section 9.4.2**.

8.2.2 Fast Internal Reset after Power-On

The C505L 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) do not enter their default reset state before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed in order to bring the device into the correct reset state. In particular, the start up time of the oscillator is relatively long (typ. 1 ms) if a crystal is used. During this time period, the pins are in an undefined state that could have severe effects, e.g., on actuators connected to port pins.

In the C505L, 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 (typically less than 2 µs). Then the watchdog circuitry detects a failure condition for the on-chip oscillator because it has not yet started.

A failure is always recognized if the watchdog’s RC oscillator runs faster than the on-chip oscillator. As long as this condition is valid the watchdog uses the RC oscillator output as clock source for the chip. This allows correct resetting of the device and brings all ports to the defined state (see also **Chapter 5** of this manual).

9 Power Saving Modes

The C505L provides three basic power-saving modes, the idle mode, the slow-down mode and the power-down mode.

9.1 Power-saving Mode Control Registers

The functions of the power-saving modes are controlled by bits in the Special Function Registers (SFRs) PCON and PCON1. The SFR PCON is located at SFR address 87_H. PCON1 is located in the mapped SFR area (RMAP = 1) at SFR address 88_H. Bit RMAP, which controls the access to the mapped SFR area, is located in SFR SYSCON ($B1_H$).

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.

The slow-down mode is controlled by the bit SD located in SFR PCON. Furthermore, SFR PCON contains two general-purpose flags. For example, the flag bits GF0 and GF1 can be used to indicate whether an interrupt occurred during normal operation or during idle mode. For that function, an instruction that activates idle mode can also set one or both flag bits. When idle mode is terminated by an interrupt, the interrupt service routine can examine the flag bits.

Special Function Register PCON (Address 87_H) **Reset Value: 00_H**

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

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

Symbol	Function
PDS	Power-Down Start bit The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode
IDLS	IDLe Start bit The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
SD	Slow-Down mode bit When set, the slow-down mode is enabled
GF1	General-purpose flag
GF0	General-purpose flag
PDE	Power-Down Enable bit When set, starting of the power-down is enabled
IDLE	IDLe mode Enable bit When set, starting of the idle mode is enabled

Register LCON (Address $F3DD_H$) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$F3DD_H$	DSB1	DSB0	0	0	0	0	CSEL	LCEN	LCON

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

Bit	Function
LCEN	ENables LCD controller LCEN = 0: LCD Controller is disabled (default after reset). LCEN = 1: LCD Controller is enabled.
CSEL	LCD input Clock SELection CSEL = 1: Use RTC Clock input (32.768 KHz) for f_{LCDIN} CSEL = 0: Use system clock (f_{OSC}) for f_{LCDIN}

Register RTCON (Address $F3F0_H$) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$F3F0_H$	0	0	0	0	RTPD	IRTC	ERTC	RTCS	RTCON

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

Bit	Function
RTPD	Real-Time clock Power-Down enable RTPD = 0: Real-Time clock is in operation RTPD = 1: Real-Time clock is powered down Real-Time clock is in operation by default after reset.

Registers LCON and RTCON are used for controlling the power-down modes of the LCD controller and the real-time clock, respectively. Please refer to **Section 9.4** for further details.

Special Function Register PCON1 (Mapped Address 88_H) **Reset Value: $0XX0XXXX_B$**

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

Symbol	Function
EWPD	External Wake-up from Power-Down enable bit Setting EWPD before entering power-down mode, enables wake-up from power-down mode capability.
WS	Wake-up from power-down Source select WS = 0: wake-up via P3.2/$\overline{INT0}$ (external wake-up) WS = 1: wake-up via real-time clock interrupt
–	Reserved bits for future use. Read by CPU returns undefined values.

9.2 Idle Mode

In the idle mode, the C505L's oscillator continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D converter (ADC), the LCD controller, the real-time clock, 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 (PSW), accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption which can be achieved in the idle mode depends on the number of peripherals running. If all timers are stopped, and the ADC and the serial interfaces are not running, the maximum power reduction can be achieved. This state is also the test condition for the idle mode current (I_{DD}).

Thus, the user has to take care as to which peripheral(s) 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 that time when the idle mode was activated. If some pins are programmed to serve as alternate functions they still continue to output during idle mode if the assigned function is on. This applies especially to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ are held at logic high levels.

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 that makes it possible to "freeze" the processor's status, either for a predefined time, or until an external event causes the controller to revert to normal operation, as discussed below. The watchdog timer is the only peripheral which is stopped automatically during 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 subsequent 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 be cleared automatically 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 entering the idle mode unintentionally, which would leave the watchdog timer unable to protect the system.

Note

PCON is not a bit-addressable register, so the sequence mentioned above for entering the idle mode is accomplished 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. The CPU operation is resumed, the interrupt will be serviced, and the next instruction to be executed after the RETI instruction will be the one following the instruction that had set the 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 for only two machine cycles for a complete reset.

9.3 Slow-down Mode Operation

In some applications where power consumption and dissipation are critical, the controller might run for a certain time at reduced speed (e.g. if the controller is waiting for an input signal). Since in CMOS devices there is an almost linear dependency between the operating frequency and the power supply current, a reduction of the operating frequency results in reduced power consumption.

In the slow-down mode, all signal frequencies that are derived from the oscillator clock are divided by 32.

The slow-down mode is activated by setting the bit SD in SFR PCON. If the slow-down mode is enabled, the clock signals for the CPU and the peripheral units are reduced to 1/32 of the nominal clock rate. The controller actually enters the slow-down mode after a short synchronization period (max. two machine cycles). The slow-down mode is terminated by clearing bit SD.

The slow-down mode can be combined with the idle mode by performing the following double instruction sequence:

```
ORL   PCON,#00000001B    ; preparing idle mode: set bit IDLE (IDLS not set)
ORL   PCON,#00110000B    ; entering idle mode combined with the slow-down mode:
                         ; (IDLS and SD set)
```

There are two ways to terminate the combined idle and slow-down mode:

- The idle mode can be terminated by activation of any enabled interrupt. The CPU operation is resumed, the interrupt will be serviced, and the next instruction to be executed after the RETI instruction will be the one following the instruction that had set the bits IDLS and SD. Nevertheless the slow-down mode remains enabled, and if it is necessary to terminate this mode, that can be accomplished by clearing the bit SD in the corresponding interrupt service routine or at any point in the program where the user no longer requires the slow-down mode.
- The combined idle and slow-down mode can also be terminated by a hardware reset. Since the oscillator is still running, the hardware reset has to be held active for only two machine cycles for a complete reset.

9.4 Software Power-down Modes

In order to achieve different levels of power-saving, the C505L has three major software power-down modes as described below:

- **Software power-down mode 1**, in which all the peripheral blocks and the CPU are stopped. In this mode, the RC oscillator and the on-chip oscillator that operates with the XTAL1 and XTAL2 pins are stopped. Therefore, all functions of the microcontroller are stopped and only the contents of the on-chip RAM, XRAM and the SFRs are maintained. The port pins, which are controlled by their port latches, output the values that are held by their SFRs. The port pins that serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode. ALE and $\overline{\text{PSEN}}$ are held at logic low level (see **Table 9-1**).
- **Software power-down mode 2**, in which only the Real-time clock and LCD controller are operating. In this mode, the CPU and the rest of the peripherals are stopped. The RC oscillator and the on-chip oscillator are stopped, the real-time clock oscillator that operates with the XTAL3 and XTAL4 pins is still running and the real-time count is maintained in this mode.
- **Software power-down mode 3**, in which only the real-time clock is operating. In this mode, the clock input into the CPU, LCD controller and the rest of the peripherals are stopped. The only difference between this mode and mode 2 is that the LCD controller is also stopped in this mode. The LCD controller output pins are inactive in this stage and should not be used for any input function.

