User's Manual, May 2001

C508

8-Bit CMOS Microcontroller

Microcontrollers

Never stop thinking.

Edition 2001-05

**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, May 2001

C508

8-Bit CMOS Microcontroller

Microcontrollers

Infineon technologies

Never stop thinking.

C508 User's Manual
Revision History: 2001-05

Previous Version: 1999-10

Page	Subjects (major changes since last revision)
3-14, 3-15, 8-2	Reset value of SFR WDTH is corrected.
6-84	Block diagram of the Combined Multi-Channel PWM Modes is updated.
–	“Phase Delay Timer” - related description is removed.

We Listen to Your Comments

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

mcdocu.comments@infineon.com

Table of Contents Page

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

1 Introduction

The C508 is a member of the Infineon Technologies C500 family of 8-bit microcontrollers. It is fully compatible to the standard 8051 microcontroller. Its features include extended power saving provisions, 256 × 8 on-chip RAM, 32K × 8 on-chip program memory, RFI related improvements, and the Capture Compare Unit (CCU) which is useful in motor control applications. The C508 has an internal PLL and, with a maximum CPU clock rate of 20 MHz, it achieves a 300 ns instruction cycle time.

The C508 operates with internal and/or external program memory. The C508-4R contains 32K × 8 on-chip program memory (ROM version) while the C508-4E has 32K × 8 One-Time Programmable program memory (OTP version). In this document, the term C508 refers to both versions unless otherwise noted.

Figure 1-1 shows the various functional units of the C508 and **Figure 1-2** shows the simplified logic symbol of the C508.

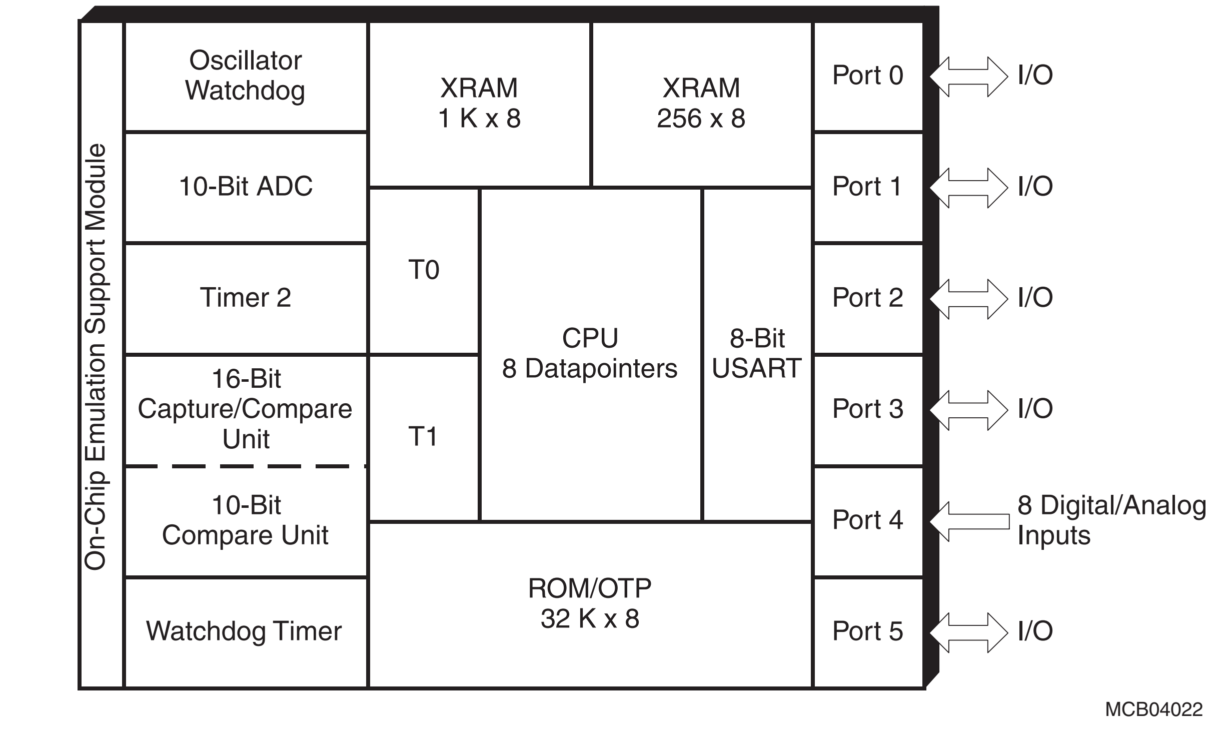


Figure 1-1 C508 Functional Units

Listed below is a summary of the main features of the C508 microcontroller:

- Fully compatible to standard 8051 microcontroller
- Superset of the 8051 architecture with eight datapointers
- 10 to 20 MHz internal CPU clock (using built-in PLL with a factor of 2)
  - external clock of 5 - 10 MHz at 50% duty cycle
  - 300 ns instruction cycle time at 20 MHz CPU clock
- 32 Kbytes on-chip ROM/OTP (with optional ROM protection)
- 256 bytes on-chip RAM
- 1024 bytes on-chip XRAM
- Six 8-bit ports,
  - Ports 1 and 2 with enhanced current sinking capabilities of 10 mA per pin (total max. of 100 mA)
  - Port 4 with pure analog/digital input channels
- Three 16-bit timers/counters
  - Timer 0 /1 (C501 compatible)
  - Timer 2 with four channels for 16-bit capture/compare operation
- Capture/Compare Unit (CCU) for PWM (Pulse Width Modulation) signal generation
  - 3-channel, 16-bit capture/compare unit
  - 1-channel, 10-bit compare unit
- Full-duplex serial interface with programmable baudrate generator (USART)
- 8-channel 10-bit A/D Converter
- 19 interrupt vectors with four priority levels
- On-chip emulation support logic (Enhanced Hooks Technology)
- 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)
  - Software power-down mode with wake up capability through P3.2/$\overline{\text{INT0}}$ or P5.7/INT7
- ALE switch-off capability for reduction in RFI emission
- P-MQFP-64-1, P-SDIP-64-2 packages
- Temperature ranges: SAB-C508 T_A = 0 to 70 °C
  SAF-C508 T_A = -40 to 85 °C

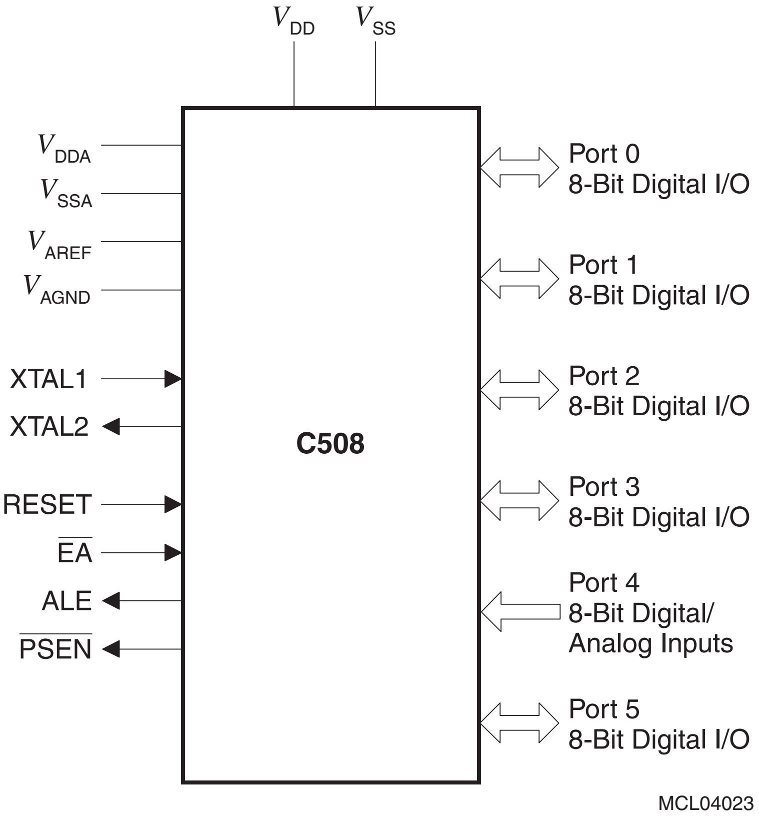


Figure 1-2 Logic Symbol

1.1 Pin Configuration

This section shows the pin configurations of the C508 microcontroller in the P-MQFP-64-1 and the P-SDIP-64-2 packages.

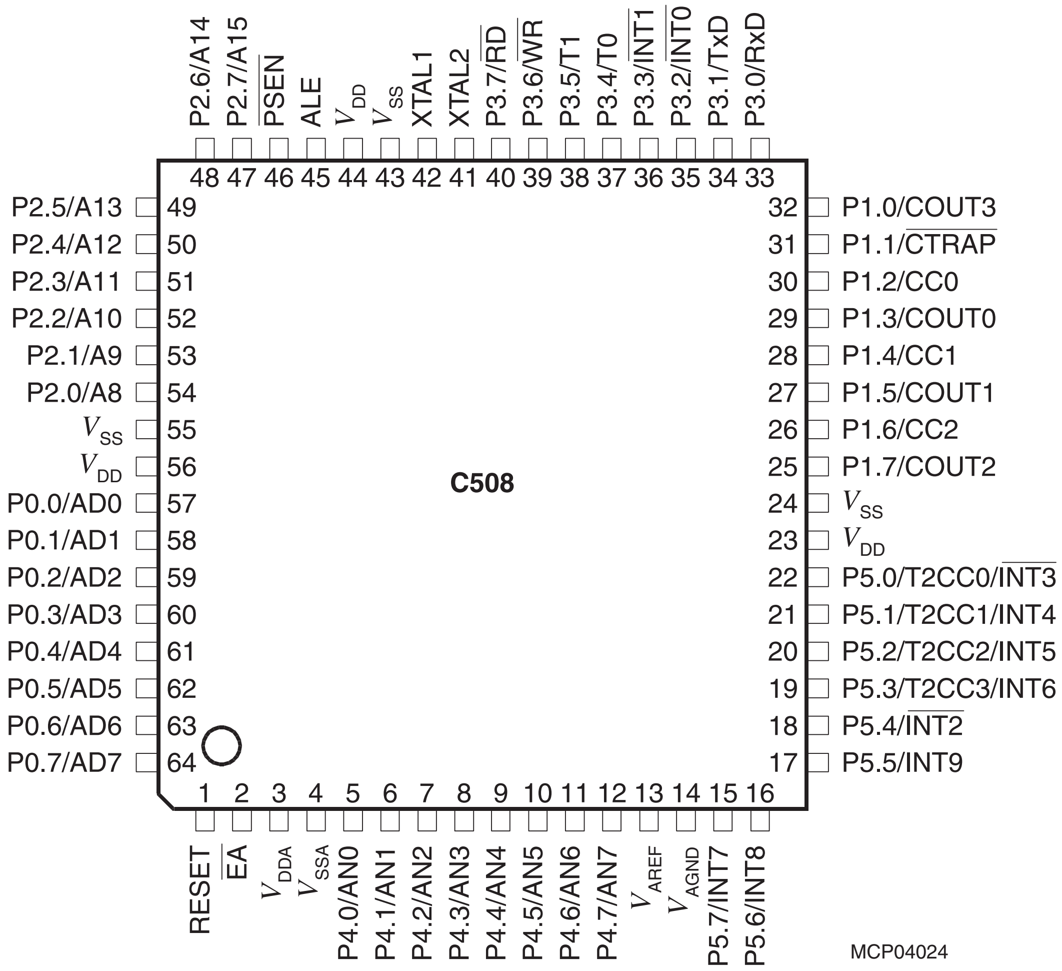


Figure 1-3 Pin Configuration for P-MQFP-64-1 Package (top view)

Figure 1-4 shows the pin configuration of the C508 in the P-SDIP-64-2 package.

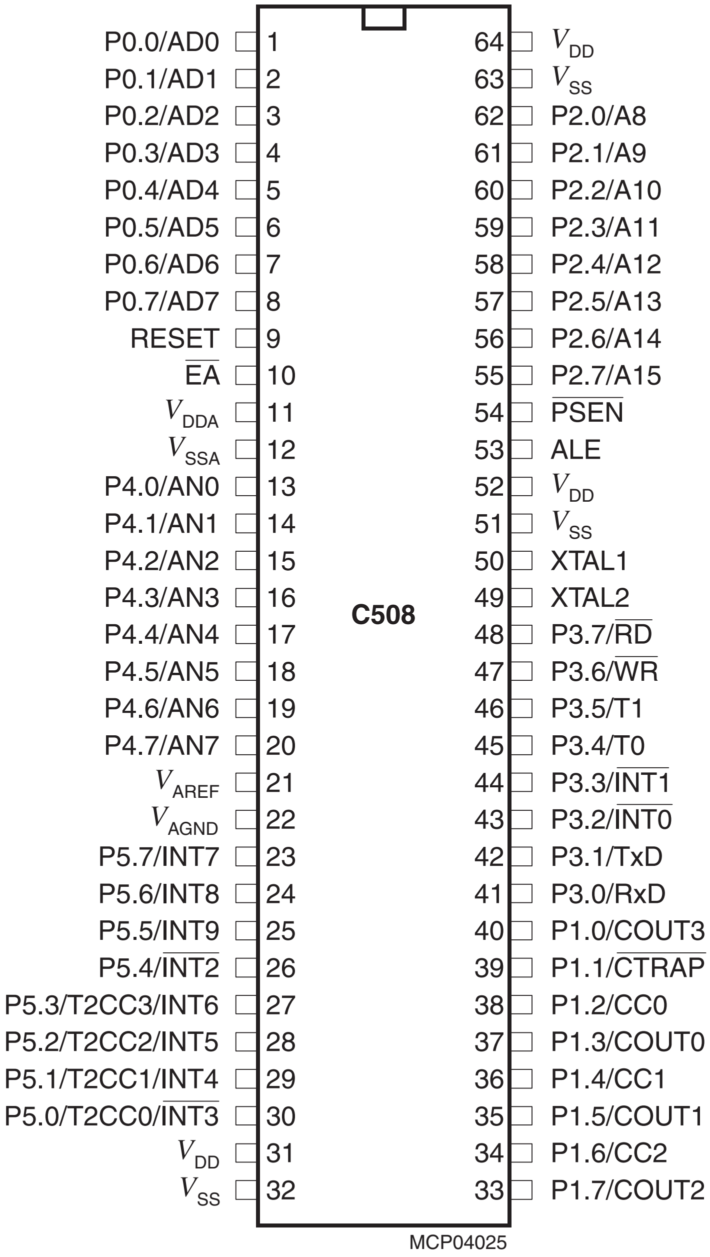


Figure 1-4 Pin Configuration for P-SDIP-64-2 Package (top view)

1.2 Pin Definitions and Functions

This section describes all external signals of the C508 and their functions.

Table 1-1 Pin Definitions and Functions

Symbol	**Pin Numbers**		**I/O**[1]	**Function**
	P-MQFP-64	**P-SDIP-64**		
P1.0-P1.7	32 - 25	40 - 33	I/O	Port 1 is an 8-bit quasi-bidirectional port with internal pull-up transistors. Port 1 pins can be used for digital input/output. Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and in that state can be used as inputs. As inputs, Port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up transistors. The output latch corresponding secondary function must be programmed to a one (1) for that function to operate. As secondary functions, Port 1 contains the capture/compare inputs/outputs as well as the CCU trap input. Port 1 pins have LED drive capability of up to 10 mA sinking current per pin. The secondary functions from the CCU unit are assigned to the pins of Port 1 as follows:
	32	40		P1.0 / COUT3 10-bit compare channel output
	31	39		P1.1 / $\overline{\text{CTRAP}}$ CCU trap input
	30	38		P1.2 / CC0 Input/Output of capture/compare Channel 0
	29	37		P1.3 / COUT0 Output of capture/compare Channel 0
	28	36		P1.4 / CC1 Input/Output of capture/compare Channel 1
	27	35		P1.5 / COUT1 Output of capture/compare Channel 1
	26	34		P1.6 / CC2 Input/Output of capture/compare Channel 2
	25	33		P1.7 / COUT2 Output of capture/compare Channel 2
RESET	1	9	I	**RESET** A high level on this pin for one machine cycle 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}.

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

Symbol	Pin Numbers		I/O[1]	Function
	P-MQFP-64	P-SDIP-64		
P3.0-P3.7	33 - 40	41 - 48	I/O	**Port 3** is an 8-bit quasi-bidirectional port with internal pull-up transistors. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and in that state can be used as inputs. As inputs, Port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up transistors. The output latch corresponding secondary function must be programmed to a one (1) for that function to operate (except for TxD and $\overline{\text{WR}}$). The secondary functions are assigned to the pins of Port 3 as follows:
	33	41		P3.0 / RxD Receiver data input (asynch.) or data input/output (synch.) of serial interface
	34	42		P3.1 / TxD Transmitter data output (asynch.) or clock output (synch.) of serial interface
	35	43		P3.2 / $\overline{\text{INT0}}$ External Interrupt 0 Input/Timer 0 gate control input
	36	44		P3.3 / $\overline{\text{INT1}}$ External Interrupt 1 Input/Timer 1 gate control input
	37	45		P3.4 / T0 Timer 0 counter input
	38	46		P3.5 / T1 Timer 1 counter input
	39	47		P3.6 / $\overline{\text{WR}}$ $\overline{\text{WR}}$ control output; latches the data byte from Port 0 into the external data memory
	40	48		P3.7 / $\overline{\text{RD}}$ $\overline{\text{RD}}$ control output; enables the external data memory

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

Symbol	Pin Numbers		I/O[1)]	Function
	P-MQFP-64	P-SDIP-64		
P2.0-P2.7	54 - 47	62 - 55	I/O	**Port 2** is an 8-bit quasi-bidirectional I/O port with internal pull-up transistors. Port 2 pins that have 1's written to them are pulled high by the internal pull-up 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 pull-up transistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-up transistors when issuing 1's. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), Port 2 issues the contents of the P2 special function register and uses only the internal pull-up transistors. As I/O functions, Port 2 pins also have LED drive capability of up to 10 mA sinking current per pin.
XTAL1	42	50	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. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
XTAL2	41	49	O	**XTAL2** Output of the inverting oscillator amplifier.
P4.0-P4.7	5 - 12	13 - 20	I	**Port 4** is an 8-bit uni-directional input port to the A/D converter. Port 4 pins can be used for digital input, if voltage levels simultaneously meet the specifications for high/low input voltages and for the eight multiplexed analog inputs.