In both software power-down modes 2 and 3, all the functions of the microcontroller other than those described above are stopped, and the contents of the on-chip RAM, XRAM and SFRs are maintained. The unused pins in these modes have the behavior as in the software power-down mode 1.

In all the software power-down modes, V_{DD} can be reduced to minimize power consumption. In the case of the software power-down mode 3, V_{DD} can be reduced to 3 V (lower specification limit). It must be ensured, however, that V_{DD} is not reduced before any of the power-down modes is invoked, and that V_{DD} is restored to its normal operating level before leaving the power-down mode.

Any of these software power-down modes can be exited either by an active reset signal or by a wake-up request. Using reset to leave power-down mode puts the microcontroller with its SFRs into the reset state. Program execution then starts from the address 0000_H. Using a wake-up request to exit the power-down mode starts the RC oscillator and the on-chip oscillator and maintains the state of the SFRs, which were frozen when power-down mode was entered.

When the C505L is in software power-down mode 1, a wake-up operation is possible only through P3.2/$\overline{\text{INT0}}$. There are two ways to use a wake-up request to exit power-down modes 2 and 3:

- Wake-up via P3.2/$\overline{\text{INT0}}$ pin when bit WS in SFR PCON1 is cleared, and
- Wake-up via the real-time clock interrupt when bit WS in SFR PCON1 is set

9.4.1 Invoking Software Power-down Modes

The C505L's software power-down modes can be entered as shown below:

- **Software Power-down Mode 1 -** This mode is entered by first ensuring that both the LCD controller and the real-time clock are disabled. This is done by clearing bit LCEN (LCON.0) and setting bit RTPD (RTCON.3).

Once these conditions are fulfilled, software power-down mode 1 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 subsequent 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 be cleared automatically after having been set, and the value shown by reading one of these bits is always 0. The double instruction is implemented to minimize the chance of entering the power-down mode unintentionally, which could possibly "freeze" the chip's activity in an undesired status.