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

Symbol	Pin Numbers		I/O[1]	Function
	P-MQFP-64	P-SDIP-64		
$\overline{\text{PSEN}}$	46	54	O	The $\overline{\textbf{Program Strobe Enable}}$ output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every one and a half 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	53	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 one and a half oscillator periods except during an external data memory access. When instructions are executed from internal ROM ($\overline{\text{EA}}$ = 1) the ALE generation can be disabled by bit EALE in SFR SYSCON. This pin should not be driven during reset operation.
$\overline{\text{EA}}$	2	10	I	$\overline{\textbf{External Access Enable}}$ When held at high level, instructions are fetched from the internal ROM when the PC is less than 8000_H. When held at low level, the C508 fetches all instructions from external program memory. This pin should not be driven during reset operation.
P0.0-P0.7	57 - 64	1 - 8	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory. In this application, it uses strong internal pull-up transistors when issuing 1's. Port 0 also outputs the code bytes during program verification in the C508-4R. External pull-up resistors are required during program verification.

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

Symbol	Pin Numbers		I/O[1]	Function
	P-MQFP-64	P-SDIP-64		
P5.0-P5.7	22 - 15	30 - 23	I/O	**Port 5** is a an 8-bit quasi-bidirectional I/O port with internal pull-up transistors. Port 5 pins that have 1's written to them are pulled high by the internal pull-up resistors, 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 pull-up transistors. As secondary functions, Port 5 contains the interrupt and Timer 2 capture/compare pins. They are assigned to the pins as follows:
	22	30		P5.0/T2CC0/$\overline{\text{INT3}}$ T2 Compare/Capture output 0/ Interrupt 3 input
	21	29		P5.1/T2CC1/INT4 T2 Compare/Capture output 1/ Interrupt 4 input
	20	28		P5.2/T2CC2/INT5 T2 Compare/Capture output 2/ Interrupt 5 input
	19	27		P5.3/T2CC3/INT6 T2 Compare/Capture output 3/ Interrupt 6 input
	18	26		P5.4/$\overline{\text{INT2}}$ Interrupt 2 input
	17	25		P5.5/INT9 Interrupt 9 input
	16	24		P5.6/INT8 Interrupt 8 input
	15	23		P5.7/INT7 Interrupt 7 input
V_{SS}	24, 43, 55	32, 51, 63	–	**Ground** (0 V)
V_{DD}	23, 44, 56	31, 52, 64	–	**Power Supply** (+5 V)
V_{DDA}	3	11	–	**Analog Power Supply** (+5 V)
V_{SSA}	4	12	–	**Analog Ground** (0 V)
V_{AREF}	13	21	–	**Reference Voltage** for the A/D Converter.
V_{AGND}	14	22	–	**Reference Ground** for the A/D Converter.

[1] I = Input
O = Output

2 Fundamental Structure

The C508 is fully compatible with the architecture of the standard 8051/C501 microcontroller family. While maintaining all architectural and operational characteristics of the C501, the C508 incorporates a CPU with eight datapointers, a 10-bit A/D Converter, a 16-bit Capture/Compare Unit, a Timer 2 with capture/compare functions, an improved interrupt structure with four priority levels, built-in PLL with a fixed factor of 2, and an XRAM data memory, as well as some enhancements in the Fail Save Mechanism Unit. **Figure 2-1** shows a block diagram of the C508 microcontroller.

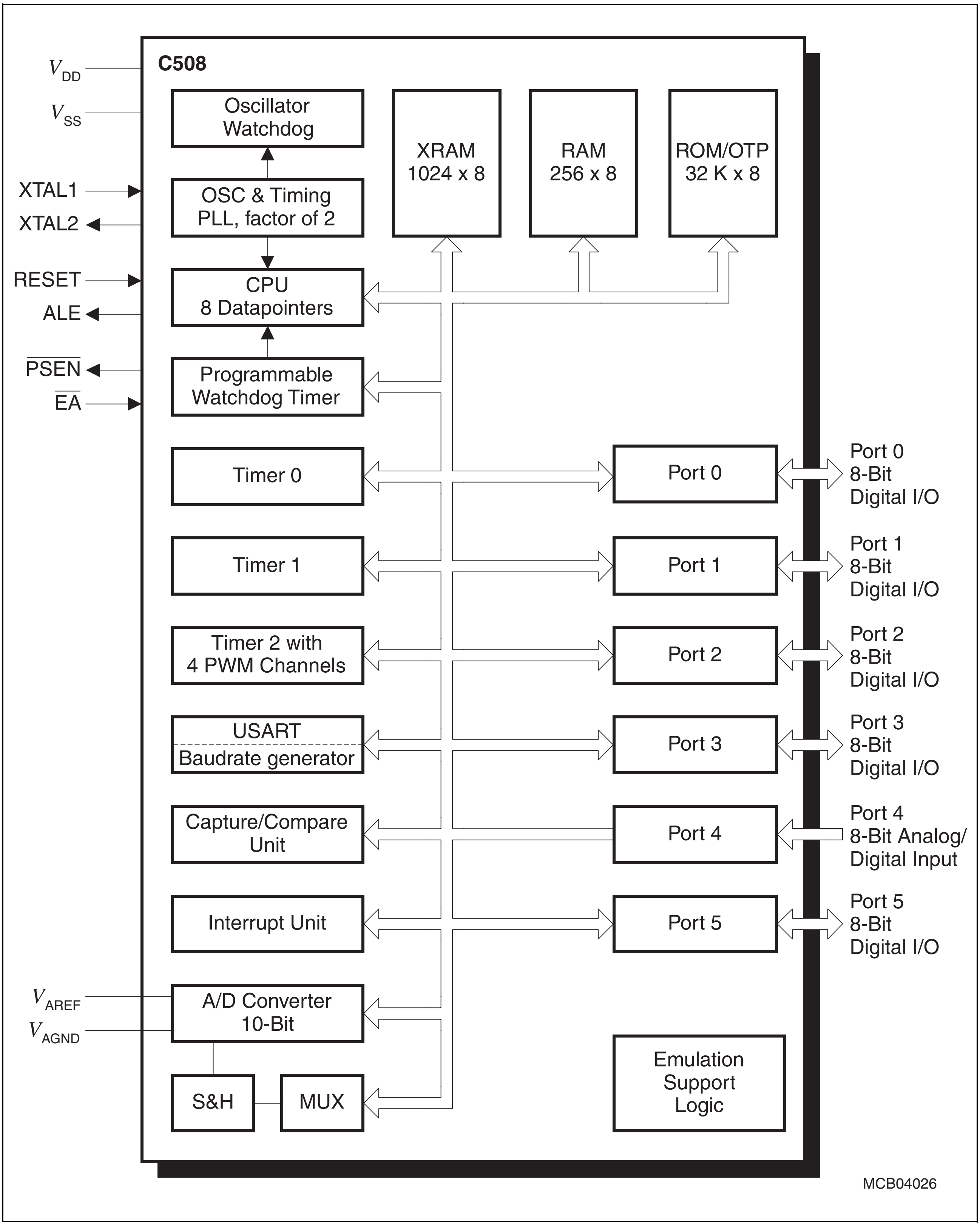


Figure 2-1 Block Diagram of the C508

2.1 CPU

The C508 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 10 MHz external crystal (giving a 20 MHz CPU clock), 58% of the instructions execute in 300 ns. For an 8 MHz crystal, the corresponding time is 375 ns.

The Central Processing Unit (CPU) of the C508 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. These internal signals have an effect on the source and destination of data transfers and control the Arithmetic/Logic Unit (ALU) processing.

The arithmetic section of the processor performs extensive data manipulation and is comprised of the ALU, an A register, a 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, substract, multiply, divide, increment, decrement, BDC-decimal-add-adjust and compare, and the logic operations AND, OR, Exclusive OR, complement, and rotate (right, left or swap nibble (left four)). Also included is a Boolean processor performing the bit operations such 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, the ALU 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.

Beyond the CPU functionality of the C501/8051 standard microcontroller, the C508 contains eight datapointers. For complex applications with peripherals located in the external data memory space or extended data storage capacity, this proved to be a “bottle neck” for the 8051’s communication to the external world. Programming in high-level languages (PLM51, C51, PASCAL51) especially requires extended RAM capacity as well as fast access to this additional RAM because of the reduced code efficiency of these languages.

Accumulator

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

Program Status Word

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

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

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

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>CY</td><td><b>Carry Flag</b>
Used by arithmetic instructions.</td></tr>
<tr><td>AC</td><td><b>Auxiliary Carry Flag</b>
Used by instructions which execute BCD operations.</td></tr>
<tr><td>F0</td><td><b>General Purpose Flag 0</b></td></tr>
<tr><td>RS1
RS0</td><td><b>Register Bank select control bits</b>
These bits are used to select one of the four register banks.
<table>
<tr><th>RS1</th><th>RS0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Bank 0 selected, data address 00_H-07_H</td></tr>
<tr><td>0</td><td>1</td><td>Bank 1 selected, data address 08_H-$0F_H$</td></tr>
<tr><td>1</td><td>0</td><td>Bank 2 selected, data address 10_H-17_H</td></tr>
<tr><td>1</td><td>1</td><td>Bank 3 selected, data address 18_H-$1F_H$</td></tr>
</table></td></tr>
<tr><td>OV</td><td><b>Overflow Flag</b>
Used by arithmetic instructions.</td></tr>
<tr><td>F1</td><td><b>General Purpose Flag 1</b></td></tr>
<tr><td>P</td><td><b>Parity Flag</b>
Set/cleared by hardware after each instruction to indicate an odd/even number of “one” bits in the accumulator.</td></tr>
</table>

B Register

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

Stack Pointer

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

2.2 CPU Timing

The C508 has no clock prescaler. Therefore, a machine cycle of the C508 consists of six states (3 oscillator periods). Each state is divided into a Phase 1 half and a Phase 2 half. Thus, a machine cycle consists of 3 oscillator periods, numbered S1P1 (State 1, Phase 1) through S6P2 (State 6, Phase 2). Each state lasts for half an 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 Address Latch Enable (ALE) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

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

Figure 2-2 **(a)** and **(b)** show the timings for a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most C508 instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete: they take four cycles. Normally, two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed. **Figure 2-2** **(c)** and **(d)** show the timings for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

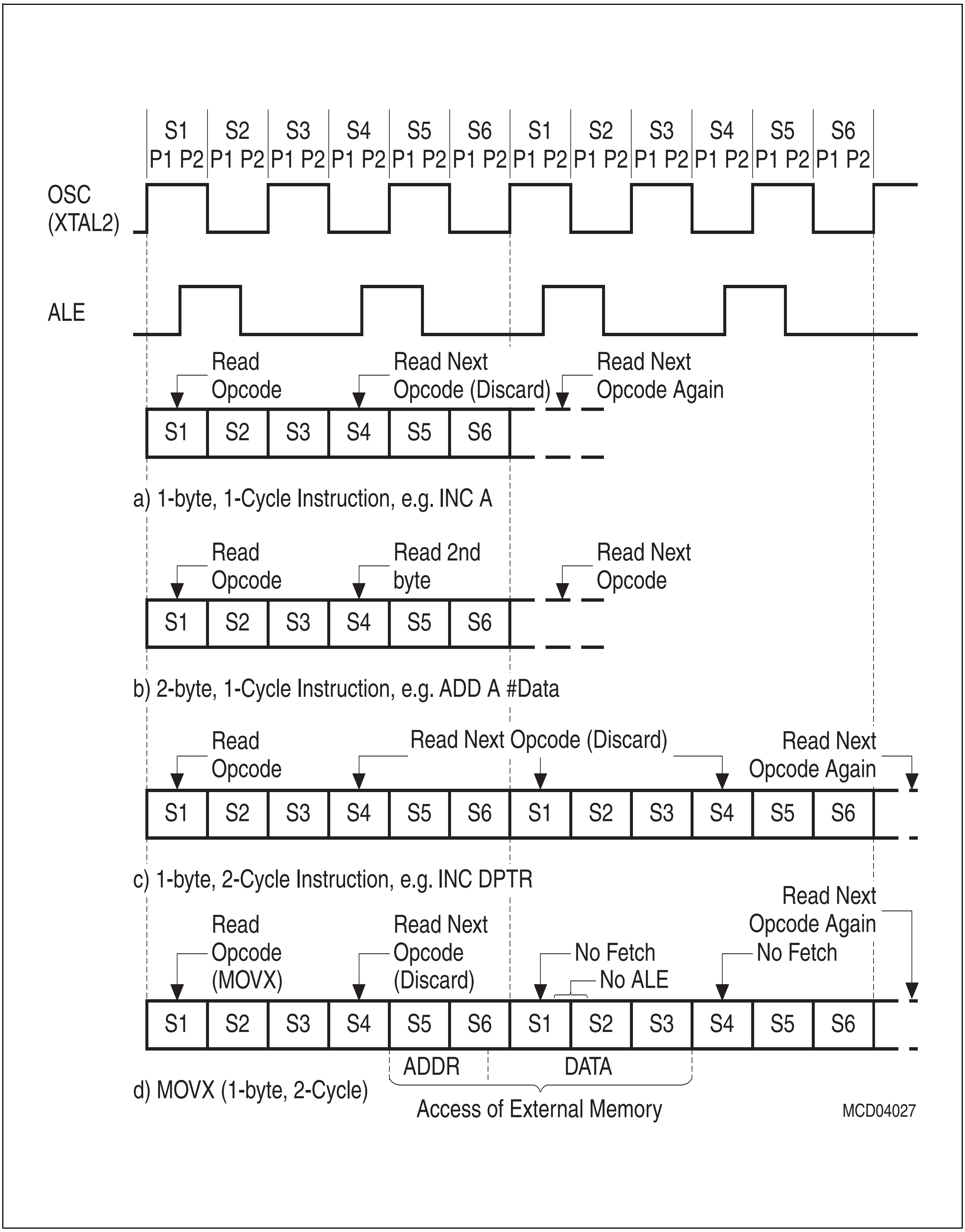

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6
P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2
OSC (XTAL2)
ALE
Read Opcode
Read Next Opcode (Discard)
Read Next Opcode Again
a) 1-byte, 1-Cycle Instruction, e.g. INC A
Read Opcode
Read 2nd byte
Read Next Opcode
b) 2-byte, 1-Cycle Instruction, e.g. ADD A #Data
Read Opcode
Read Next Opcode (Discard)
Read Next Opcode Again
c) 1-byte, 2-Cycle Instruction, e.g. INC DPTR
Read Opcode (MOVX)
Read Next Opcode (Discard)
No Fetch
No ALE
Read Next Opcode Again
No Fetch
ADDR
DATA
Access of External Memory
d) MOVX (1-byte, 2-Cycle)
MCD04027

Figure 2-2 Fetch Execute Sequence

3 Memory Organization

The C508 CPU manipulates operands in the following five address spaces:

- Up to 64 Kbytes of program memory: 32K ROM for the C508-4R
  : 32K OTP for the C508-4E
- Up to 64 Kbytes of external data memory
- 256 bytes of internal data memory
- 1024 bytes of internal XRAM data memory
- a 128-byte special function register area.

Figure 3-1 illustrates the memory address spaces of the C508.

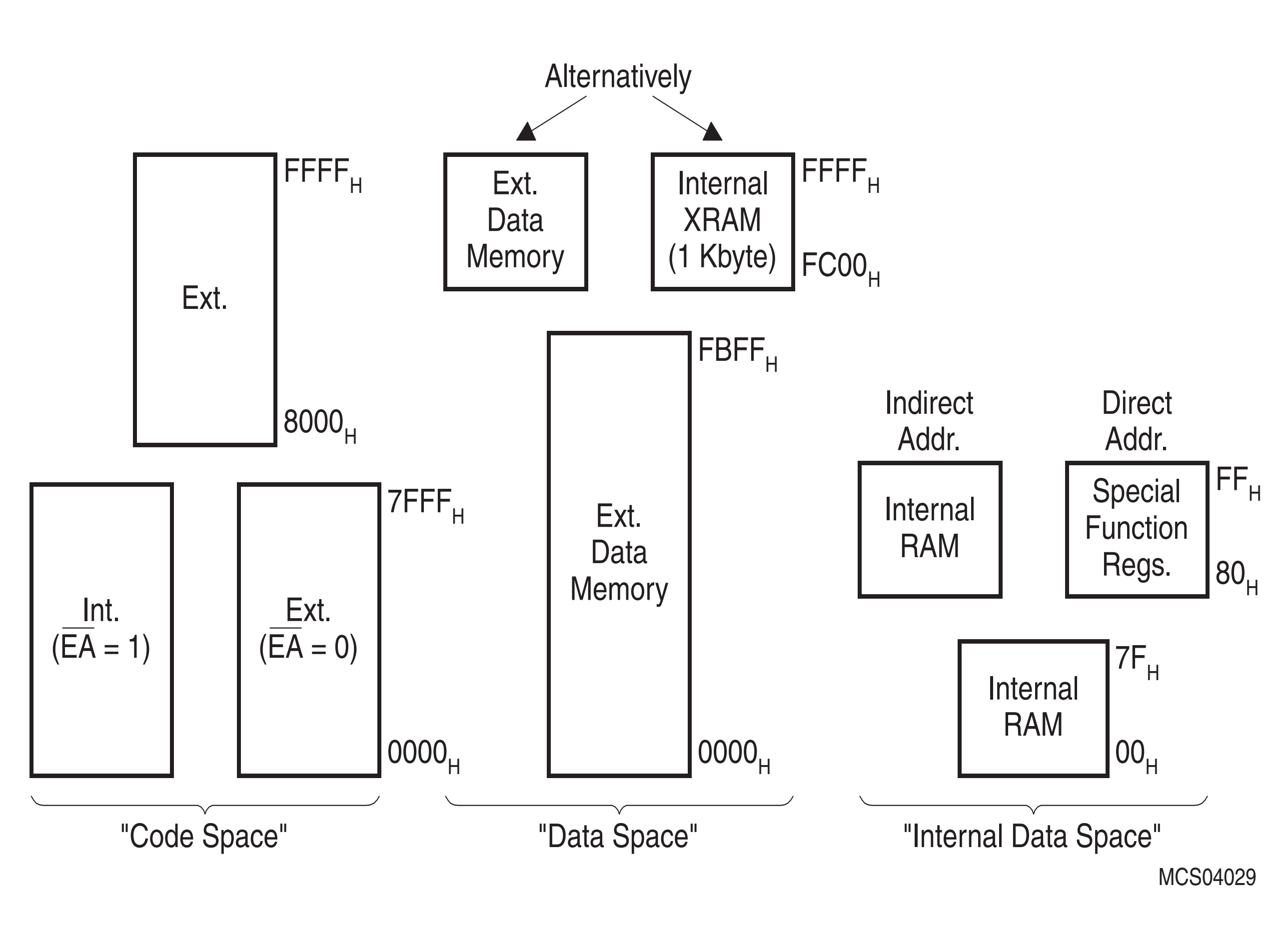


Figure 3-1 C508 Memory Map

3.1 Program Memory, “Code Space”

The C508-4R has 32 Kbytes of Read-Only program Memory (ROM), while the C508-4E provides 32 Kbytes of OTP program memory. The program memory can be externally expanded up to 64 Kbytes. If the $\overline{\text{EA}}$ pin is held high, the C508-4R executes program code out of the internal ROM unless the program counter address exceeds $7FFF_H$. Address locations 8000_H through $FFFF_H$ are then fetched from the external program memory. If the $\overline{\text{EA}}$ pin is held low, the C508 fetches all instructions from the external program memory.

3.2 Data Memory, “Data Space”

The data memory address space consists of an internal and an external memory space. The internal data memory is divided into three physically separate and distinct blocks: the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128-byte Special Function Register (SFR).

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 XRAM is located in the external address memory area at addresses $FC00_H$ to $FFFF_H$. Using MOVX instruction with addresses pointing to this address area, allows access to either internal XRAM or external data RAM.

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 op code indicates which register is to be used. For indirect addressing, R0 and R1 are used as pointers or index registers to address internal or external memory (for example: 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 more than one register bank is required, the SP should be initialized to a different location of the RAM which is not used for data storage.

3.4 XRAM Operation

The XRAM in the C508 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 Controller Access Control

Two bits in SFR SYSCON, XMAP0 and XMAP1, control the accesses to XRAM. XMAP0 is a general access enable/disable control bit and XMAP1 controls the external signal generation during XRAM 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 functions of the shaded bits are not described here.

Bit	Function
XMAP1	**XRAM visible access control** Control bit for $\overline{RD}$/$\overline{WR}$ signals during XRAM accesses. If addresses are outside the XRAM 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. XMAP1 = 1: Ports 0, 2 and the signals $\overline{RD}$ and $\overline{WR}$ are activated during accesses to XRAM. In this mode, address and data information during XRAM accesses are visible externally.
XMAP0	**Global XRAM access enable/disable control** XMAP0 = 0: The access to XRAM is enabled. XMAP0 = 1: The access to XRAM is 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, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ become active and Port 0 and Port 2 drive the actual address/data information which is read/written from/to XRAM. This feature allows checking of the internal data transfers to XRAM. When Ports 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 the accesses to XRAM are generally disabled. In this case, all accesses using MOVX instructions within the address range of FC00_H to FFFF_H generate external data memory bus cycles. When XMAP0 is cleared, the access to XRAM is enabled and all accesses using MOVX instructions with an address in the range of FC00_H to FFFF_H will access the internal XRAM.

Bit XMAP0 is hardware protected. If it is cleared once (that is, if internal XRAM access enabled), it cannot be set by software. Only a reset operation will set the XMAP0 bit again. This hardware protection mechanism is implemented by an asymmetric latch at XMAP0 bit. An unintentional disabling of XRAM could be dangerous as 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, during reset, an internal capacitor is charged. So the reset status is a disabled XRAM. After a ‘0’ is written to XMAP0 bit (that is, discharging the capacitor), it is not possible to set it back again to ‘1’ due to the charge time of the capacitor. On the other hand, any distortion (such as a software hang up, noise, etc.) also cannot charge this capacitor. Thus, the stable status is the enabled XRAM.

The clear instruction for the XMAP0 bit should be integrated into the program initialization routine before XRAM 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 can be accessed by two read/write instructions, which use the 16-bit DPTR for indirect addressing. These instructions are:

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

For accessing the XRAM, the effective address stored in DPTR must be in the range of $FC00_H$ to $FFFF_H$.

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

The 8051 architecture also provides instructions for accesses to the external data memory range which 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 C508 to enable accessing the XRAM with the MOVX @Ri instructions as well; that is, XPAGE serves the same function for the XRAM as Port 2 does 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 high address** XPAGE.7-0 is the address part A15-A8 when 8-bit MOVX instructions are used to access internal XRAM.

Figure 3-2 to **Figure 3-4** show the dependencies of XPAGE and Port 2 addressing in order to illustrate the differences in accessing XRAM, external RAM, or the use of Port 2 as an I/O-port.

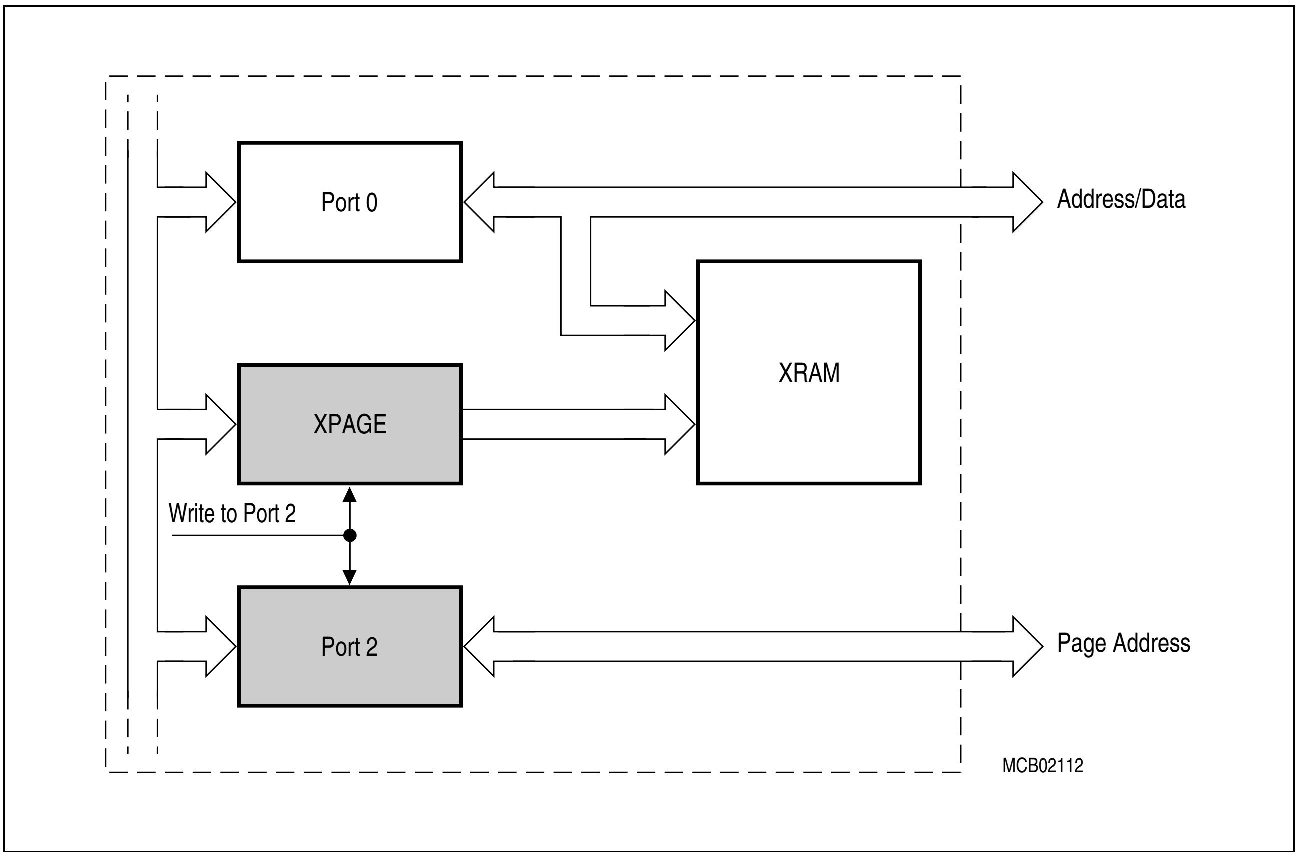


Figure 3-2   Write Page Address to Port 2

Moving the page address to Port 2 by using either the immediate addressing instruction (MOV P2, #pageaddress) or the direct addressing instruction (MOV P2, PAL; where 'PAL' is the internal RAM location containing the page address) will write the page address to Port 2 and also to the XPAGE-Register.

When external RAM is to be accessed in the XRAM address range, the XRAM must be disabled. When the additional external RAM is to be addressed in an address range 0000_H to $FC00_H$, the XRAM may remain enabled.

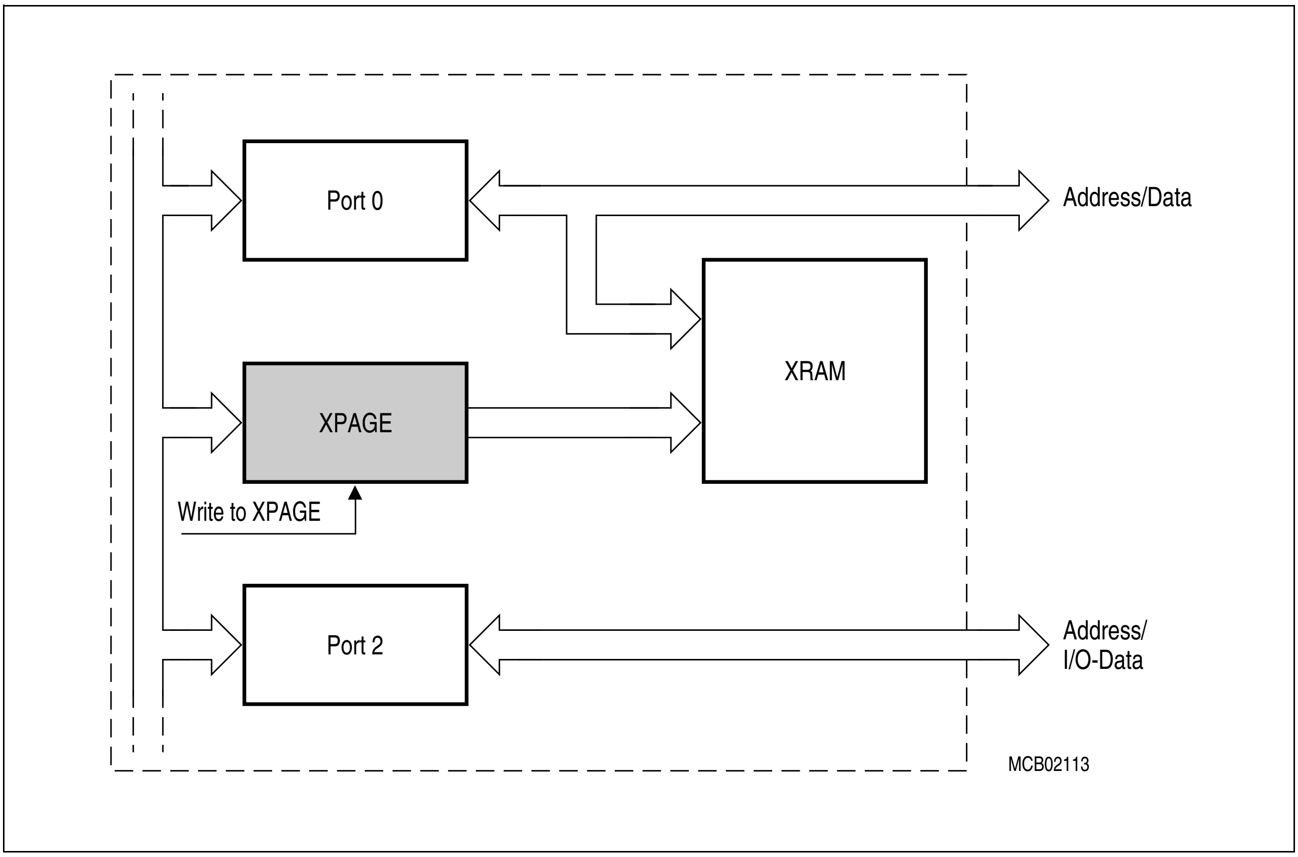


Figure 3-3 Write Page Address to XPAGE

“MOV XPAGE, #pageaddress” or “MOV XPAGE, PAL”, where ‘PAL’ is internal RAM location containing the page address, will write the page address only to the XPAGE register. Port 2 is thus available for addresses or for I/O data.