PCON is not a bit-addressable register, so the above mentioned sequence for entering the power-down mode can be accomplished 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, enter power-down
```

- **Software Power-down Mode 2** - This mode is entered by first ensuring that both the LCD controller and the real-time clock are enabled. The following conditions should be met:
  - bit LCEN (LCON.0) is set,
  - bit CSEL (LCON.1) is set, and
  - bit RTPD (RTCON.3) is cleared.

Once these conditions are fulfilled, the C505L can enter software power-down mode 2 with the two instruction sequence as in mode 1.

- **Software Power-down Mode 3 -** This mode is entered by first ensuring that the real-time clock is enabled and the LCD controller is disabled. The following conditions should be met:
  - bit RTPD (RTCON.3) is cleared, and
  - bit LCEN (LCON.0) is cleared.

Once these conditions are fulfilled, the C505L can enter software power-down mode 3 with the two instruction sequence as in mode 1.

In any of the above modes, the instruction that sets bit PDS is the last instruction executed before going into power-down mode.

Note: Before entering the power-down mode, any A/D conversion in progress must be stopped.

9.4.2 Exit from Software Power-down Mode

The C505L can exit the software power-down modes in one of the following 2 ways:

- Hardware reset
- Wake-up from power-down mode through pin P3.2/ $\overline{\text{INT0}}$ or real-time clock interrupt

If the bit EWPD in SFR PCON1 is 0 during power-down entry, the only way to exit from the power-down mode is a hardware reset. This reset will redefine all the SFRs but will not change the contents of the internal RAM and XRAM. The reset signal that terminates the power-down mode also restarts the RC oscillator and the on-chip oscillator. The reset operation should not be activated before V_{DD} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

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

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

Figure 9-1 shows the procedure which must be executed when power-down modes are exited via the P3.2/$\overline{INT0}$ wake-up request capability.

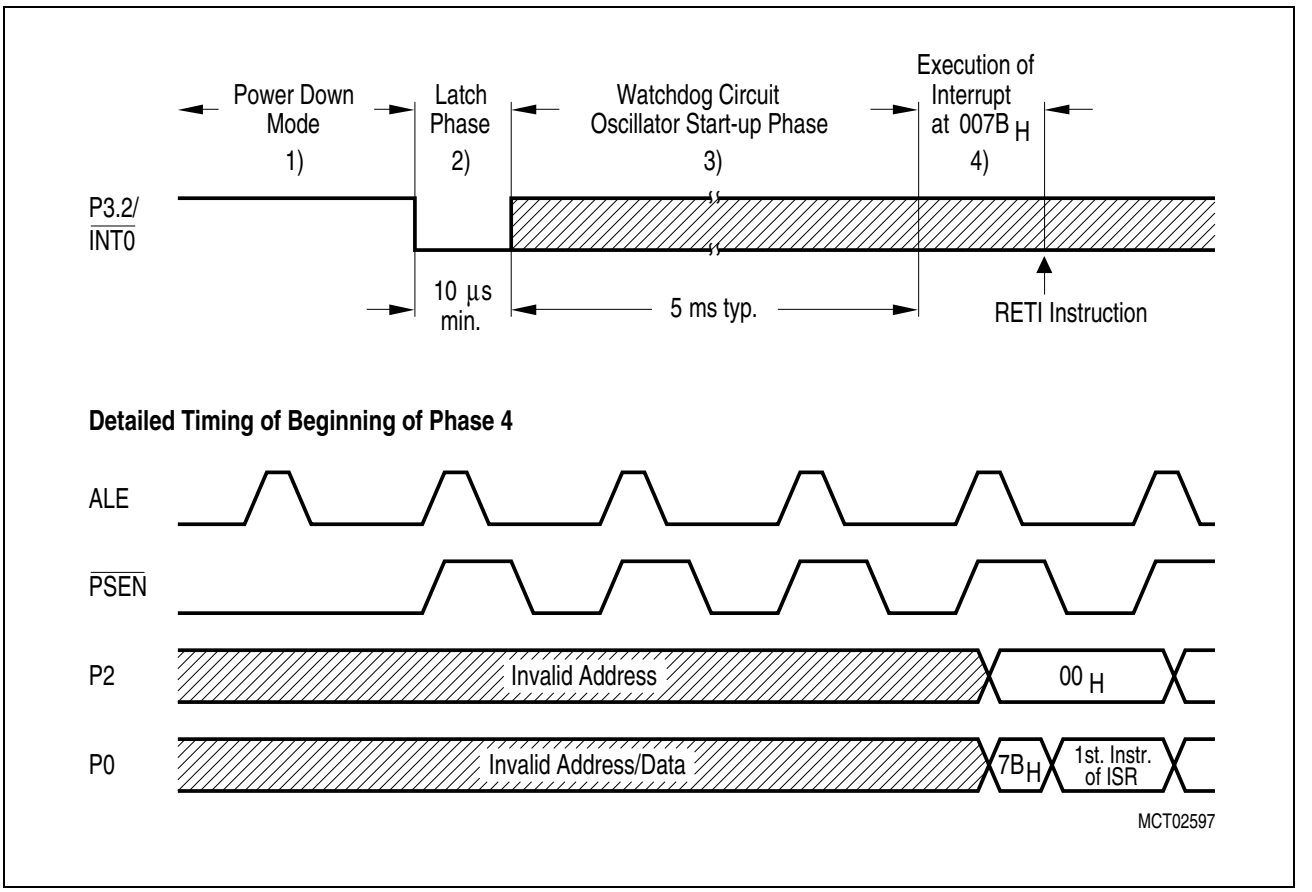


Figure 9-1
Wake-up from Power-down Mode Procedure

When the power-down mode wake-up capability has been enabled (bit EWPD in SFR PCON1 set) prior to entering power-down mode, and bit WS in SFR PCON1 is cleared, the power-down mode can be exited via $\overline{INT0}$ while executing the following procedure:

1. In power-down mode, pin P3.2/$\overline{INT0}$ must be held at high level.
2. Power-down mode is exited when P3.2/$\overline{INT0}$ goes low for at least 10 µs (latch phase). After this delay, the internal RC oscillator, and the on-chip oscillator are started, the state of pin P3.2/$\overline{INT0}$ is internally latched, and P3.2/$\overline{INT0}$ can be set again to a high level if required. Thereafter, the oscillator watchdog unit controls the wake-up procedure in its start-up phase.
3. The oscillator watchdog unit starts operation. When the on-chip oscillator clock's stable nominal frequency has been detected, the microcontroller starts again and initiates the power-down wake-up interrupt. The interrupt address of the first instruction to be executed after wake-up is $007B_H$. ALE and $\overline{PSEN}$ are in their power-down state up to this time. At the end of phase 3, the CPU processes the interrupt call and during these two machine cycles, ALE and $\overline{PSEN}$ behave as shown in **Figure 9-1** (i.e., at the beginning of phase 4). Instruction fetches during the interrupt call are discarded, however.