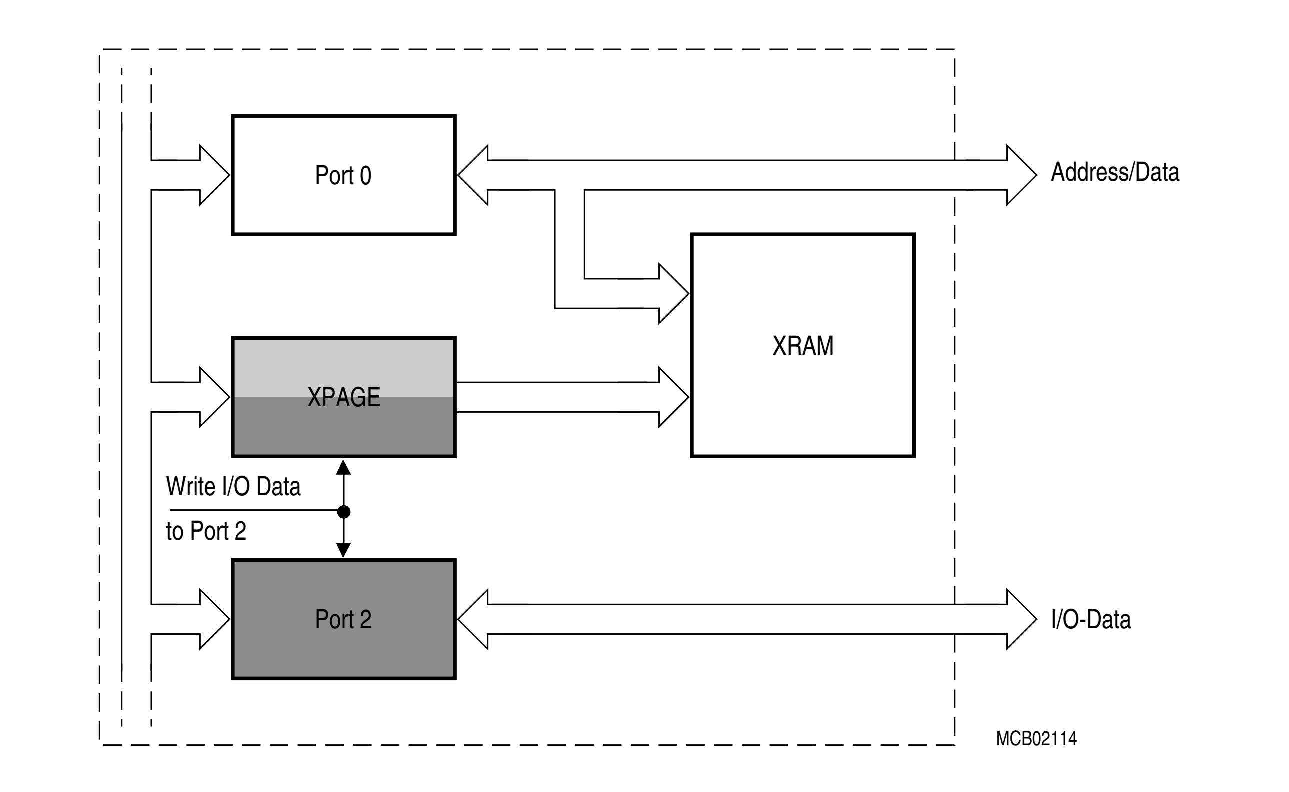


Figure 3-4 Use of Port 2 as I/O Port

With a write to Port 2, the XRAM address in XPAGE register will be overwritten because of the concurrent write to Port 2 and the XPAGE register. So, whenever XRAM is used and the XRAM address differs from the byte written to the 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 the accumulator
```

The register XPAGE provides the upper address byte for accesses to XRAM with MOVX @Ri instructions. If the address formed by XPAGE and Ri points outside the XRAM address range, an external access is performed. For the C508, the content of XPAGE must be FC_H - FF_H in order to use the XRAM.

The software must distinguish two cases, if the MOVX @Ri instructions with paging will be used:

a) Access to XRAM: The upper address byte must be written to XPAGE or P2; both writes select the XRAM 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, while 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 effect on the contents of a XRAM memory location depends on the cycle in which the active reset signal is detected (MOVX is a two-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 the SYSCON Register and on the state of Pin $\overline{\text{EA}}$. **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 an external address/data bus. If (internal) XRAM is accessed, the data written to the XRAM can be seen on the bus in debug mode.
   I/O: The pins work as Input/Output lines under control of their latch.

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

c) Use of internal (XRAM) or external XDATA memory.

The shaded areas in **Table 3-1** describe the standard operation for each C500 device without on-chip XRAM.

		$\overline{\text{EA}}$ = 0			$\overline{\text{EA}}$ = 1		
		XMAP1, XMAP0			XMAP1, XMAP0		
		00	10	X1	00	10	X1
MOVX @DPTR	DPTR < XRAM address range	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used
	DPTR ≥ XRAM address range	a)P0/P2→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c)XRAM is used	a)P0/P2→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)XRAM is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used	a)P0/P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c)XRAM is used	a)P0/P2→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)XRAM is used	a)P0/P2→Bus b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c) ext. memory is used
MOVX @ Ri	XPAGE < XRAM addr. page range	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used
	XPAGE ≥ XRAM addr. page range	a)P0→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c)XRAM is used	a)P0→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)XRAM is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used	a)P0/P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ inactive c)XRAM is used	a)P0→Bus ($\overline{\text{RD}}$/$\overline{\text{WR}}$-Data) P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)XRAM is used	a)P0→Bus P2→I/O b)$\overline{\text{RD}}$/$\overline{\text{WR}}$ active c)ext. memory is used

▨ modes compatible to 8051/C501 family

Table 3-1 - Behavior of P0/P2 and $\overline{\text{RD}}$/$\overline{\text{WR}}$ During MOVX Accesses

3.5 Special Function Registers

With the exception of the program counter and the four general purpose register banks, the registers reside in the special function register area. The special function register area consists of two portions: the standard special function register area and the mapped special function register area. One special function register of the C508 (PCON1) is located in the mapped special function register area. To access the mapped special function register area, bit RMAP in the special function register SYSCON must be set. All other special function registers are located in the standard special function register area which is accessed when RMAP is cleared (0).

Special Function Register SYSCON (Address B1$_H$) **Reset Value: XX10XX01$_B$**

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

▨ The functions of the shaded bits are not described here.

Bit	Function
RMAP	**Special Function Register Map bit** RMAP = 0: Access to the non-mapped (standard) special function register area is enabled. RMAP = 1: Access to the mapped special function register area is enabled.
–	Reserved bits for future use. Read by CPU returns undefined values.

As long as bit RMAP is set, the mapped special function register 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 Special Function Registers (SFRs) with addresses whose address bits 0-2 are 0 (such as: 80$_H$, 88$_H$, 90$_H$, 98$_H$, …, F0$_H$, F8$_H$) are bit-addressable.

The 81 SFRs in the standard and mapped SFR areas include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. The SFRs of the C508 are listed in **Table 3-2** and **Table 3-3**. In **Table 3-2**, they are organized in groups which refer to the functional blocks of the C508. **Table 3-3** illustrates the contents of the SFRs in numeric order by their addresses.

Table 3-2 Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	**$E0_H$**[3]	00_H
	B	B-Register	**$F0_H$**[3]	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	$XXXXX000_B$[4]
	PSW	Program Status Word Register	**$D0_H$**[3]	00_H
	SP	Stack Pointer	81_H	07_H
	SYSCON[1]	System Control Register	$B1_H$	$XX10XX01_B$[4]
	VR0[2]	Version Register 0	FC_H	$C5_H$
	VR1[2]	Version Register 1	FD_H	08_H
	VR2[2]	Version Register 2	FE_H	[5]
A/D-Converter	ADCON0[1]	A/D Converter Control Register 0	**$D8_H$**[3]	$00X00000_B$[4]
	ADCON1	A/D Converter Control Register 1	DC_H	$01XXX000_B$[4]
	ADDATH	A/D Converter Data Register High Byte	$D9_H$	00_H
	ADDATL	A/D Converter Start Register Low Byte	DA_H	$00XXXXXX_B$[4]
Interrupt System	IEN0[1]	Interrupt Enable Register 0	$A8_H$[3]	00_H
	IEN1[1]	Interrupt Enable Register 1	$B8_H$[3]	$X0000000_B$
	IEN2	Interrupt Enable Register 2	$9A_H$	$XX0000XX_B$
	IEN3	Interrupt Enable Register 3	BE_H	$XXX000XX_B$
	IP0[1]	Interrupt Priority Register 0	$A9_H$	00_H
	IP1	Interrupt Priority Register 1	$B9_H$	$XX000000_B$[4]
	TCON[1]	Timer Control Register	88_H[3]	00_H
	T2CON[1]	Timer 2 Control Register	$C8_H$[3]	00_H
	SCON[1]	Serial Channel Control Register	98_H[3]	00_H
	IRCON	Interrupt Request Control Register	$C0_H$[3]	$X0000000_B$
	EINT	External Interrupt Control Register	FB_H	$XX000000_B$
XRAM	XPAGE	Page Address Register for Extended on-chip XRAM and CAN Controller	91_H	00_H
	SYSCON[1]	System Control Register	$B1_H$	$XX10XX01_B$[4]
Ports	P0	Port 0	80_H[3]	FF_H
	P1	Port 1	90_H[3]	FF_H
	P2	Port 2	$A0_H$[3]	FF_H
	P3	Port 3	$B0_H$[3]	FF_H
	P4	Port 4, Analog/Digital Input	DB_H	–
	P5	Port 5	$F8_H$[3]	FF_H

Table 3-2 Special Function Registers - Functional Blocks (cont'd)

Block	Symbol	Name	Address	Contents after Reset
Serial Channel	ADCON0[1]	A/D Converter Control Register 0	$\mathbf{D8_H}$[3]	$00X00000_B$[4]
	PCON[1]	Power Control Register	87_H	00_H
	SBUF	Serial Channel Buffer Register	99_H	XX_H[4]
	SCON	Serial Channel Control Register	$\mathbf{98_H}$[3]	00_H
	SRELL	Serial Channel Reload Register, Low Byte	AA_H	$D9_H$
	SRELH	Serial Channel Reload Register, High Byte	BA_H	$XXXXXX11_B$[4]
Timer 0/ Timer 1	TCON	Timer 0/1 Control Register	$\mathbf{88_H}$[3]	00_H
	TH0	Timer 0, High Byte	$8C_H$	00_H
	TH1	Timer 1, High Byte	$8D_H$	00_H
	TL0	Timer 0, Low Byte	$8A_H$	00_H
	TL1	Timer 1, Low Byte	$8B_H$	00_H
	TMOD	Timer Mode Register	89_H	00_H
Timer 2	CCEN	Compare/Capture Enable Register	$C1_H$	00_H
	T2CCH1	Compare/Capture Register 1, High Byte	$C3_H$	00_H
	T2CCH2	Compare/Capture Register 2, High Byte	$C5_H$	00_H
	T2CCH3	Compare/Capture Register 3, High Byte	$C7_H$	00_H
	T2CCL1	Compare/Capture Register 1, Low Byte	$C2_H$	00_H
	T2CCL2	Compare/Capture Register 2, Low Byte	$C4_H$	00_H
	T2CCL3	Compare/Capture Register 3, Low Byte	$C6_H$	00_H
	CRCH	Comp./Rel./Capt. Register, High Byte	CB_H	00_H
	CRCL	Comp./Rel./Capt. 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	$\mathbf{C8_H}$[3]	00_H

Table 3-2 Special Function Registers - Functional Blocks (cont'd)

Block	Symbol	Name	Address	Contents after Reset
Compare/ Capture Unit	CT1CON	Compare Timer 1 Control Register	E1$_H$	00010000$_B$
	CCPL	Compare Timer 1 Period Register, Low Byte	DE$_H$	00$_H$
	CCPH	Compare Timer 1 Period Register, High Byte	DF$_H$	00$_H$
	CT1OFL	Compare Timer 1 Offset Register, Low Byte	E6$_H$	00$_H$
	CT1OFH	Compare Timer 1 Offset Register, High Byte	E7$_H$	00$_H$
	CMSEL0	Capture/Compare Mode Select Register 0	E3$_H$	00$_H$
	CMSEL1	Capture/Compare Mode Select Register 1	E4$_H$	00$_H$
	COINI	Compare Output Initialization Register	E2$_H$	FF$_H$
	CCL0	Capture/Compare Register 0, Low Byte	F2$_H$	00$_H$
	CCH0	Capture/Compare Register 0, High Byte	F3$_H$	00$_H$
	CCL1	Capture/Compare Register 1, Low Byte	F4$_H$	00$_H$
	CCH1	Capture/Compare Register 1, High Byte	F5$_H$	00$_H$
	CCL2	Capture/Compare Register 2, Low Byte	F6$_H$	00$_H$
	CCH2	Capture/Compare Register 2, High Byte	F7$_H$	00$_H$
	TRCON	Trap Enable Control Register	FF$_H$	00$_H$
	COTRAP	Compare Output In Trap State Register	F9$_H$	00$_H$
	CCIR	Capture/Compare Interrupt Request Flag Reg.	E5$_H$	00$_H$
	CCIE[1)]	Capture/Compare Interrupt Enable Register	D6$_H$	00$_H$
	CT2CON	Compare Timer 2 Control Register	F1$_H$	00010000$_B$
	CP2L	Compare Timer 2 Period Register, Low Byte	D2$_H$	00$_H$
	CP2H	Compare Timer 2 Period Register, High Byte	D3$_H$	XXXXXX00$_B$[4)]
	CMP2L	Compare Timer 2 Compare Register, Low Byte	D4$_H$	00$_H$
	CMP2H	Compare Timer 2 Compare Register, High Byte	D5$_H$	XXXXXX00$_B$[4)]
	BCON	Block Commutation Control Register	D7$_H$	00$_H$
Watchdog Timer	WDTL	Watchdog Timer Register, Low Byte	84$_H$	00$_H$
	WDTH	Watchdog Timer Register, High Byte	85$_H$	X0000000$_B$
	WDTREL	Watchdog Timer Reload Register	86$_H$	00$_H$
	IEN0[1)]	Interrupt Enable Register 0	**A8**$_H$[3)]	00$_H$
	IEN1[1)]	Interrupt Enable Register 1	**B8**$_H$[3)]	00$_H$
	IP0[1)]	Interrupt Priority Register 0	A9$_H$	00$_H$
Power Save Modes	PCON[1)]	Power Control Register	87$_H$	00$_H$
	PCON1[2)]	Power Control Register 1	**88**$_H$[3)]	0XX0XXXX$_B$[4)]

[1)] This special function register is listed repeatedly as some bits of it also belong to other functional blocks.

[2)] This SFR is a mapped SFR. To access this SFR, bit RMAP in SFR SYSCON must be set.

[3)] Bit-addressable special function registers.

[4)] "X" means that the value is undefined and the location is reserved.

[5)] The content of this SFR varies with the actual step of the C508 (e.g. 01$_H$ for C508-4E, first step and 11$_H$ for C508-4R, first step).

Table 3-3 Contents of the SFRs, SFRs in Numeric Order by Address

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
84_H	WDTL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
85_H	WDTH	X000-0000_B	–	.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-$XXXX_B$	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	.7	.6	.5	.4	.3	.2	.1	.0
91_H	XPAGE	00_H	.7	.6	.5	.4	.3	.2	.1	.0
92_H	DPSEL	XXXX-$X000_B$	–	–	–	–	–	.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
9A	IEN2	XX00-$00XX_B$	–	–	ECT1	ECCM	ECT2	ECEM	–	–
$A0_H$[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

Table 3-3 Contents of the SFRs, SFRs in Numeric Order by Address (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	XMAP 0
B8$_H$[2]	IEN1	X000-0000$_B$	–	SWDT	EX6	EX5	EX4	EX3	EX2	EADC
B9$_H$	IP1	XX00-0000$_B$	–	–	.5	.4	.3	.2	.1	.0
BA$_H$	SRELH	XXXX-XX11$_B$	–	–	–	–	–	–	.1	.0
BE$_H$	IEN3	XXX0-00XX$_B$	–	–	–	EX9	EX8	EX7	–	–
C0$_H$[2]	IRCON	X000-0000$_B$	–	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC
C1$_H$	CCEN	00$_H$	COCA H3	COCA L3	COCA H2	COCA L2	COCA H1	COCA L1	COCA H0	COCA L0
C2$_H$	T2CCL1	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C3$_H$	T2CCH1	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C4$_H$	T2CCL2	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C5$_H$	T2CCH2	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C6$_H$	T2CCL3	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C7$_H$	T2CCH3	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
C8$_H$[2]	T2CON	0000-X0X0$_B$	T2PS	I3FR	I2FR	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
D2$_H$	CP2L	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
D3$_H$	CP2H	XXXX.XX00$_B$	–	–	–	–	–	–	.1	.0
D4$_H$	CMP2L	00$_H$	.7	.6	.5	.4	.3	.2	.1	.0
D5$_H$	CMP2H	XXXX.XX00$_B$	–	–	–	–	–	–	.1	.0

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

Addr	Register	Content after Reset[1)]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$D6_H$	CCIE	00_H	ECTP	ECTC	CC2 FEN	CC2 REN	CC1 FEN	CC1 REN	CC0 FEN	CC0 REN
$D7_H$	BCON	00_H	BCMP BCEM	PWM1	PWM0	EBCE	BCER R	BCEN	BCM1	BCM0
$D8_H$[2)]	ADCON0	$00X0\text{-}0000_B$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0
$D9_H$	ADDATH	00_H	.9	.8	.7	.6	.5	.4	.3	.2
DA_H	ADDATL	$00XX\text{-}XXXX_B$	.1	.0	–	–	–	–	–	–
DB_H	P4	–	.7	.6	.5	.4	.3	.2	.1	.0
DC_H	ADCON1	$01XX\text{-}X000_B$	ADCL1	ADCL0	–	–	–	MX2	MX1	MX0
DE_H	CCPL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
DF_H	CCPH	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E0_H$[2)]	ACC	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E1_H$	CT1CON	$0001\text{-}0000_B$	CTM	ETRP	STE1	CT1 RES	CT1R	CLK2	CLK1	CLK0
$E2_H$	COINI	FF_H	COUT 3I	COUT XI	COUT 2I	CC2I	COUT 1I	CC1I	COUT 0I	CC0I
$E3_H$	CMSEL0	00_H	CMSE L13	CMSE L12	CMSE L11	CMSE L10	CMSE L03	CMSE L02	CMSE L01	CMSE L00
$E4_H$	CMSEL1	00_H	ESMC	NMCS	0	0	CMSE L23	CMSE L22	CMSE L21	CMSE L20
$E5_H$	CCIR	00_H	CT1FP	CT1FC	CC2F	CC2R	CC1F	CC1R	CC0F	CC0R
$E6_H$	CT1OFL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E7_H$	CT1OFH	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F0_H$[2)]	B	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F1_H$	CT2CON	$0001\text{-}0000_B$	CT2P	ECT2O	STE2	CT2 RES	CT2R	CLK2	CLK1	CLK0
$F2_H$	CCL0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F3_H$	CCH0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F4_H$	CCL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F5_H$	CCH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F6_H$	CCL2	00_H	.7	.6	.5	.4	.3	.2	.1	.0

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

Addr	Register	Content after Reset[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$F7_H$	CCH2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F8_H$[2]	P5	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
$F9_H$	COTRAP	00_H	BCT SEL	RES	COUT2 T	CC2T	COUT1 T	CC1T	COUT0 T	CC0T
FB_H	EINT	$XX00\text{-}0000_B$	–	–	IEX9	I9FR	IEX8	I8FR	IEX7	I7FR
FC_H [3][4]	VR0	$C5_H$	1	1	0	0	0	1	0	1
FD_H [3][4]	VR1	08_H	0	0	0	0	1	0	0	0
FE_H [3][4]	VR2	[5]	.7	.6	.5	.4	.3	.2	.1	.0
FF_H	TRCON	00_H	TRPEN	TRF	TREN5	TREN4	TREN3	TREN2	TREN1	TREN 0

[1] X means that the value is undefined and the location is reserved.

[2] Bit-addressable special function registers.

[3] SFR is located in the mapped SFR area. To access this SFR, bit RMAP in SFR SYSCON must be set.

[4] These are read-only registers.

[5] The content of this SFR varies with the actual step of the C508 (e.g. 01_H for C508-4E, first step and 11_H for C508-4R, first step).

4 External Bus Interface

The C508 allows for external memory expansion. The functionality and implementation of the external bus interface are identical to the common interface for the 8051 architecture with one exception. The exception is the suppression of the ALE signal generation when the C508 is used in systems with no external memory. Resetting the EALE bit in SFR SYSCON gates off the ALE signal. This feature reduces RFI emissions of the system.

4.1 Accessing External Memory

It is possible to distinguish between accesses to external program memory and accesses to external data memory or other peripheral components. This distinction is possible because 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 to access 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 pull-up resistors.

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

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

Thus, the Port 2 latch does not need to contain ‘1’s and the contents of the Port 2 SFR are not modified.

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

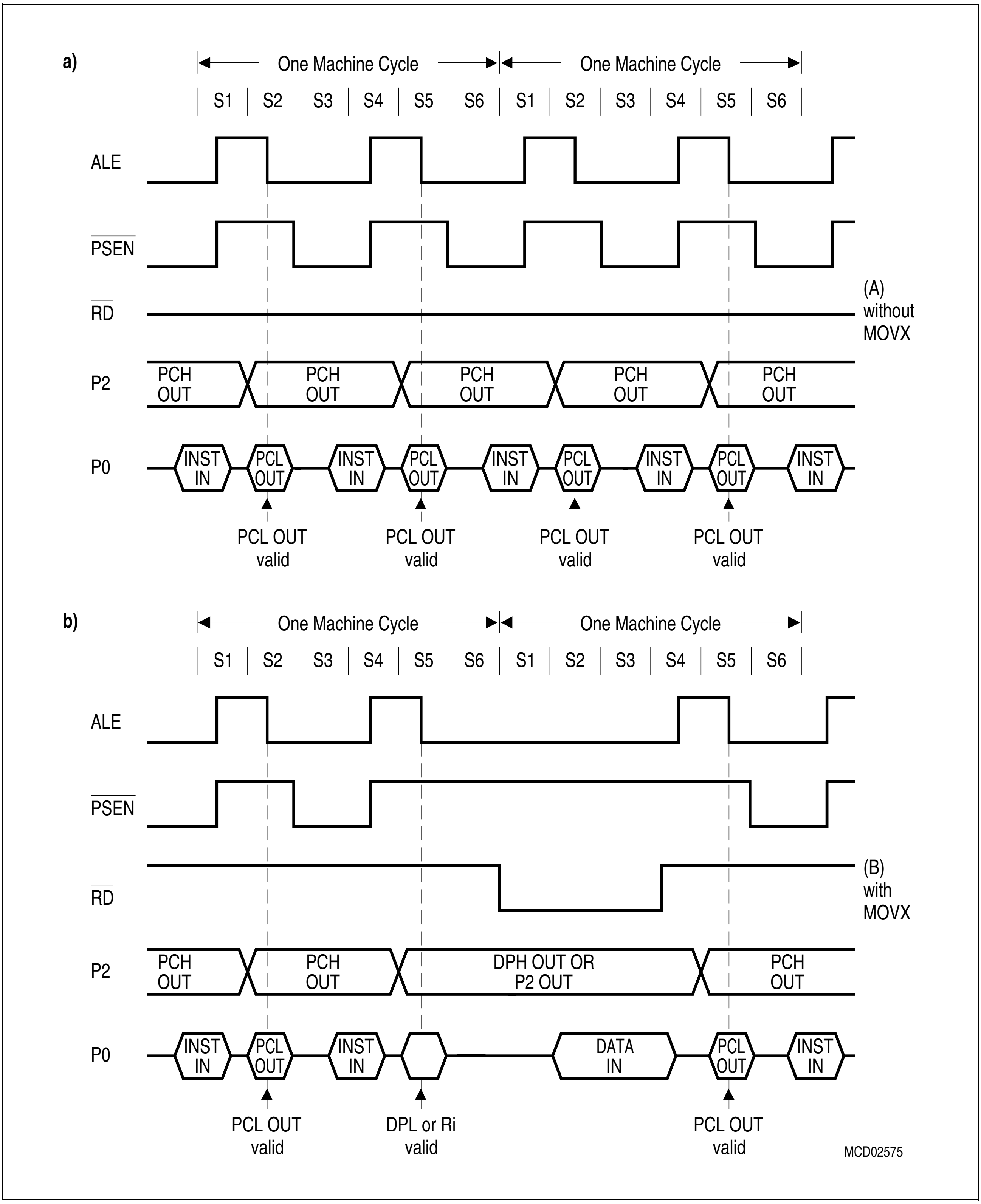


Figure 4-1 External Program Memory Execution

4.1.2 Timing

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

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

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

4.1.3 External Program Memory Access

The external program memory is accessed under two conditions:

– whenever signal $\overline{\text{EA}}$ is active (low) or
– whenever the program counter (PC) content is greater than 7FFF$_H$

When the CPU is executing out of external program memory, all eight bits of Port 2 are dedicated to an output function and must not be used for general-purpose I/O. The contents of the Port 2 SFR, however, are not affected. During external program memory fetches, Port 2 lines output the high byte of the PC, and during accesses to external data memory, they output either DPH or the Port 2 SFR (determined by whether 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 three oscillator periods. A complete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$, takes 1.5 oscillator periods. (The execution sequence for these two types of read cycles is shown in **Figure 4-1 a)** and **b)**.

4.3 Overlapping External Data and Program Memory Spaces

In some applications, it is desirable to execute a program from the same physical memory that is used for storing data. In the C508, 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. As the $\overline{\text{PSEN}}$ cycle is faster than the $\overline{\text{RD}}$ cycle, the external memory must be fast enough to adapt to the $\overline{\text{PSEN}}$ cycle.

4.4 ALE, Address Latch Enable

The C508 allows the ALE output signal to be switched off. If the internal ROM is used ($\overline{\text{EA}}$ = 1 and PC ≤ $7FFF_H$) and ALE is switched off by EALE = 0, then, ALE will go active only during external data memory accesses (MOVX instructions). If $\overline{\text{EA}}$ = 0, the ALE generation is always enabled and the bit EALE has no effect.

After a hardware reset, ALE generation is enabled.

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

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
EALE	**Enable ALE output** EALE = 0: ALE generation is disabled; disables ALE signal generation during internal code memory accesses ($\overline{\text{EA}}$ = 1). With $\overline{\text{EA}}$ = 1, ALE is automatically generated at MOVX instructions. EALE = 1: ALE generation is enabled. If $\overline{\text{EA}}$ = 0, the ALE generation is always enabled and the bit EALE has no effect on the ALE generation.
–	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 about 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. This also ensure that emulation and production chips are identical.

The Enhanced Hooks Technology requires embedded logic in the C500 and allows the C500 when used with an EH-IC, to function in a manner similar to 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 various 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.

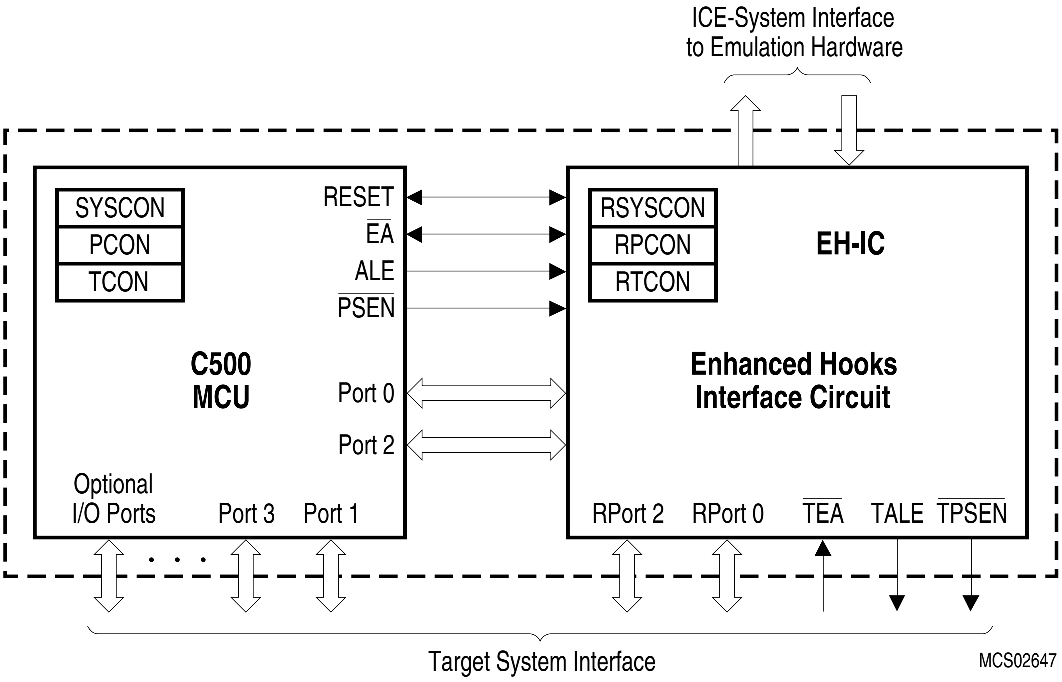


Figure 4-2 Basic C500 MCU Enhanced Hooks Concept Configuration

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

4.6 Eight Datapointers for Faster External Bus Access

4.6.1 The Importance of Additional Datapointers

The standard 8051 architecture provides only 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 must be handled bytewise. For complex applications with peripherals located in the external data memory space or extended data storage capacity, this factor 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 Implementation of the Eight Datapointers

Simply adding more datapointers is not suitable because of the need to keep up 100% compatibility with the 8051 instruction set. This instruction set, however, 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 and 100% compatibility with 8051 architecture), the C508 contains a set of eight 16-bit registers from which the actual datapointer can be selected.

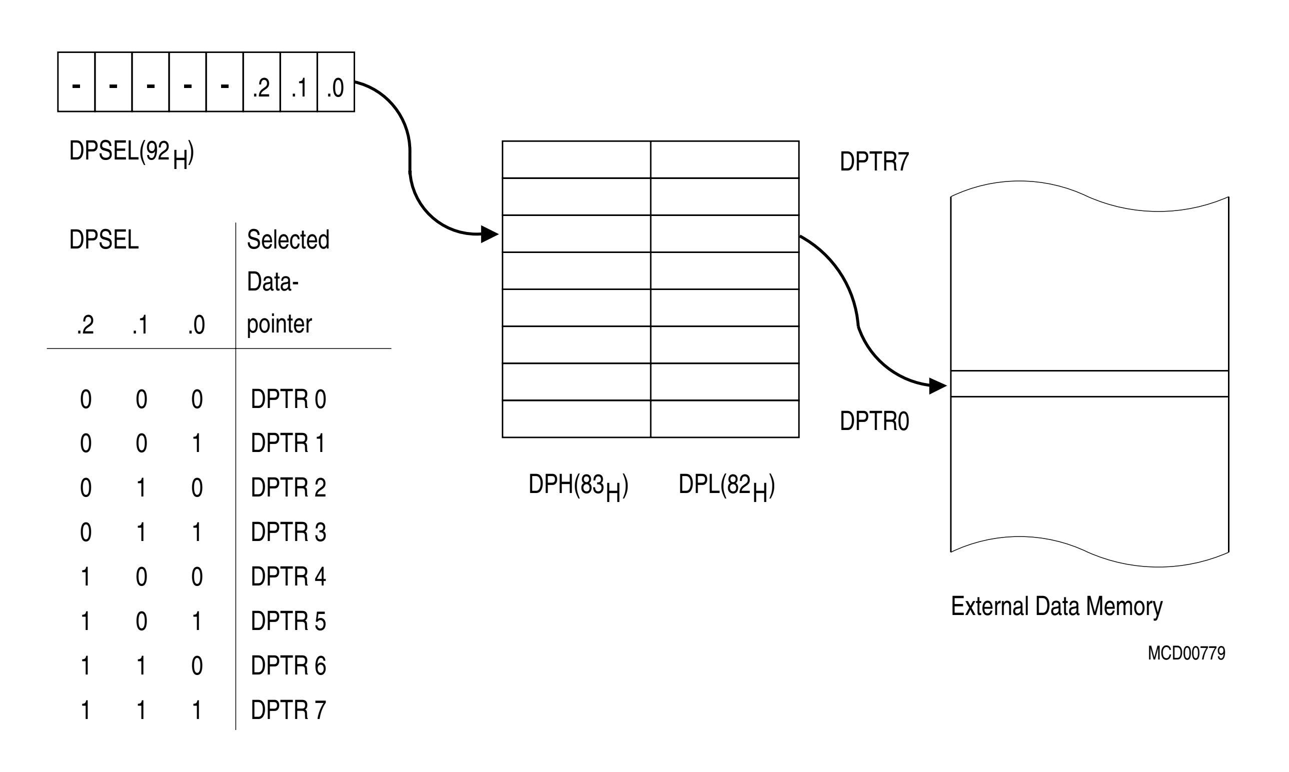
This means that the user’s program may keep up to eight 16-bit addresses resident in these registers; but, only one register at a time is selected to be the datapointer. Thus, the datapointer in turn is accessed (or selected) via indirect addressing. This indirect addressing is done through a special function register called DPSEL (Data Pointer Select register). All instructions of the C508 which handle the datapointer, therefore, affect only one of the eight pointers which is addressed by DPSEL at that very moment.

Figure 4-3 illustrates the addressing mechanism. A 3-bit field in register DPSEL points to the DPTRx currently used. Any standard 8051 instruction (such as 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.



DPSEL .2	 .1	 .0	Selected Data-pointer
0	0	0	DPTR 0
0	0	1	DPTR 1
0	1	0	DPTR 2
0	1	1	DPTR 3
1	0	0	DPTR 4
1	0	1	DPTR 5
1	1	0	DPTR 6
1	1	1	DPTR 7

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

4.6.3 Advantages of Multiple Datapointers

This mechanism for addressing 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 single instruction to select a new datapointer is sufficient. If the program uses only one datapointer, then it must save the old value (with two 8-bit instructions) and load the new address, byte-by-byte. This not only takes more time, it also requires additional space in the internal RAM.

4.6.4 Application Example and Performance Analysis

The following example demonstrates the involvement 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), #0FFH  ;Initialize shadow_variables with
                              source_pointer
MOV     HIGH(SRC_PTR), #1FH
MOV     LOW(DES_PTR), #0A0H  ;Initialize shadow_variables with
                              destination_pointer
MOV     HIGH(DES_PTR), #2FH
```

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 C508)

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 C508's multiple datapointers can make external bus accesses twice 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, as well. For some applications in which all eight datapointers are employed, this means that a C508 program has up to 24 bytes (16 variables and 8 stack bytes) of the internal RAM available for other uses.

4.7 ROM/OTP Protection for the C508-4R / C508-4E

The C508-4R allows protection of the contents of the internal ROM against unauthorized read out. The type of ROM protection is fixed with the ROM mask. Therefore, users of the C508-4R version must define whether ROM protection has to be selected or not.

The C508-4E OTP version also allows program memory protection at several levels (see **Chapter 10.6**). The program memory protection for the C508-4E can be activated after programming of the device.

The C508-4R devices, which operate from internal ROM, are always checked for correct ROM contents during production test. Therefore, unprotected as well as protected ROMs must provide a procedure to verify the ROM contents. In ROM verification mode 1, which is used to verify unprotected ROMs, a ROM address is applied externally to the C508-4R and the ROM data byte is output at Port 0. ROM verification mode 2, which is used to verify ROM protected devices, operates differently. In this mode, ROM addresses are generated internally and the expected data bytes must be applied externally to the device (by the manufacturer or by the customer) and are compared internally with the data bytes from the ROM. After 16 byte verify operations, the state of the P3.5 pin shows whether the last 16 bytes have been verified as expected.

This mechanism provides very high security for ROM protection. Only the owner of the ROM code and the manufacturer who knows the contents of the ROM can read out and verify it.

The behavior of the move code instruction, when the code is executed from the external ROM, is such that accessing a code byte from a protected on-chip ROM address is not possible. In this case, the byte accessed will be invalid.

4.7.1 Unprotected ROM Mode

If the ROM is unprotected, ROM verification mode 1 as shown in **Figure 4-4** is used to read out the contents of the ROM. Please refer to the AC specifications in the C508 Data Sheet for the AC timing characteristics of the ROM verification modes.

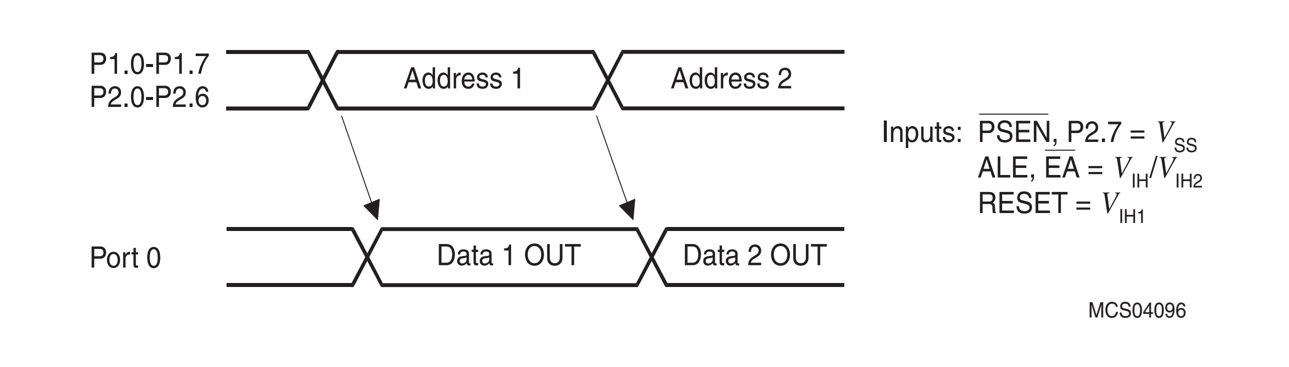
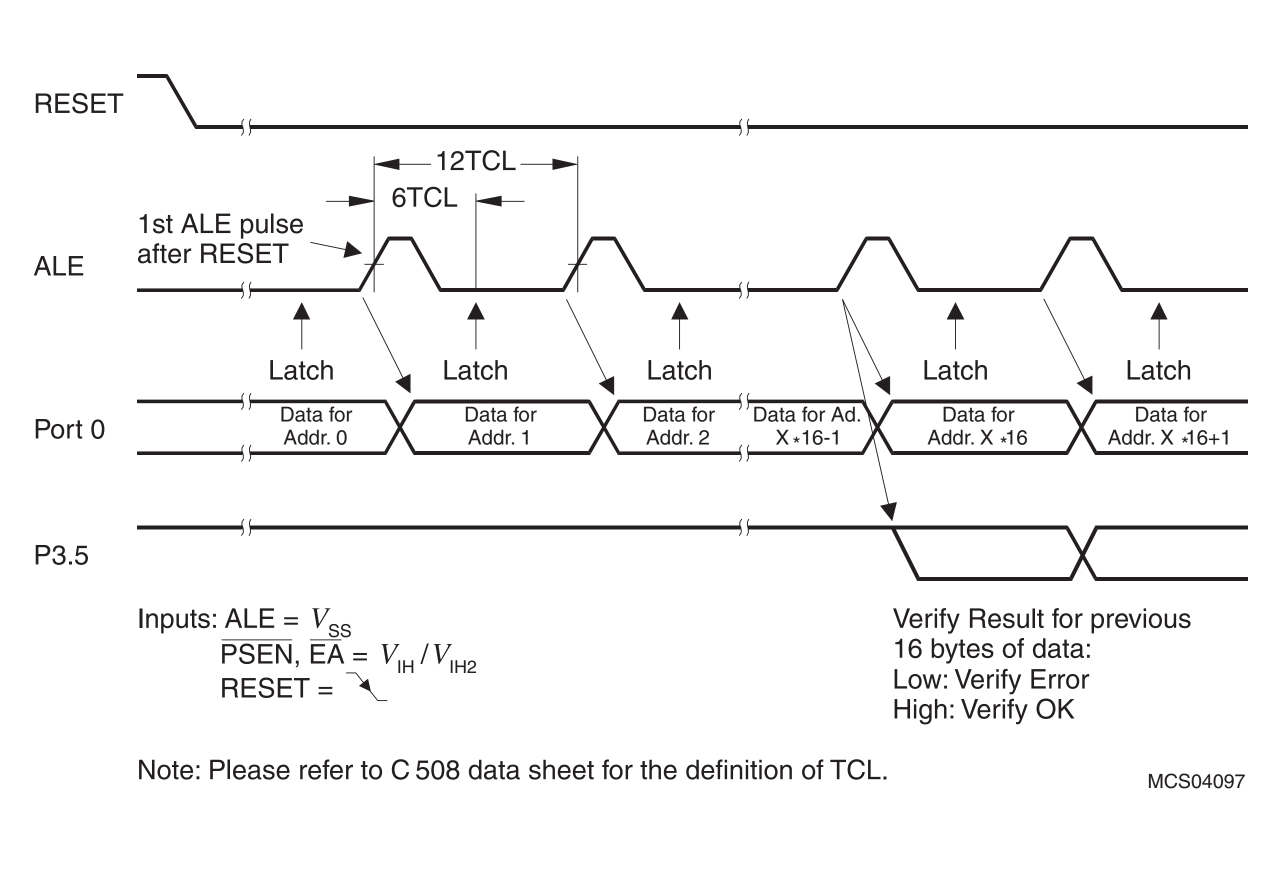

P1.0-P1.7
P2.0-P2.6
Address 1
Address 2
Inputs: $\overline{\text{PSEN}}$, P2.7 = V_{SS}
ALE, $\overline{\text{EA}}$ = V_{IH}/V_{IH2}
RESET = V_{IH1}
Port 0
Data 1 OUT
Data 2 OUT
MCS04096

Figure 4-4 ROM Verification Mode 1

ROM verification mode 1 is selected if the inputs $\overline{\text{PSEN}}$, ALE, $\overline{\text{EA}}$, and RESET are put to the specified logic level. Then, the 15-bit address of the internal ROM byte to be read is applied to the Port 1 and Port 2 lines. After a delay, Port 0 outputs the content of the addressed ROM cell. In ROM verification mode 1, the C508 must be provided with a system clock at the XTAL pins and pull-up resistors on the Port 0 lines.

4.7.2 Protected ROM/OTP Mode

If the C508-4R ROM is protected by mask (or C508-4E OTP is used in protection level 1), ROM/OTP verification mode 2 is used to verify the contents of the ROM, as shown in **Figure 4-5**. Please refer the AC specifications in the C508 Data Sheet for detailed timing characteristics of the ROM verification modes.



Inputs: ALE = V_{SS}
$\overline{\text{PSEN}}$, $\overline{\text{EA}}$ = V_{IH} / V_{IH2}
RESET = ↘

Verify Result for previous
16 bytes of data:
Low: Verify Error
High: Verify OK

Note: Please refer to C 508 data sheet for the definition of TCL.