4. After the RETI instruction of the power-down wake-up interrupt routine has been executed, the instruction that follows the double instruction sequence that initiates the power-down mode will be executed. The peripheral units timer 0/1/2 and watchdog timer are frozen until the end of phase 4.

All of the C505L's interrupts are disabled from phase 2 until the end of phase 4. Other interrupts can be handled first after the RETI instruction of the wake-up interrupt routine.

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

Prior to entering the software power-down mode, the port latch of SFR P3.2 (P3.2/$\overline{INT0}$ pin) should contain a "1". Otherwise, the wake-up sequence described above will be started immediately after the power-down mode has been entered.

The wake-up routine initiated by the real-time clock interrupt is similar to the wake-up from P3.2/$\overline{INT0}$, and can be used to wake-up from power-down modes 2 and 3. For this to occur, it is necessary to enable both the wake-up capability (bit EWPD in SFR PCON1 should be set) and the real-time clock interrupt have to be enabled (bit ERTC in RTCON). Additionally, the real-time clock should be selected as the source of the wake-up request (bit WS in SFR PCON1 must be set). An interrupt can then be generated by the real-time clock at a predetermined time, depending on the setting of the RTINT register. This interrupt will then be used as a wake-up request. The flag IRTC is set by hardware and has to be cleared by software. The handling of such a wake-up request is, however, identical to the handling of the wake-up through P3.2/$\overline{INT0}$.

The real-time clock wake-up interrupt has no effect on the device, unless the C505L has entered the power-down mode.

9.5 State of Pins in Software-initiated Power-saving Mode

In the idle mode and in the software power-down mode 1, the port pins of the C505L have a well-defined status which is listed in the following **Table 9-1**. This state of some pins also depends on the location of the code memory (internal or external).

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

Outputs	Last Instruction Executed from Internal Code Memory		Last Instruction Executed from External Code Memory	
	Idle	**Power-down**	**Idle**	**Power-down**
ALE	High	Low	High	Low
$\overline{\text{PSEN}}$	High	Low	High	Low
PORT 0	Data	Data	Float	Float
PORT 2	Data	Data	Address	Data
PORT 1, 3, 4 and 5	Data/alternate outputs	Data/last output	Data/alternate outputs	Data/last output

In software power-down mode 2, all the port pins are in the states as shown in **Table 9-1**, except for the enabled LCD output pins at Ports 3, 4 and 5. These pins output values corresponding to their digit registers.

In the software power-down mode 3, all the port pins are in the states as shown in **Table 9-1**.

10 OTP Memory Operation

The C505L is the one-time programmable (OTP) version of the C505L microcontroller with a 32-Kbyte OTP program memory. The C505L has fast programming cycles (1 byte per 100 µs). Also, several levels of OTP memory protection can be selected.

10.1 Programming Configuration

To program the device, the C505L must be put into the programming mode. Typically, this is done not “in-system” but with special programming hardware instead. In the programming mode, the C505L operates as a slave device similar to an EPROM stand-alone memory device, and must be controlled with address/data information, control lines, and an external 11.5 V programming voltage.