Figure 4-5 ROM Verification Mode 2

ROM/OTP verification mode 2 is selected if the inputs $\overline{\text{PSEN}}$, $\overline{\text{EA}}$, and ALE are put to the specified logic levels. When RESET goes inactive, the ROM/OTP verification mode 2 sequence is started. The C508 outputs an ALE signal with a period of 12TCL and expects data bytes at Port 0. The data bytes at Port 0 are assigned to the ROM addresses in the following way:

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

The C508 does not output any address information during the ROM/OTP verification mode 2. The first data byte to be verified is always the byte which is assigned to the internal ROM address 0000_H and must be put onto the data bus with the falling edge of RESET. With each following ALE pulse, the ROM/OTP address pointer is internally incremented and the expected data byte for the next ROM address must be delivered externally.

Between two ALE pulses, the data at Port 0 is latched (at 6TCL after ALE rising edge) and is compared internally with the ROM/OTP contents 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 a static level (low for fail and high for pass) while the next 16 bytes are checked. The output of P3.5 will be updated according to the cumulated verify result of the previous 16 bytes of data. In ROM verification mode 2, the C508 must be provided with a system clock at the XTAL pins.

Figure 4-6 shows an application example of external circuitry which allows verification of a protected ROM inside the C508-4R in ROM/OTP verification mode 2. When RESET goes inactive, the C508 starts the ROM/OTP verify sequence. Its ALE is clocking a 15-bit address counter. This counter generates the addresses for an external EPROM which is programmed with the contents of the internal (protected) ROM/OTP. The verify detect logic typically displays the state of the verify error output at P3.5. P3.5 can be latched with the falling edge of ALE.

The CY signal of the address counter indicates to the verify detect logic the end of the internal ROM verification.

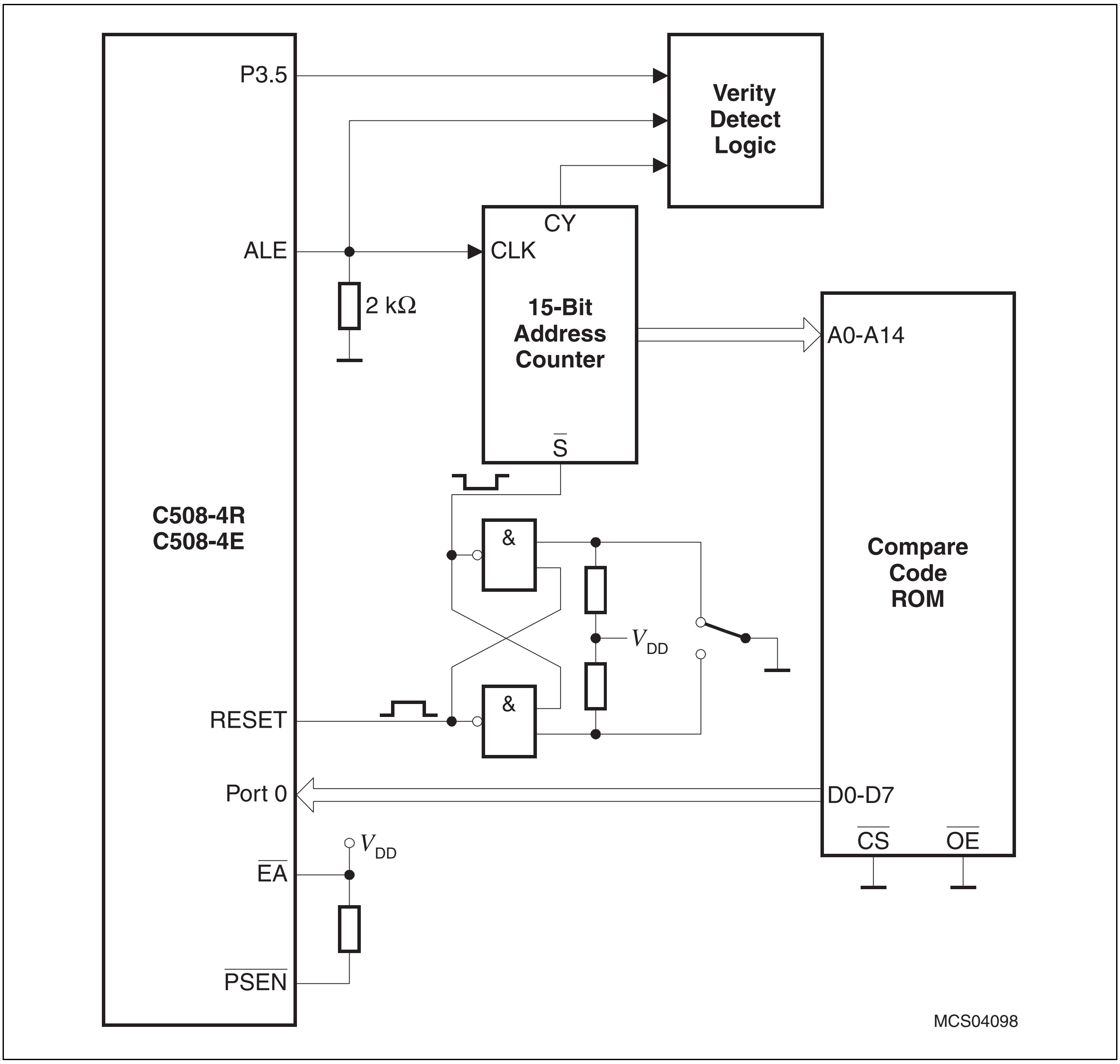


Figure 4-6 ROM/OTP Verification Mode 2 - External Circuitry Example

4.8 Version Registers

Version Registers are typically used for adapting the programming firmware to specific device characteristics such as ROM/OTP size etc.

Three Version Registers are implemented in the C508. They can be read during normal program execution mode as mapped SFRs when the bit RMAP in SFR SYSCON is set.

5 Reset and System Clock Operation

5.1 Hardware Reset Operation

The hardware reset function incorporated in the C508 allows easy automatic startup with minimal additional hardware and forces the controller into a predefined default state. The hardware reset function can also be used during normal operation to restart the device. This is particularly useful for terminating the power-down mode.

The hardware reset is applied externally to the C508. Additionally, there are three internal reset sources: the Watchdog Timer, the Oscillator Watchdog, and the PLL. This section deals with the external hardware reset only.

The reset input is an active high input. An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the RESET pin must be held high for at least two machine cycles (six oscillator periods) while the oscillator is running. The internal reset is executed during the second machine cycle while the oscillator is running 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. (External stimulation of these lines during reset activates several reserved test modes. This, in turn, may cause unpredictable output operations at several port pins).

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

The time required for a reset operation includes the oscillator startup time, the PLL lock time, and the time for two machine cycles, must be at least 10 - 20 ms, under normal conditions. 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 that at least two machine cycles have passed before the reset signal goes inactive.

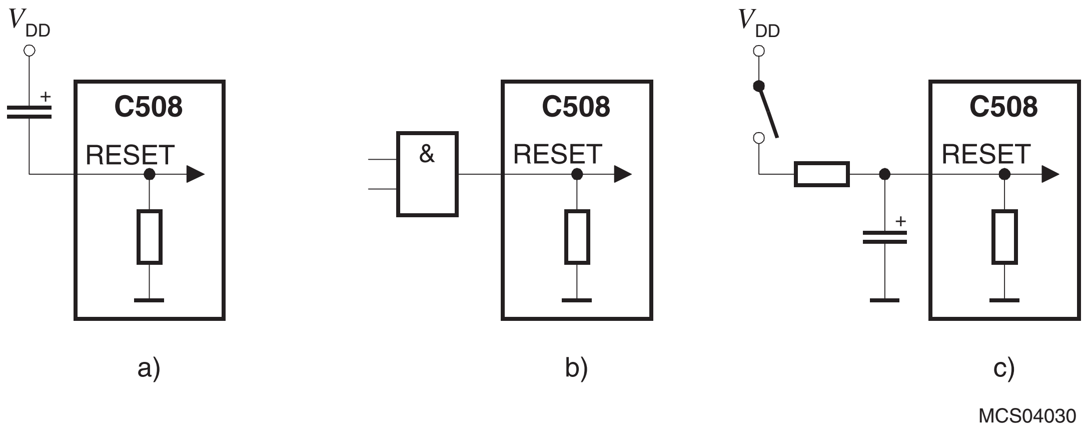


Figure 5-1 Reset Circuitries

A correctly executed reset leaves the processor in a defined state. The program execution starts at location 0000_H. After reset is internally accomplished, the port latches of Ports 0, 1, 2, 3, and 5 default to 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 5) output a one (1). Port 2 lines output a zero after reset, if the $\overline{\text{EA}}$ pin is held low; or one if $\overline{\text{EA}}$ is held high. Port 4 is a uni-directional input port. It has no internal latch; therefore, the contents of the Special Function Register P4 depend on the levels applied to Port 4. 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 C508 are not affected by a reset. The contents are undefined after power-up; the contents remain unchanged during reset if the power supply is not turned off.

5.2 Fast Internal Reset after Power-On

The C508 uses the Oscillator Watchdog unit for a fast internal reset procedure after power-on. The clock source is provided by the RC Oscillator during the internal reset procedure. When the on-chip oscillator is stabilized, its clock output is multiplied by a fixed factor of two by the on-chip PLL. The clock from the PLL is then provided as the system clock. Thus, the system clock frequency is twice the external oscillator frequency. **Figure 5-2** shows the power-on sequence under the control of the oscillator watchdog.

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

The Oscillator Watchdog unit in the C508 avoids this situation because its RC Oscillator starts working within a very short startup time after power-on (typically less than 2 µs). The on-chip oscillator that feeds the PLL has not started yet, and, thus, the PLL remains unlocked. As long as the PLL is not locked, the watchdog uses the RC Oscillator output as the clock source for the chip. This allows the part to be correctly reset and also brings all ports to the defined state (see **Figure 5-2, II**). The exception is Port 1 which is used as compare/capture outputs. These pins will be set to their default levels as soon as the external reset is active. This is illustrated in **Figure 5-3**.

Under worst case conditions (fast V_{DD} rise time - such as 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 started (**Figure 5-2, III)** and the PLL is locked, the Oscillator Watchdog detects the correct function. Then, the watchdog continues to hold the reset active for a time period of 768 cycles (maximum) of the RC oscillator clock to ensure that a stable clock is available from the PLL (**Figure 5-2, IV**). Subsequently, the system clock is supplied by the PLL and the oscillator watchdog's reset is released (**Figure 5-2, V**). However, an externally applied reset still remains active and the device does not start program execution before the external reset is also released (**Figure 5-2, VI**).

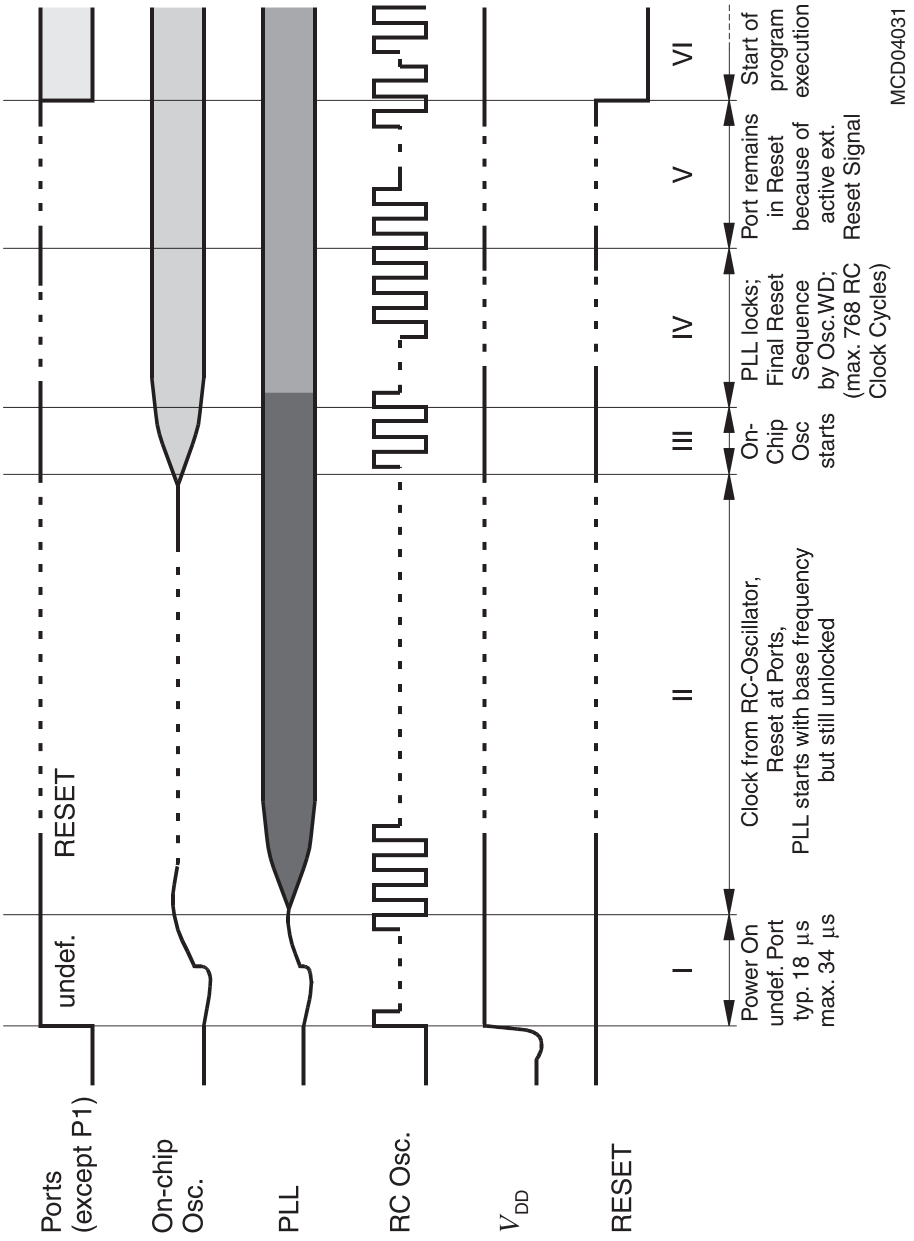


Figure 5-2 Power-On Reset of the C508

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

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

Fast reset of Port 1, that is the Compare/Capture pins, during power-on.

If a crystal or ceramic resonator is used 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 IV in **Figure 5-2**). When an external clock generator is used, phase II is very short. Therefore, an external reset time of 1 ms (typical) is sufficient in most applications.

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

Figure 5-3 is a close-up view of Phase 1 shown in **Figure 5-2**. When RESET is high after V_{DD} is stable, Port 1 will be defined with its default value (high). All other ports will still remain undefined for at most 34 µs.

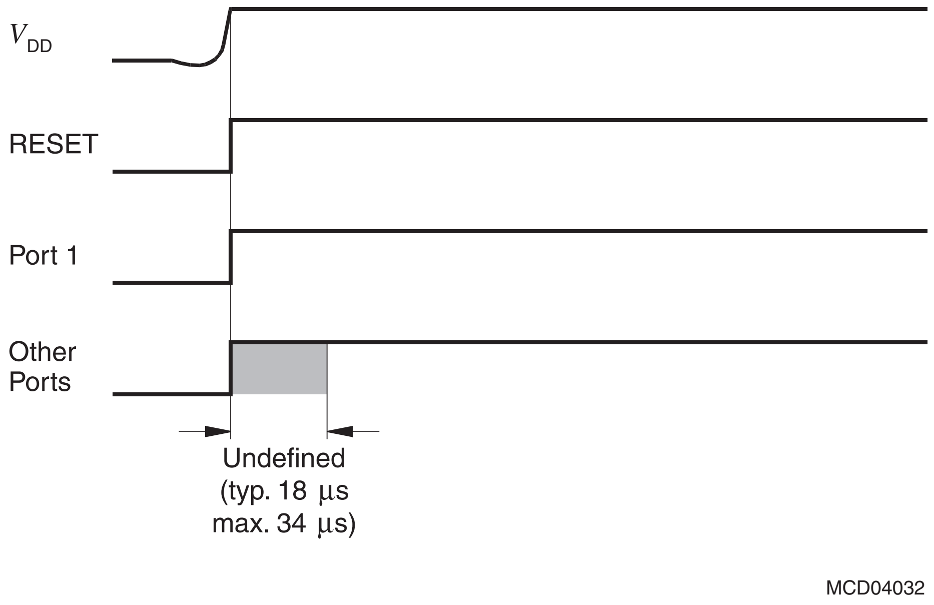


Figure 5-3 Fast Reset of Port 1 Pins

5.3 Hardware Reset Timing

This section describes the timing of the hardware reset signal.

The input pin RESET is sampled once during each machine cycle. This happens in State 5 Phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (high level), the internal reset procedure is started. It needs two complete machine cycles to put the complete device into its correct reset state. In that state, all special function registers contain their default values, the port latches contain '1's, etc. Note that this reset procedure is also performed if there is no clock available to the device. (This is done by the Oscillator Watchdog, which provides an auxiliary clock for performing a perfect reset without clock at the XTAL1 and XTAL2 pins). The RESET signal must be active for at least two machine cycles; after this time, the C508 remains in its reset state as long as the signal is active. When the signal goes inactive, this transition is recognized in the following State 5 Phase 2 of the machine cycle. Then, the processor starts its address output (when configured for external ROM) in the following State 5 Phase 1. One phase later (State 5 Phase 2) the first falling edge at pin ALE occurs.

Figure 5-4 shows this timing for a configuration with $\overline{\text{EA}}$ = 0 (external program memory). Thus, between the release of the RESET signal and the first falling edge at ALE, there is a time period of at least one machine cycle but less than two machine cycles.