In the programming mode, port 0 provides the bidirectional data lines and port 2 is used for the multiplexed address inputs. The upper address information at port 2 is latched with the signal PALE. The inputs RESET, $\overline{\text{PSEN}}$, $\overline{\text{EA}}/V_{PP}$, ALE and PMSEL1/0 and $\overline{\text{PSEL}}$ are used for basic programming mode selection. Furthermore, the inputs PMSEL1,0 are required to select the access types (e.g., program/verify data, write lock-bits, and so forth) in the programming mode. V_{DD}/V_{SS} and a clock signal at the XTAL pins must be applied to the C505L in the programming mode. The 11.5 V external programming voltage is input through the $\overline{\text{EA}}/V_{PP}$ pin.

Figure 10-1 shows the pins of the C505L that are required for controlling of the OTP programming mode.

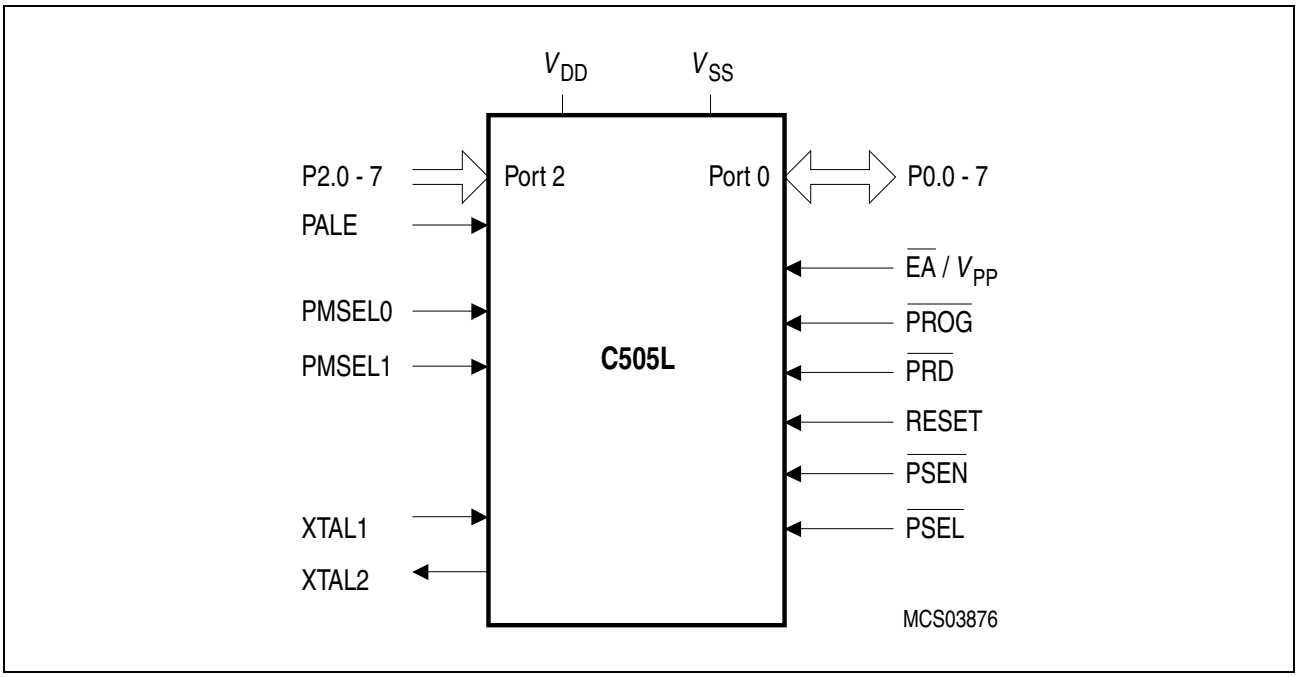


Figure 10-1
Programming Mode Configuration

10.2 Pin Configuration

Figure 10-2 shows the detailed pin configuration of the C505L in programming mode.

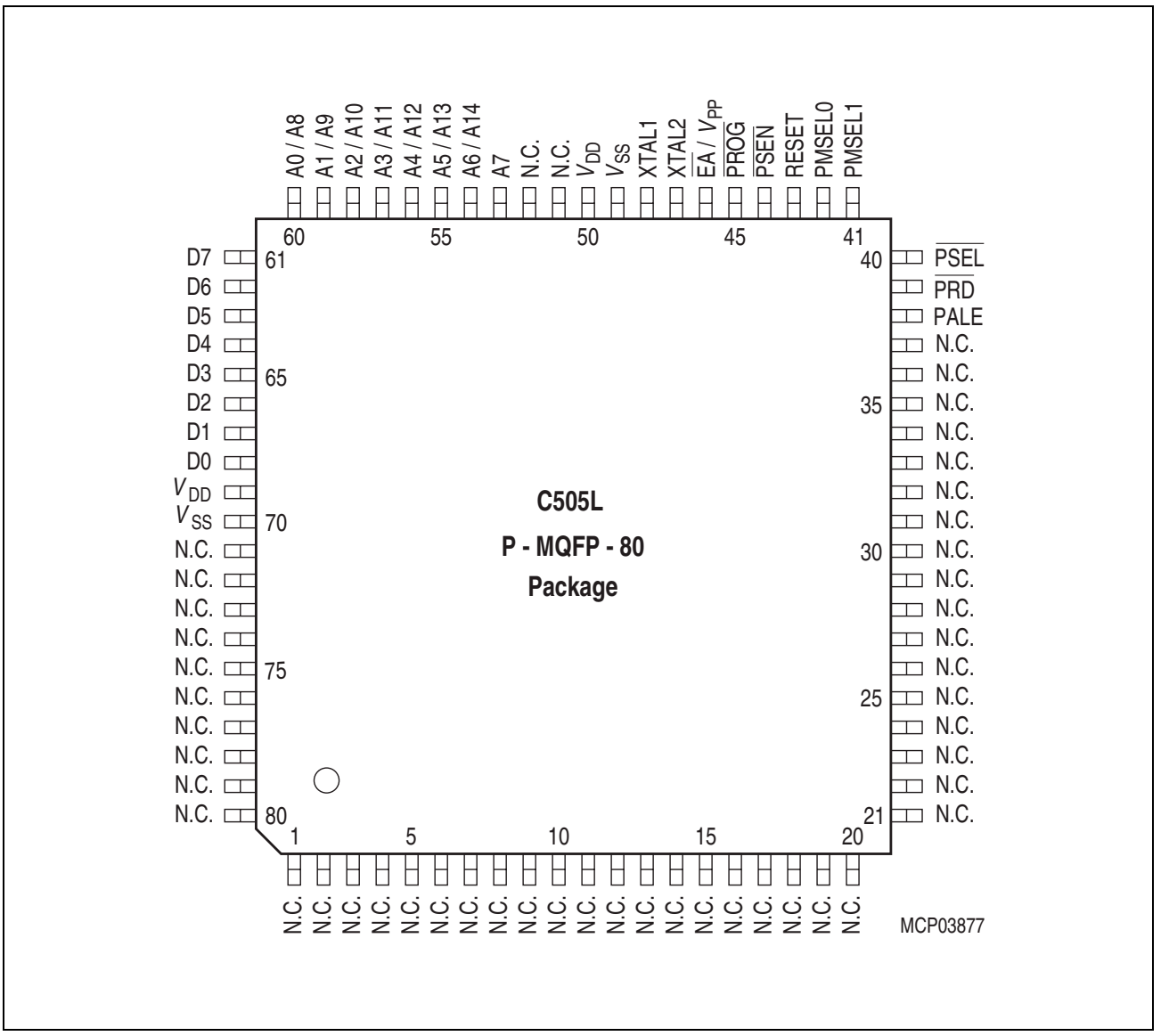


Figure 10-2
OTP Programming Mode Pin Configuration (top view)

10.3 Pin Definitions

Table 10-2 is a functional description of all C505L pins that are required for OTP memory programming.

Table 10-2
Pin Definitions and Functions of the C505L in Programming Mode

<table>
<tr><th>Symbol</th><th>Pin Number
P-MQFP-80</th><th>I/O *)</th><th>Function</th></tr>
<tr><td>RESET</td><td>43</td><td>I</td><td>Reset
This input must be at static “1” (active) level during the whole programming mode.</td></tr>
<tr><td>PMSEL0
PMSEL1</td><td>42
41</td><td>I
I</td><td>Programming Mode SELection pins
These pins are used to select the different access modes in programming mode. PMSEL1,0 must satisfy a setup time to the rising edge of PALE. When the logic level of PMSEL1,0 is changed, PALE must be at low level.
<table>
<tr><th>PMSEL1</th><th>PMSEL0</th><th>Access Mode</th></tr>
<tr><td>0</td><td>0</td><td>Reserved</td></tr>
<tr><td>0</td><td>1</td><td>Read signature bytes</td></tr>
<tr><td>1</td><td>0</td><td>Program/read lock-bits</td></tr>
<tr><td>1</td><td>1</td><td>Program/read OTP memory byte</td></tr>
</table></td></tr>
<tr><td>$\overline{\text{PSEL}}$</td><td>40</td><td>I</td><td>Basic Programming mode SELect
This input is used for the basic programming mode selection and must be switched according to Figure 10-3.</td></tr>
<tr><td>$\overline{\text{PRD}}$</td><td>39</td><td>I</td><td>Programming mode ReaD strobe
This input is used for read access control for OTP memory read, version byte read, and lock-bit read operations.</td></tr>
<tr><td>PALE</td><td>38</td><td>I</td><td>Programming Address Latch Enable
PALE is used to latch the high address lines. The high address lines must satisfy a setup and hold time to/from the falling edge of PALE. PALE must be at a low level when the logic level of PMSEL1,0 is changed.</td></tr>
<tr><td>XTAL2</td><td>47</td><td>O</td><td>XTAL2
Output of the inverting oscillator amplifier.</td></tr>
<tr><td>XTAL1</td><td>48</td><td>I</td><td>XTAL1
Input to the oscillator amplifier.</td></tr>
</table>

*) I = Input
O = Output

Table 10-2
Pin Definitions and Functions of the C505L in Programming Mode (cont'd)

Symbol	Pin Number P-MQFP-80	I/O *)	Function
V_{SS}	49, 70	–	**Circuit ground potential** Must be applied in programming mode.
V_{DD}	50, 69	–	**Power supply terminal** Must be applied in programming mode.
P2.0-7	60-53	I	**Address lines** P2.0-7 are used as multiplexed address input lines A0-A7 and A8-A14. A8-A14 must be latched with PALE.
$\overline{\text{PSEN}}$	44	I	**Program Store ENable** This input must be at static "0" level during the whole programming mode.
$\overline{\text{PROG}}$	45	I	**PROGramming mode write strobe** This input is used in programming mode as a write strobe for OTP memory program, and lock-bit write operations. During basic programming mode selection a low level must be applied to $\overline{\text{PROG}}$.