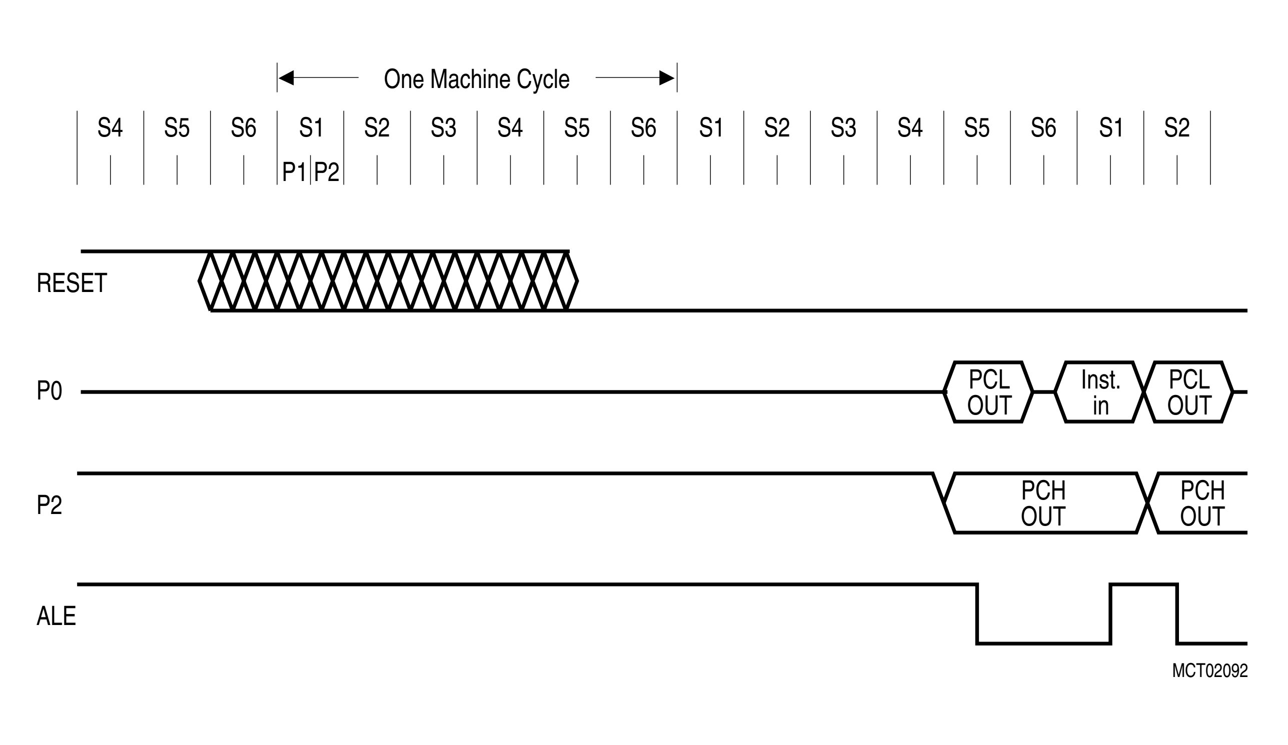


Figure 5-4 CPU Timing after Reset

5.4 Clock Generation

The top-level view of the system clock generation of the C508 is shown in **Figure 5-5**.

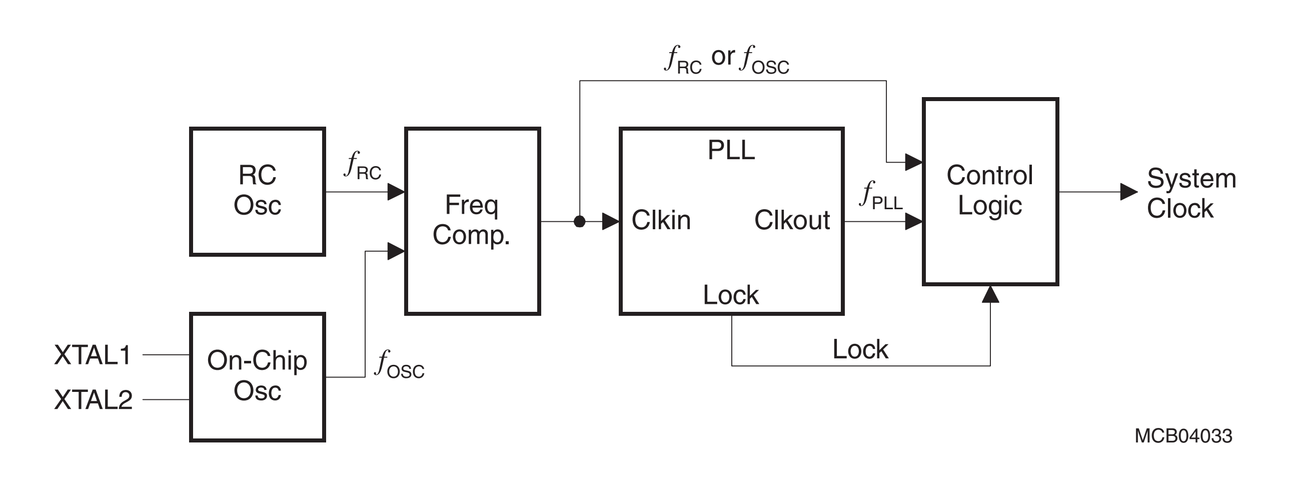


Figure 5-5 Block Diagram of the Clock Generation

The clock generation block consists of the RC oscillator, the on-chip oscillator, and the PLL.

At power-on reset, the RC oscillator takes a shorter time to start in comparison to the on-chip oscillator (typically 2 µs versus 10 ms). While the on-chip oscillator is still unstable, the PLL remains unlocked. Thus, the RC clock is provided as the system clock.

When the on-chip oscillator has stabilized, the PLL locks within 1 ms, providing a clock frequency twice that of the on-chip oscillator's frequency. The system clock source is now switched to the PLL clock.

External reset from the pin should be released only after this stage.

5.5 PLL Operation

Within 1 ms after stable oscillations of the input clock within the specified frequency range, the PLL will be synchronous with this clock at a frequency twice that of the input frequency. In other words, the PLL locks onto its input clock.

Since the PLL constantly adapts to the external clock to remain locked, the CPU clock generated has a slight variation known as jitter. This jitter is irrelevant for longer time periods. For short periods (one to four CPU clock cycles), it remains below 4%.

When the PLL detects a missing input clock signal, it releases the lock signal. Consequently, an internal reset will be active until the PLL is locked again. This may occur if the input clock is unstable or fails completely; for example, due to a broken crystal. In this case, an Oscillator Watchdog reset will also occur.

When software power-down mode is entered, the PLL is powered down together with the RC oscillator and the on-chip oscillator. In this mode, the PLL is marked unlocked; however, no internal resets will be generated.

5.6 Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter which can be configured with off-chip components 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 5-6 shows the recommended oscillator circuit.

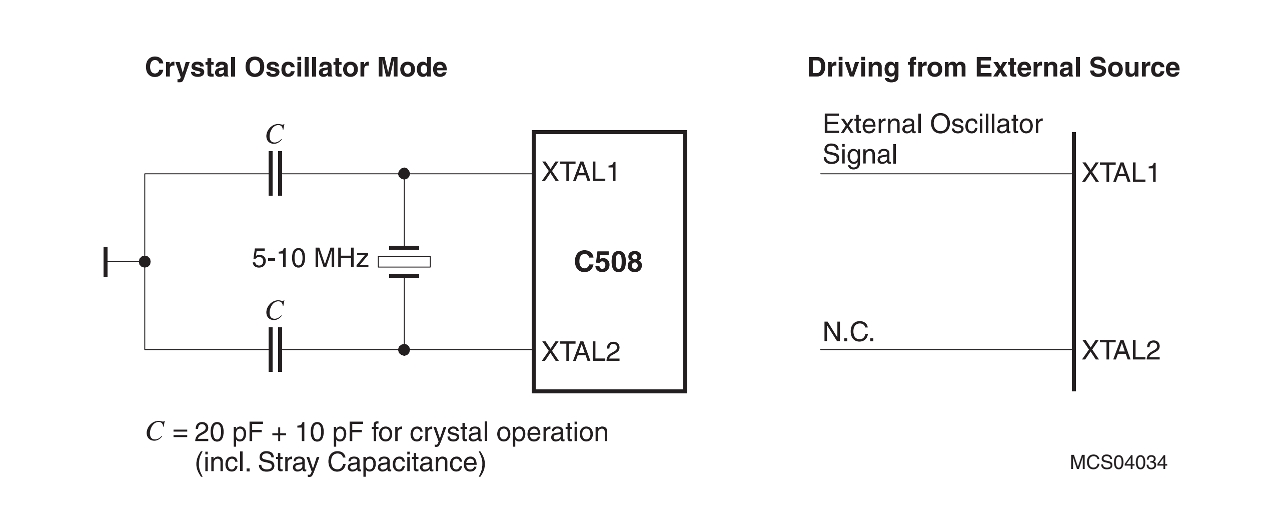

Crystal Oscillator Mode
Driving from External Source
C
C
5-10 MHz
XTAL1
C508
XTAL2
External Oscillator Signal
XTAL1
N.C.
XTAL2
C = 20 pF + 10 pF for crystal operation
(incl. Stray Capacitance)
MCS04034

Figure 5-6 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 5-7**). The on-chip oscillator is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. 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. It is recommended that the manufacturer of the ceramic resonator be consulted for value specifications of these capacitors.

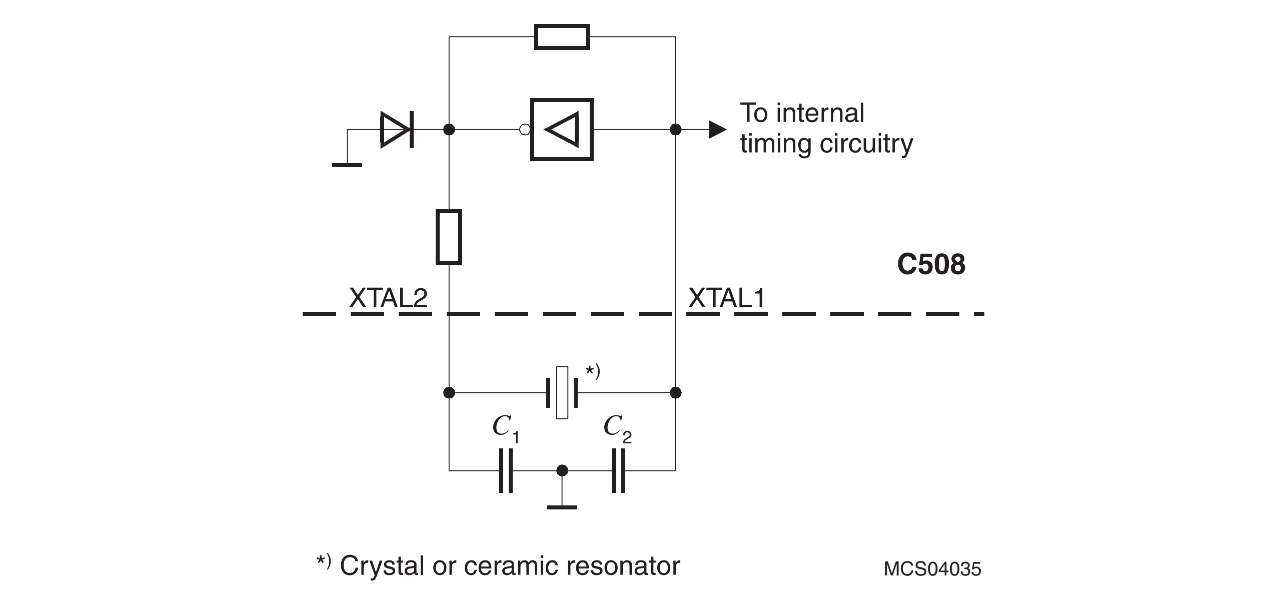


Figure 5-7 On-Chip Oscillator Circuitry

To drive the C508 with an external clock source, the external clock signal must be applied to XTAL1, as shown in **Figure 5-8**. XTAL2 must be left unconnected. A pull-up resistor is suggested (to increase the noise margin), but is optional if V_{OH} of the driving gate corresponds to the V_{IH3} specification of XTAL1.

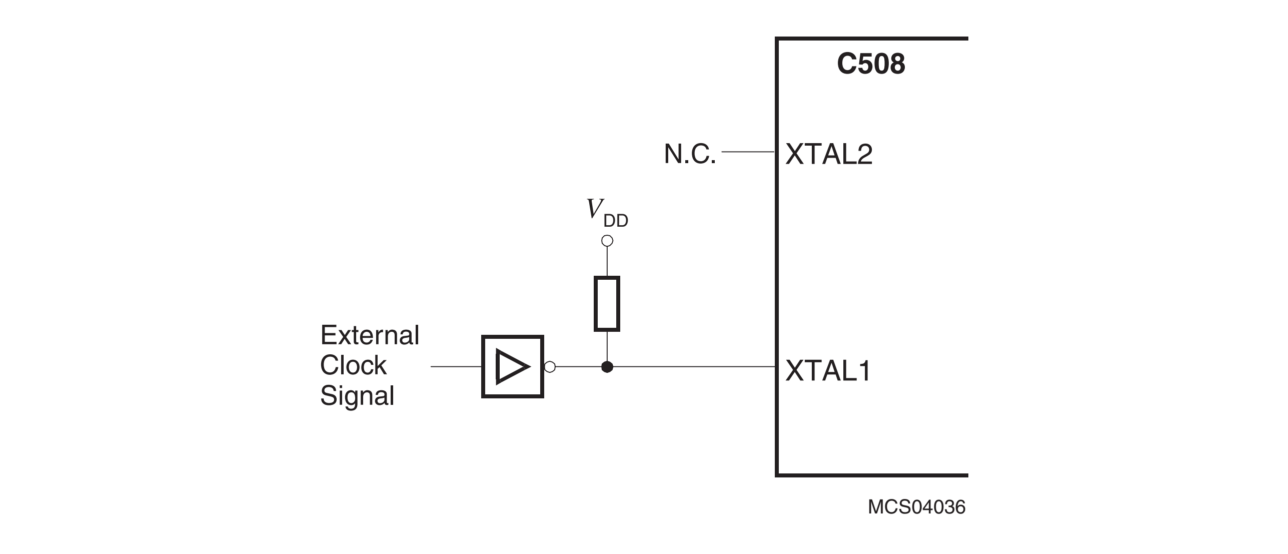


Figure 5-8 External Clock Source

6 On-Chip Peripheral Components

This chapter provides detailed information about all on-chip peripherals of the C508 except for the integrated interrupt controller, which is described separately in **Chapter 7**.

6.1 Parallel I/O

The C508 has one 8-bit analog or digital input port and five 8-bit I/O ports. Port 4 is a uni-directional input port. Port 0 is an open-drain bi-directional I/O port; Ports 1, 2, 3, and 5 are quasi-bi-directional I/O ports with internal pull-up transistors. This means that when these ports are configured as inputs, they will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

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

Port 4 provides the analog input channels to the A/D Converter.

6.1.1 Port Structures

The C508 generally allows digital I/O on 32 lines grouped into four bi-directional 8-bit ports and analog/digital input on one unidirectional 8-bit port. Except for Port 4 which is the uni-directional input port, 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 (except P4) are performed via their corresponding Special Function Registers.

When Port 4 is used as analog input, an analog channel is switched to the A/D Converter through a 3-bit multiplexer, which is controlled by three bits in SFR ADCON (see **Chapter 6.5**). Port 4 lines may also be used as digital inputs. In this case, they are addressed as an input port via SFR P4. Since Port 4 has no internal latch, the contents of SFR P4 only depend on the levels applied to the input lines. It makes no sense to output a value to this input-only port by writing to the SFR P4. This will have no effect.

The parallel I/O ports of the C508 can be grouped into four different types which are listed in **Table 6-1**.

Table 6-1 C508 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	Digital/analog uni-directional input port.
D	Standard digital I/O with push-pull drive capability.

Type A and B port pins are standard C501 compatible I/O port lines which can be used for digital I/O. Type A port (Port 0) is also designed for accessing external data or program memory. Type B port lines are located at Port 2, Port 3, and Port 5 to provide alternate functions for the serial interface, LED drive interface, and PWM signals; or are used as control outputs during external data memory accesses. Type C port (Port 4) provides the analog input port. Type D port lines can be switched to push-pull drive capability when they are used as compare outputs of the CAPCOM unit.

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

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

Port	Alternate Function	Port Type	Function
P1.0	COUT3	D	10-bit compare channel output
P1.1	$\overline{\text{CTRAP}}$	D	CCU trap input
P1.2	CC0	D	CAPCOM Channel 0 input/output
P1.3	COUT0	D	CAPCOM Channel 0 output
P1.4	CC1	D	CAPCOM Channel 1 input/output
P1.5	COUT1	D	CAPCOM Channel 1 output
P1.6	CC2	D	CAPCOM Channel 2 input/output
P1.7	COUT2	D	CAPCOM Channel 2 output
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
P3.3	$\overline{\text{INT1}}$	B	External interrupt 1 input
P3.4	T0	B	Timer 0 external counter input
P3.5	T1	B	Timer 1 external counter input
P3.6	$\overline{\text{WR}}$	B	External data memory write strobe
P3.7	$\overline{\text{RD}}$	B	External data memory read strobe
P5.0	T2CC0 / $\overline{\text{INT3}}$	B	T2 Compare/Capture output 0/External interrupt 3 input
P5.1	T2CC1 / INT4	B	T2 Compare/Capture output 1/External interrupt 4 input
P5.2	T2CC2 / INT5	B	T2 Compare/Capture output 2/External interrupt 5 input
P5.3	T2CC3/INT6	B	T2 Compare/Capture output 3/External interrupt 6 input
P5.4	INT2	B	External interrupt 2 input
P5.5	INT9	B	External interrupt 9 input
P5.6	INT8	B	External interrupt 8 input
P5.7	INT7	B	External interrupt 7 input

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 make up 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 which will clock in a value from the internal bus in response to a "write-to-latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read-latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read-pin" signal from the CPU. Some instructions that read from a port (that is, from the corresponding port SFR P0 to P4) activate the "read-latch" signal, while others activate the "read-pin" signal.

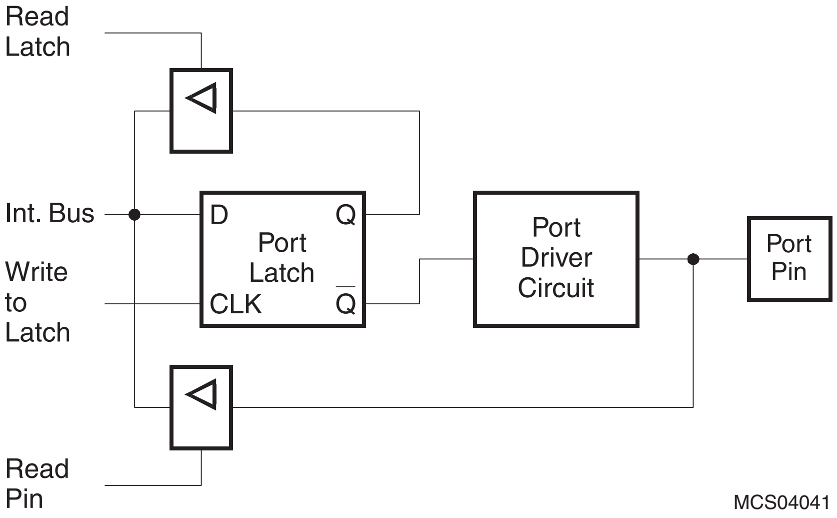


Figure 6-1 Basic Structure of a Port Circuitry

The output drivers of Port 1, 2, 3, and 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 a one (1). This means for **Figure 6-2**, $\overline{Q}$ = 0, which turns off the output driver FET n1. Then, for Ports 1 to 5, except Port 4, 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-bi-directional”.