$\overline{\text{EA}}/V_{PP}$	46	–	**Programming voltage** This pin must be at 11.5 V (V_{PP}) voltage level during programming of an OTP memory byte or lock-bit. During an OTP memory read operation, this pin must be at V_{IH} high level. This pin is also used for basic programming mode selection. At basic programming mode selection a low level must be applied to $\overline{\text{EA}}/V_{PP}$.
P0.7-0	68-61	I/O	**Data lines 0-7** During programming mode, data bytes are transferred via the bidirectional D7-0 lines that are located at port 0.
N.C.	1-37, 51-52, 71-80	–	**Not Connected** These pins should not be connected in programming mode.

*) I = Input
O = Output

10.4 Programming Mode Selection

The selection of the OTP programming mode can be separated into two different parts:

- Basic programming mode selection
- Access mode selection

With basic programming mode selection, the device is put into the mode in which it is possible to access the OTP memory through the programming interface logic. After selection of the basic programming mode, OTP memory accesses are executed by using one of the access modes. These access modes include OTP memory byte program/read, version byte read, and program/read lock byte operations.

10.4.1 Basic Programming Mode Selection

The basic programming mode selection scheme is shown in **Figure 10-3**.

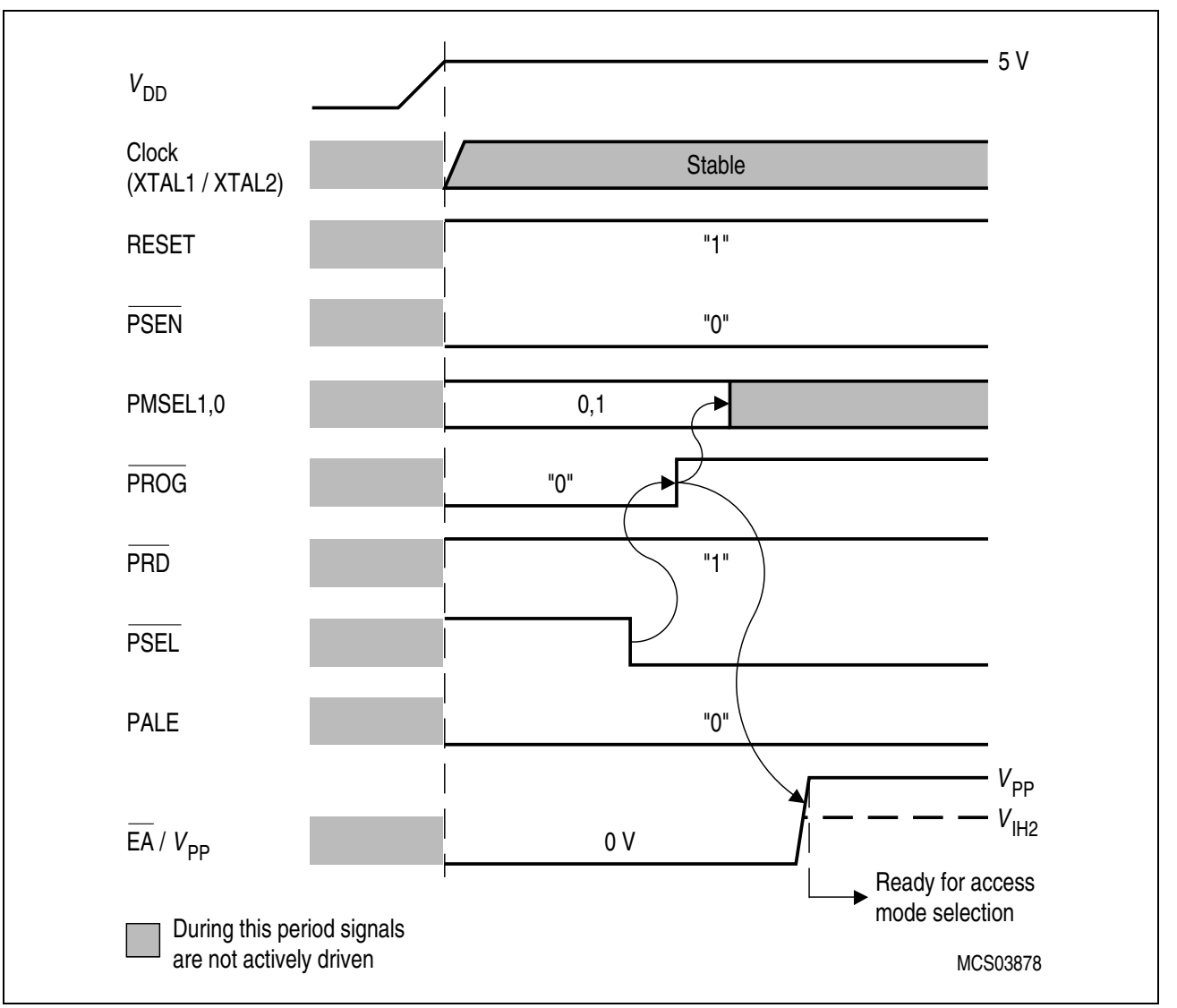


Figure 10-3
Basic Programming Mode Selection

The basic programming mode is selected by executing the following steps:

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

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

*Note: If protection level 1 to 3 has been programmed (see **Section 10.6**) and the programming mode has been left, it is no longer possible to enter the programming mode!*

10.4.2 OTP Memory Access Mode Selection

When the C505L has been put into the programming mode using the basic programming mode selection, several access modes of the OTP memory programming interface are available. The conditions for the different control signals of these access modes are listed in **Table 10-3**.

Table 10-3
Access Modes Selection

Access Mode	$\overline{\text{EA}}/V_{PP}$	$\overline{\text{PROG}}$	$\overline{\text{PRD}}$	PMSEL		Address (Port 2)	Data (Port 0)
				1	0		
Program OTP memory byte	V_{PP}	⊓	H	H	H	A0-7 A8-14	D0-7
Read OTP memory byte	V_{IH}	H	⊓				
Program OTP lock-bits	V_{PP}	⊓	H	H	L	–	D1, D0 see **Table 10-4**
Read OTP lock-bits	V_{IH}	H	⊓				
Read OTP version byte	V_{IH}	H	⊓	L	H	Byte addr. of version byte	D0-7

The access modes shown above are selected by setting the two PMSEL1,0 lines to the required logic level. The $\overline{\text{PROG}}$ and $\overline{\text{PRD}}$ signal are the write and read strobe signal. Data is transferred via port 0 and addresses are applied to port 2.

The following sections describe the various access modes.

10.5 Program/Read OTP Memory Bytes

The program/read OTP memory byte access mode is defined by PMSEL1,0 = 1,1. It is initiated when the PMSEL1,0 = 1,1 is valid at the rising edge of PALE. With the falling edge of PALE, the upper addresses A8-A14 of the 15-bit OTP memory address are latched. After A8-A14 has been latched, A0-A7 is put on the address bus (port 2). A0-A7 must be stable when $\overline{\text{PROG}}$ is low or $\overline{\text{PRD}}$ is low. If subsequent OTP address locations are accessed with constant address information at the high address lines A8-A14, A8-A14 must be latched only once (page address mechanism).

Figure 10-4 shows a typical basic OTP memory programming cycle with a subsequent OTP memory read operation. In this example A8-A14 of the read operation are identical to A8-A14 of the proceeding programming operation.

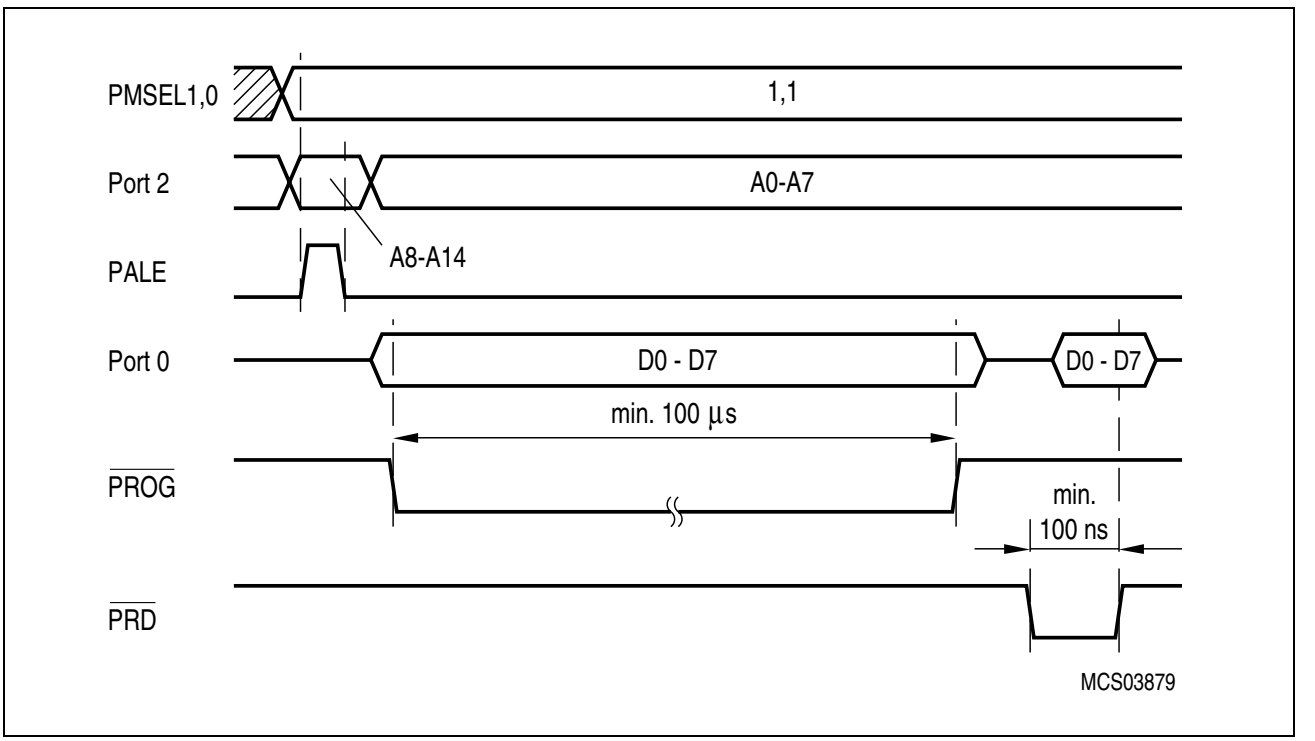


Figure 10-4
Programming/Verify OTP Memory Access Waveform

If the address lines A8-A14 must be updated, PALE must be activated to latch the new A8-A14 value. Control, address, and data information must only be switched when the $\overline{\text{PROG}}$ and $\overline{\text{PRD}}$ signals are at a high level. The PALE high pulse must always be executed if a different access mode has been used prior to the actual access mode.

Figure 10-5 shows a waveform example of the program/read mode access for several OTP memory bytes. In this example, OTP memory locations $3FD_H$ to 400_H are programmed. Thereafter, OTP memory locations 400_H and $3FD_H$ are read.

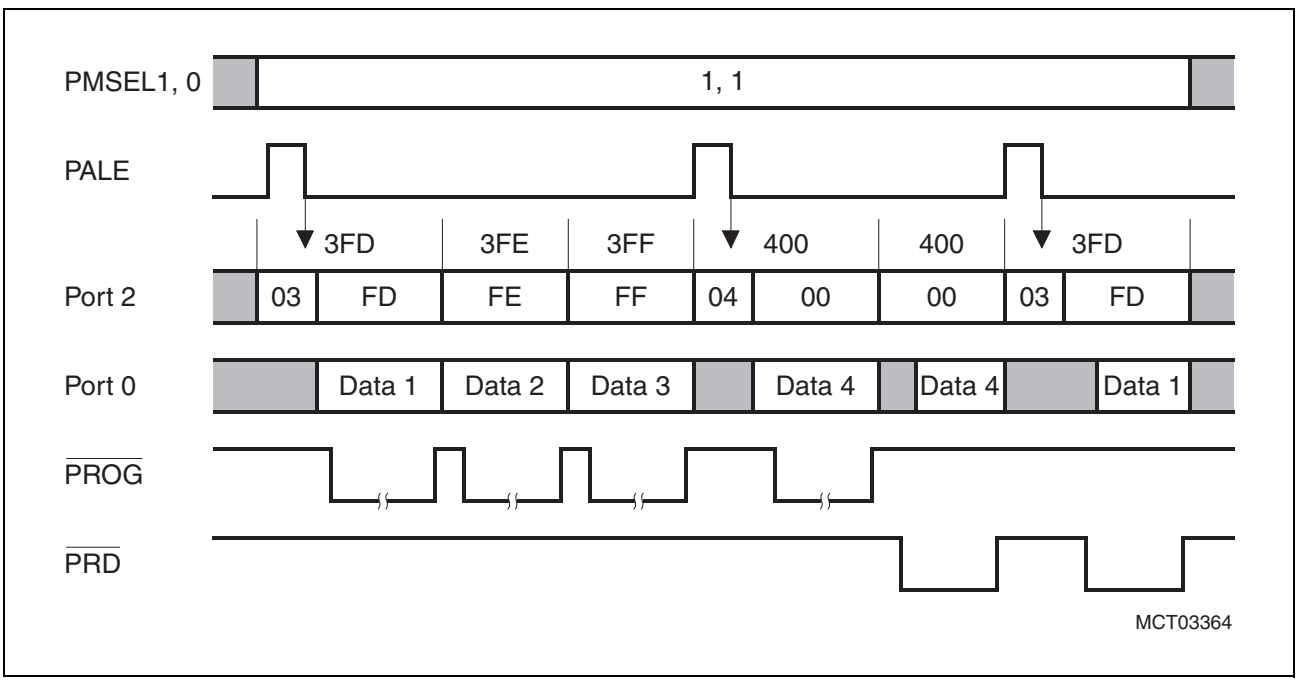


Figure 10-5
Typical OTP Memory Programming/Verify Access Waveform

10.6 Programming and Reading Lock Bits

The C505L has two programmable lock-bits that, when programmed according to **Table 10-4**, provide four levels of protection for the on-chip OTP code memory.

Table 10-4
Lock Bit Protection Types

Lock Bits at D1, D0		Protection Level	Protection Type
D1	D0		
1	1	Level 0	The OTP lock feature is disabled. During normal operation of the C505L, the state of the $\overline{EA}$ pin is not latched on reset.
1	0	Level 1	During normal operation of the C505L, MOVC instructions executed from external program memory are prevented from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset. An OTP memory read operation is only possible in the OTP verification mode. Further programming of the OTP memory is disabled (reprogramming security).
0	1	Level 2	Same as level 1, but OTP memory read operation using OTP verification mode is disabled.
0	0	Level 3	Same as level 2, but external code execution by setting $\overline{EA}$ = low during normal operation of the C505L is not possible. External code execution, which is initiated by an internal program (e.g. by an internal jump instruction above the OTP memory boundary), is still possible.

Note: A “1” means that the lock-bit is unprogrammed. “0” means that lock-bit is programmed.

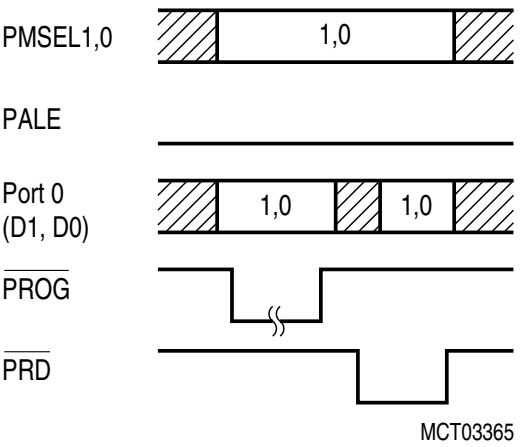
For an OTP verify operation at protection level 1, the C505L must be put into the OTP verification mode.

If a device is programmed with protection level 2 or 3, it is no longer possible to verify the OTP content of a customer rejected (FAR) OTP device.

When a protection level has been activated by programming the lock-bits, the basic programming mode must be exited in order to activate the protection mechanisms. This means that after the activation of a protection level, further OTP program/verify operations are still possible if the basic programming mode is maintained.

The state of the lock-bits can always be read if protection level 0 is selected. If protection level 1 to 3 has been programmed and the programming mode has been exited, it is not possible to re-enter the programming mode. In this case, the lock-bits cannot be read anymore.

Figure 10-6 shows the waveform of a lock-bit write/read access. For a simple drawing, the $\overline{PROG}$ pulse is shortened. In practice, a 100 µs $\overline{PROG}$ low pulse must be applied for lock-bit programming.



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

Figure 10-6
Write/Read Lock Bit Waveform

10.6.1 Access of Version Bytes

The C505L provides 3 version bytes at address locations FC_H, FD_H, and FE_H. The information stored in the version bytes, is defined by the mask of each microcontroller step. Therefore, the version bytes can be read but not written. The three version registers hold information such as manufacturer's code, device type, and stepping code.

To read the version bytes the control lines must be used according to **Table 10-3** and **Figure 10-7**. The address of the version byte must be applied to the port 2 address lines. PALE must not be activated.