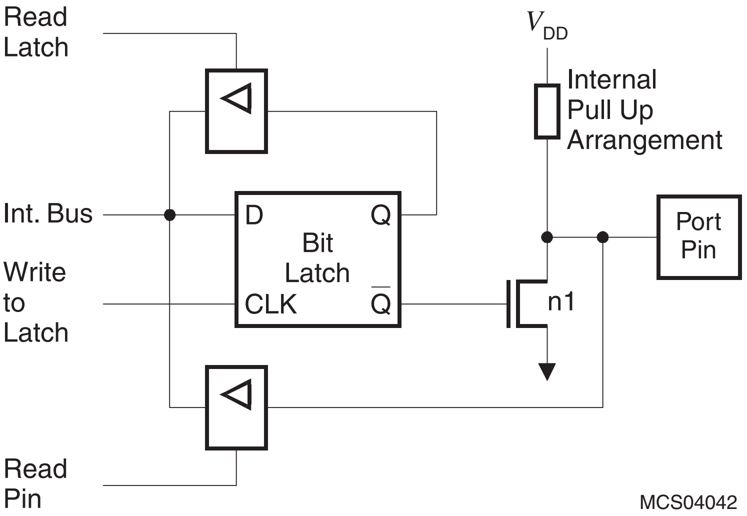


Figure 6-2 Basic Output Driver Circuit of Ports 1, 2, 3, and 5

6.1.2.1 Port 0 Circuitry

Port 0, in contrast to Ports 1 to 5, is considered a “true” bidirectional port because its 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 ‘1’s during the external memory accesses. Otherwise, the pull-up is always off. Consequently, P0 lines that are used as output port lines are open drain lines. Writing a ‘1’ to the port latch leaves both output FETs off and the pin floats. In this condition, it can be used as high-impedance input. If Port 0 is configured as a general I/O port and must emit logic high-level (1); then, external pull-ups are required.

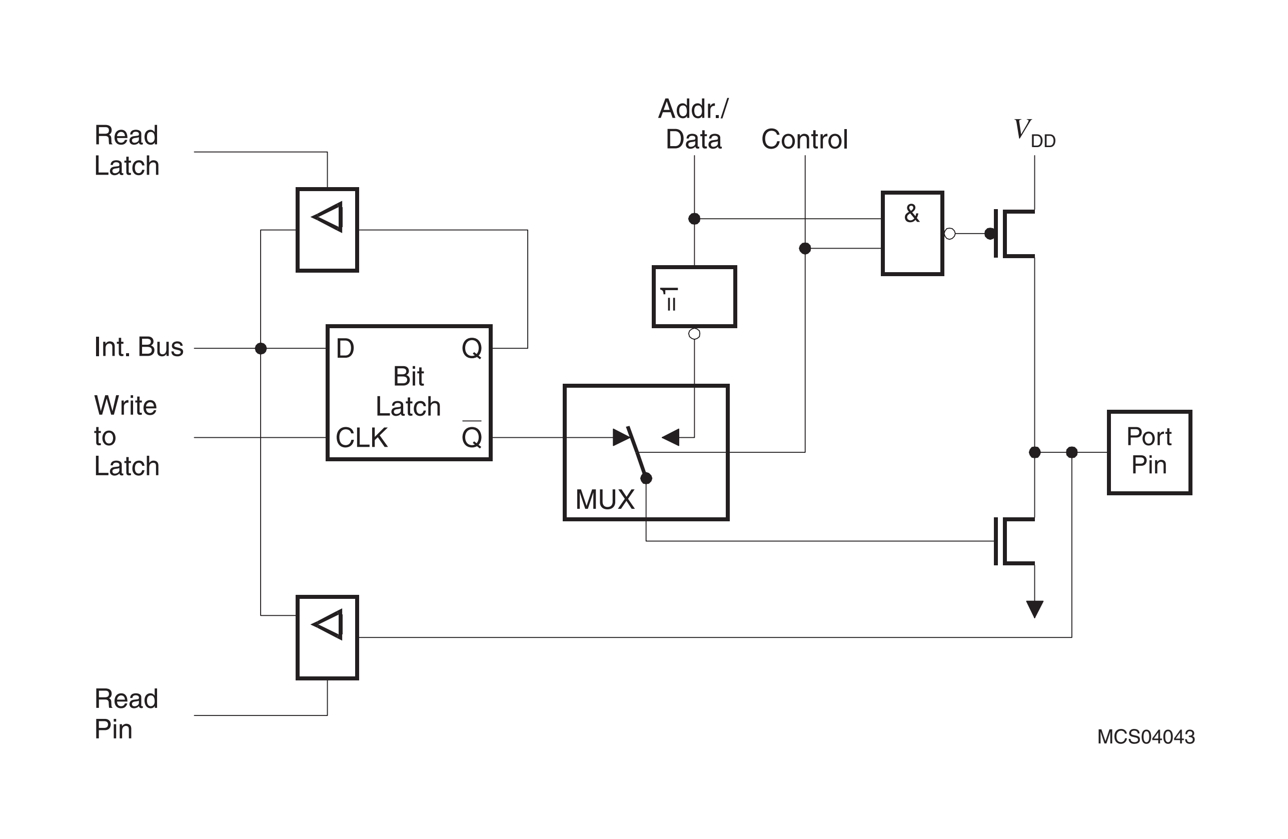


Figure 6-3 Port 0 Circuitry

6.1.2.2 Port 1, Port 3, and Port 5 Circuitry

The pins of Ports 1, 3, and 5 are multifunctional. They are port pins and also serve to implement 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 must contain a one (1); otherwise, the pull-down FET is on and the port pin is stuck at ‘0’. After reset, all port latches contain ones (1).

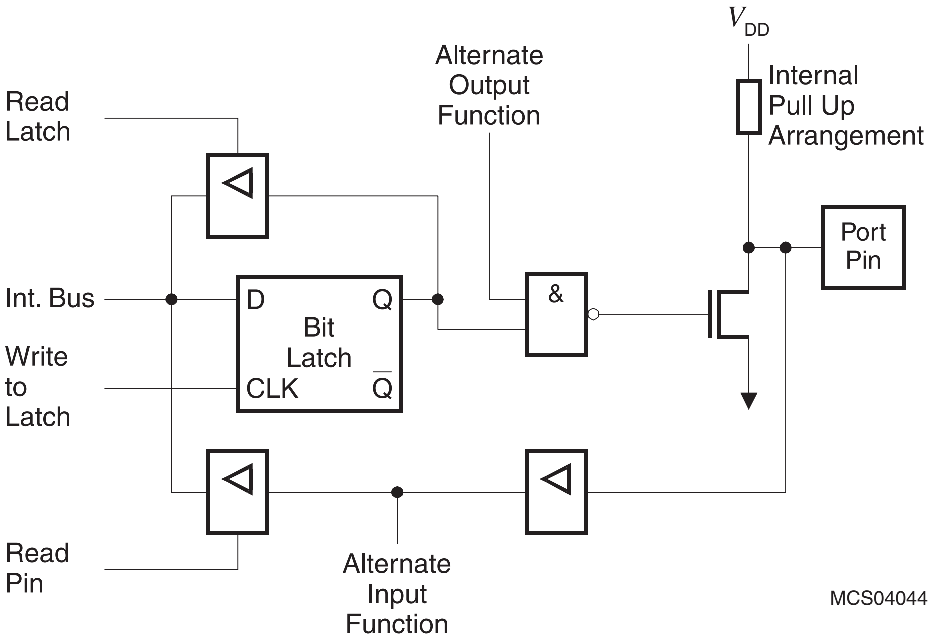


Figure 6-4 Ports 1, 3, and 5 Circuitry

6.1.2.3 Port 2 Circuitry

As shown in **Figure 6-3** and in **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 two 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 $\overline{\text{EA}}$ pin and/or the contents of the program counter. If the ports are configured as an address/data bus, the port latches are disconnected from the driver circuit. During this time, the P0/P2 SFR remains unchanged. As 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-ups p1 (**Figure 6-5a**) to emit ‘1’s for the entire external memory cycle instead of the weak ones (p2 and p3) used during normal port activity.

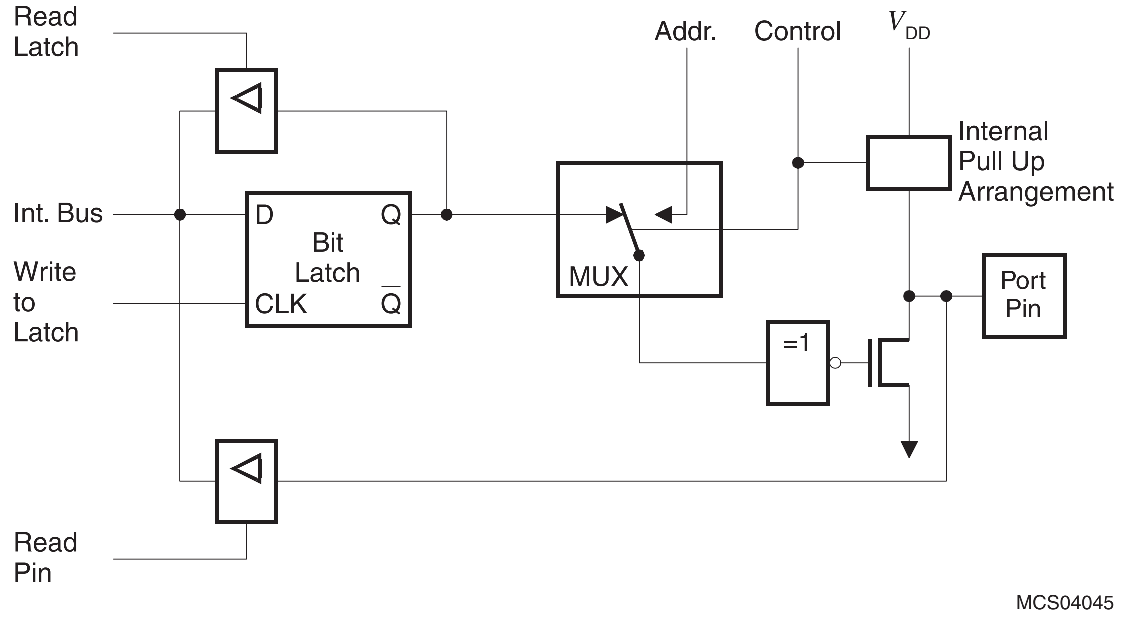


Figure 6-5 Port 2 Circuitry

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

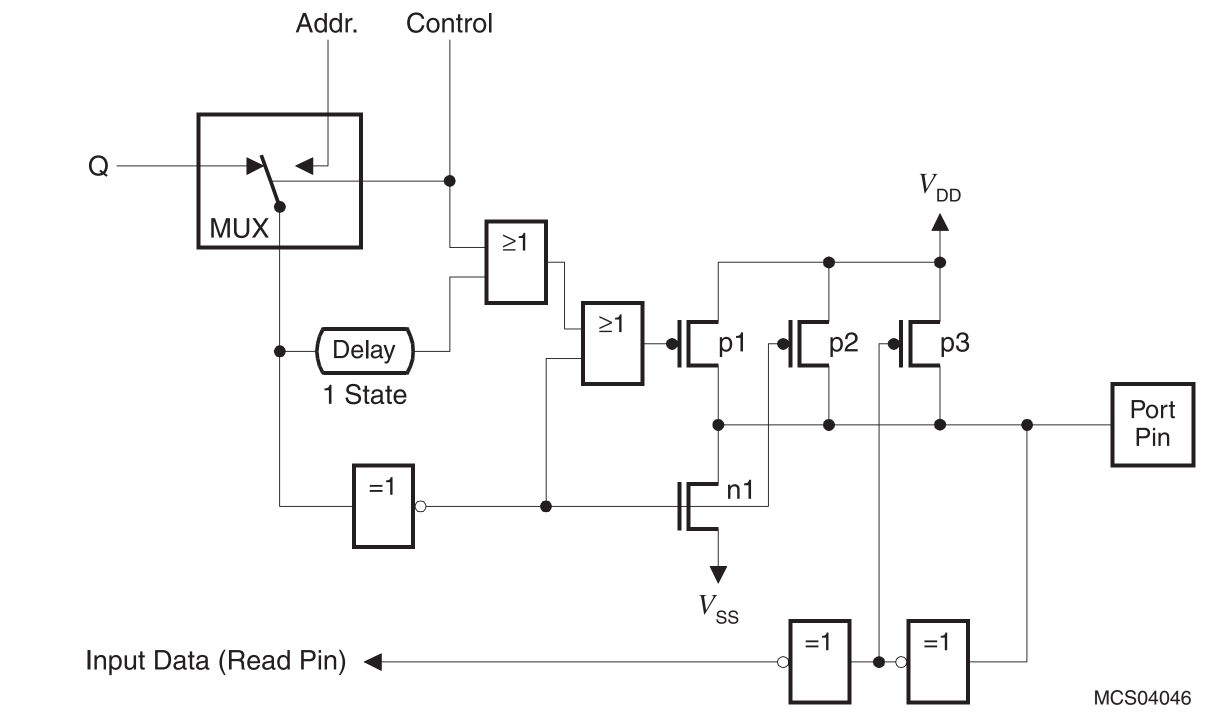


Figure 6-5a Port 2 Pull-up Arrangement

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

6.1.3 Detailed Output Driver Circuitry

In fact, the pull-ups mentioned before and included in **Figure 6-2**, **Figure 6-4**, and **Figure 6-5** are pull-up arrangements. The differences which apply to the various port types available in the C508 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 pull-down FET and three pull-up FETs:

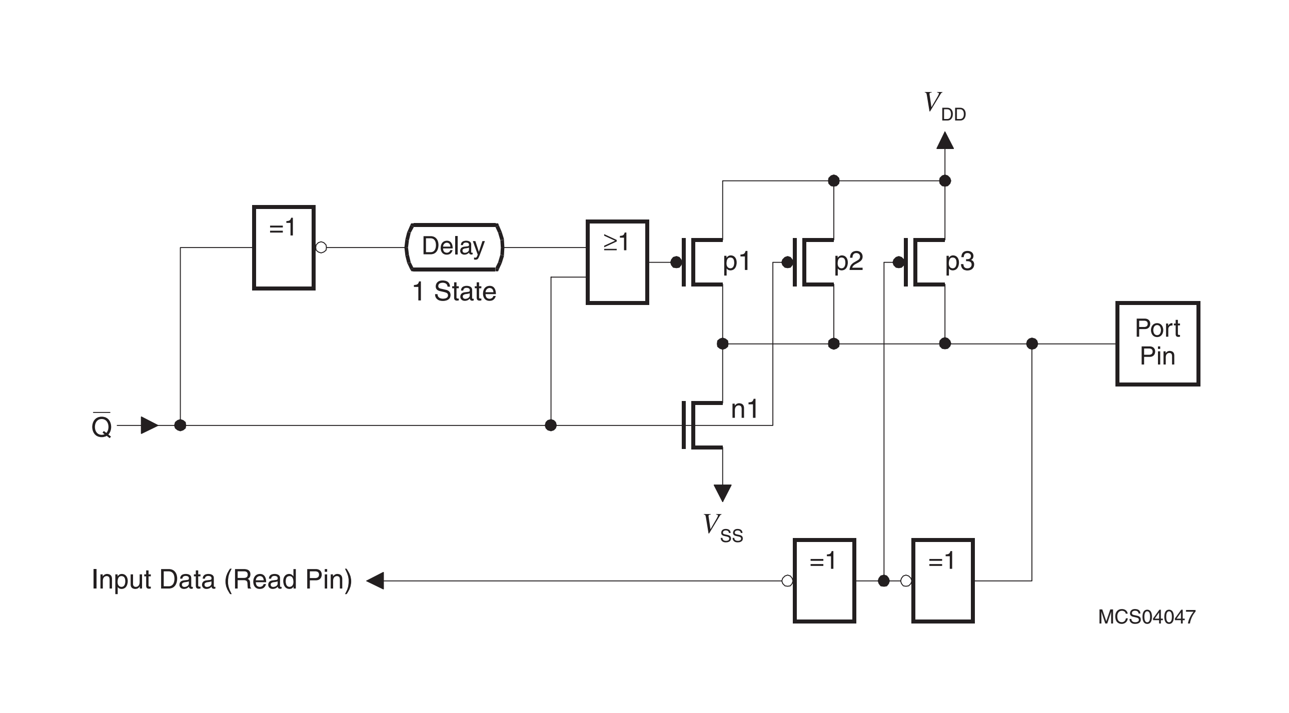


Figure 6-6 Driver Circuit of Type B Port Pins

- The **pull-down FET n1** is an n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a ‘0’ is programmed to the port pin. A short circuit to V_{DD} must be avoided if the transistor is turned on, since the high current might destroy the FET. This also means that no ‘0’ must be programmed into the latch of a pin that is used as input.
- The **pull-up FET p1** is a 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; that is, a ‘1’ is programmed to the port latch which contained a ‘0’. The extra pull-up can drive a similar current as the pull-down FET n1. This provides a fast transition of the logic levels at the pin.
- The **pull-up FET p2** is a 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, for example, when used as input with logic low input level.

- The **pull-up FET p3** is a p-channel type. It is activated only if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pull-up current if a logic high level shall be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level; for example, when used as input. In this configuration, only the weak pull-up FET p2 is active, which sources the current I_{IL}. If, in addition, the pull-up FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, additional power consumption can be avoided if port pins are used as inputs with a low level applied. However, the driving capability is stronger if a logic high level is output.

The activating and deactivating of the four different transistors translates into four states possible for the pins:

- 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 a ‘0’, the pin will be in OL state.

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

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

The same is true if a pin is used as a bi-directional 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 D Port Driver Circuitry

The driver and control structure of the port pins used for Compare output functions have a port structure which allows a true push-pull output driving capability (Type D). This output driver characteristic is only enabled/used when the corresponding port lines are used as Compare outputs.

The push-pull port structure is illustrated in **Figure 6-7**.