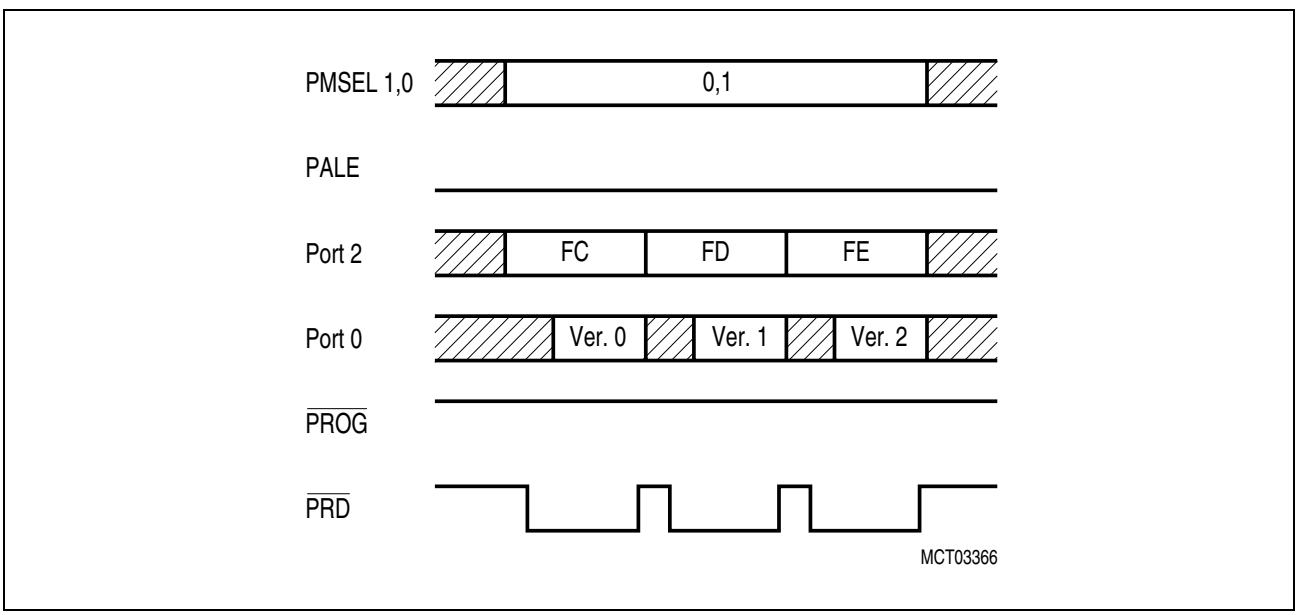


Figure 10-7
Read Version Register(s) Waveform

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

Note: The 3 version bytes are implemented in a way that allows them be read during normal program execution mode as a mapped register with bit RMAP in Special Function register (SFR) SYSCON set. The addresses of the version bytes in normal mode and programming mode are identical and therefore they are located in the SFR address range.

The steppings of the C505L versions will contain the following version register/byte information:

Stepping	Version Byte 0 = VR0 (mapped addr. FC_H)	Version Byte 1 = VR1 (mapped addr. FD_H)	Version Byte 2 = VR2 (mapped addr. FE_H)
C505L ES-AA-Step	$C5_H$	85_H	01_H

Note: Future steppings of C505L would have a different version byte 2 content.

10.7 OTP Verification Mode

The OTP verification mode shown in **Figure 10-8** is used to verify the contents of the OTP when the protection level 1 has been set. The detailed timing characteristics of the OTP verification mode are shown in the AC specifications (refer to Data Sheet).

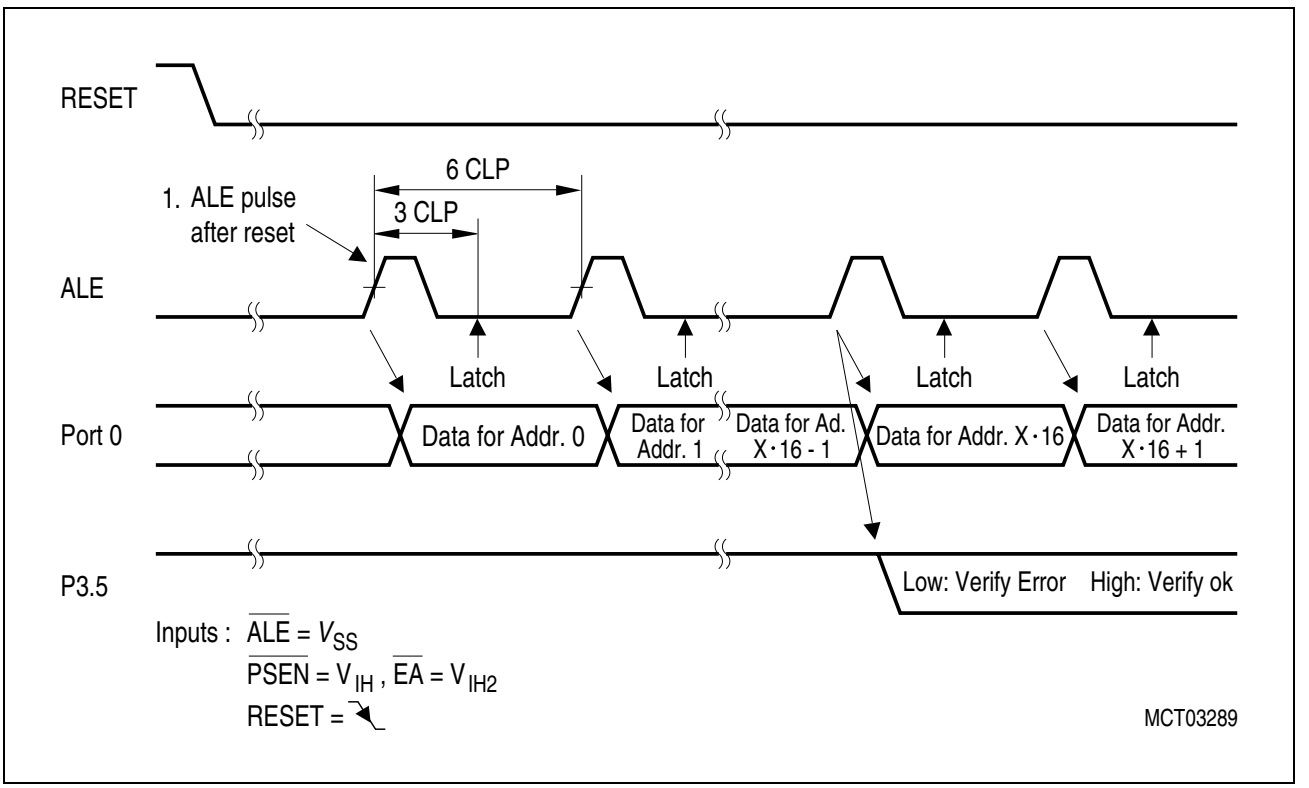


Figure 10-8
OTP Verification Mode

OTP verification mode is selected if the inputs $\overline{PSEN}$, $\overline{EA}$, and ALE are set at the specified logic levels. With RESET going inactive, the OTP verification mode sequence is started. The C505L outputs an ALE signal with a period of 3 clock periods (CLP) and expects data bytes at port 0. The data bytes at port 0 are assigned to the OTP addresses in the following way:

1. Data Byte = content of OTP address 0000_H
2. Data Byte = content of OTP address 0001_H
3. Data Byte = content of OTP address 0002_H
:
16. Data Byte = content of OTP address $000F_H$
:

The C505L does not output any address information during the OTP verification mode. The first data byte to be verified is always the byte that is assigned to the OTP address 0000_H, and it must be put onto the data bus with the falling edge of RESET. With each following ALE pulse, the OTP address pointer is internally incremented and the expected data byte for the next OTP address must be delivered externally.

Between two ALE pulses the data at port 0 is latched (at 3 CLP after ALE rising edge) and compared internally with the OTP content of the actual address. If a verify error is detected, the error condition is stored internally. After each 16th data byte, the cumulated verify result (pass or fail) of the last 16 verify operations is output at P3.5. This means that P3.5 stays at static level (low for fail and high for pass) during the time when the subsequent 16 bytes are checked. In OTP verification mode, the C505L must be provided with a system clock at the XTAL pins.

Figure 10-9 shows an application example of an external circuitry that allows verification of the OTP, with protection level 1, inside the C505L in the OTP verification mode. When RESET goes inactive, the C505L starts the OTP verify sequence. Its ALE is clocking a 15-bit address counter. This counter generates the addresses for an external EPROM that is programmed with the contents of the OTP. The verify detect logic typically displays the pass/fail information of the verify operation. P3.5 can be latched with the falling edge of ALE.

When the last byte of the OTP has been handled, the C505L starts generating a $\overline{\text{PSEN}}$ signal. This signal or the CY signal of the address counter indicate to the verify detect logic the end of the OTP verification.

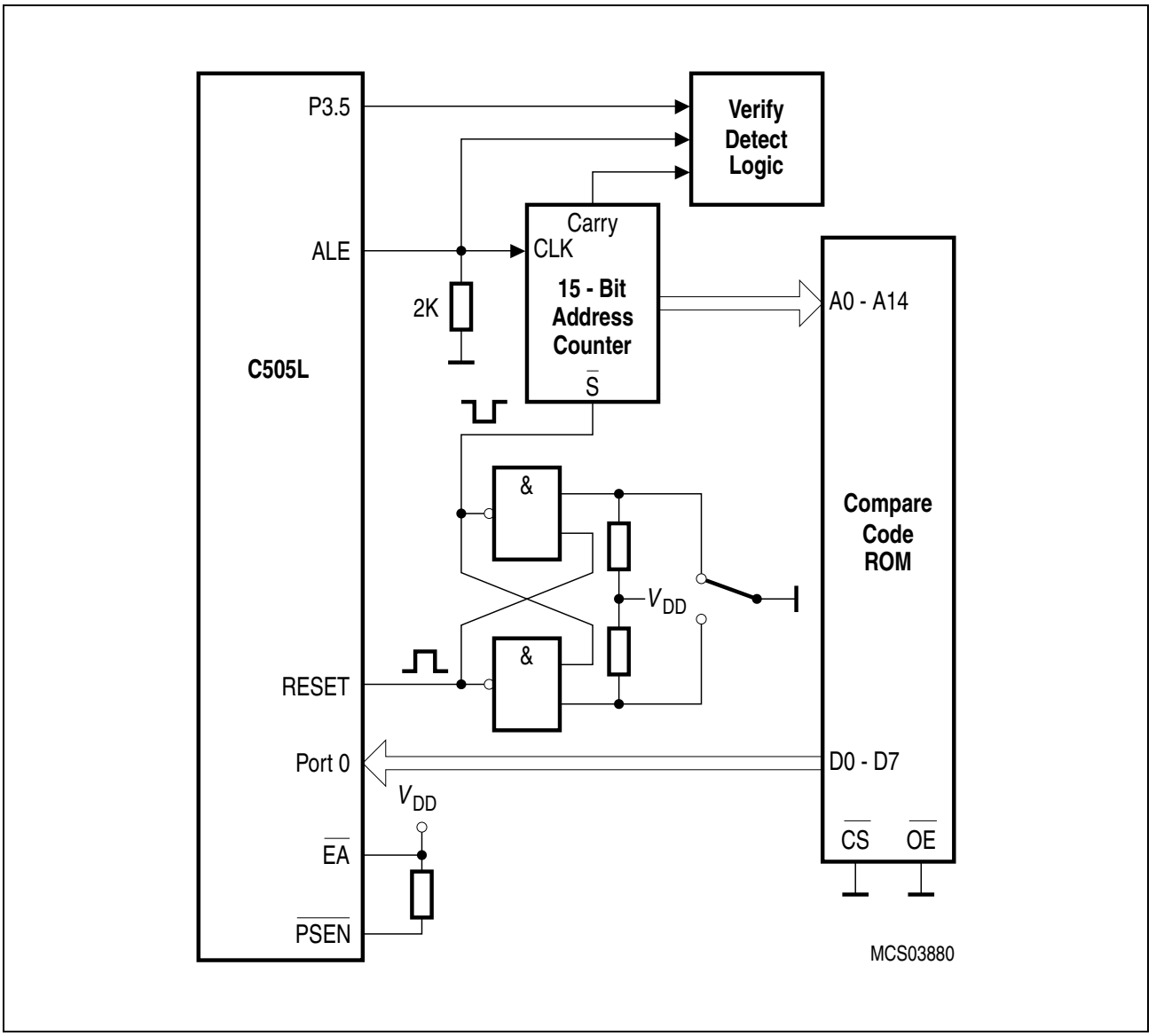


Figure 10-9
OTP Verification Mode - External Circuitry Example

11 Index

X

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG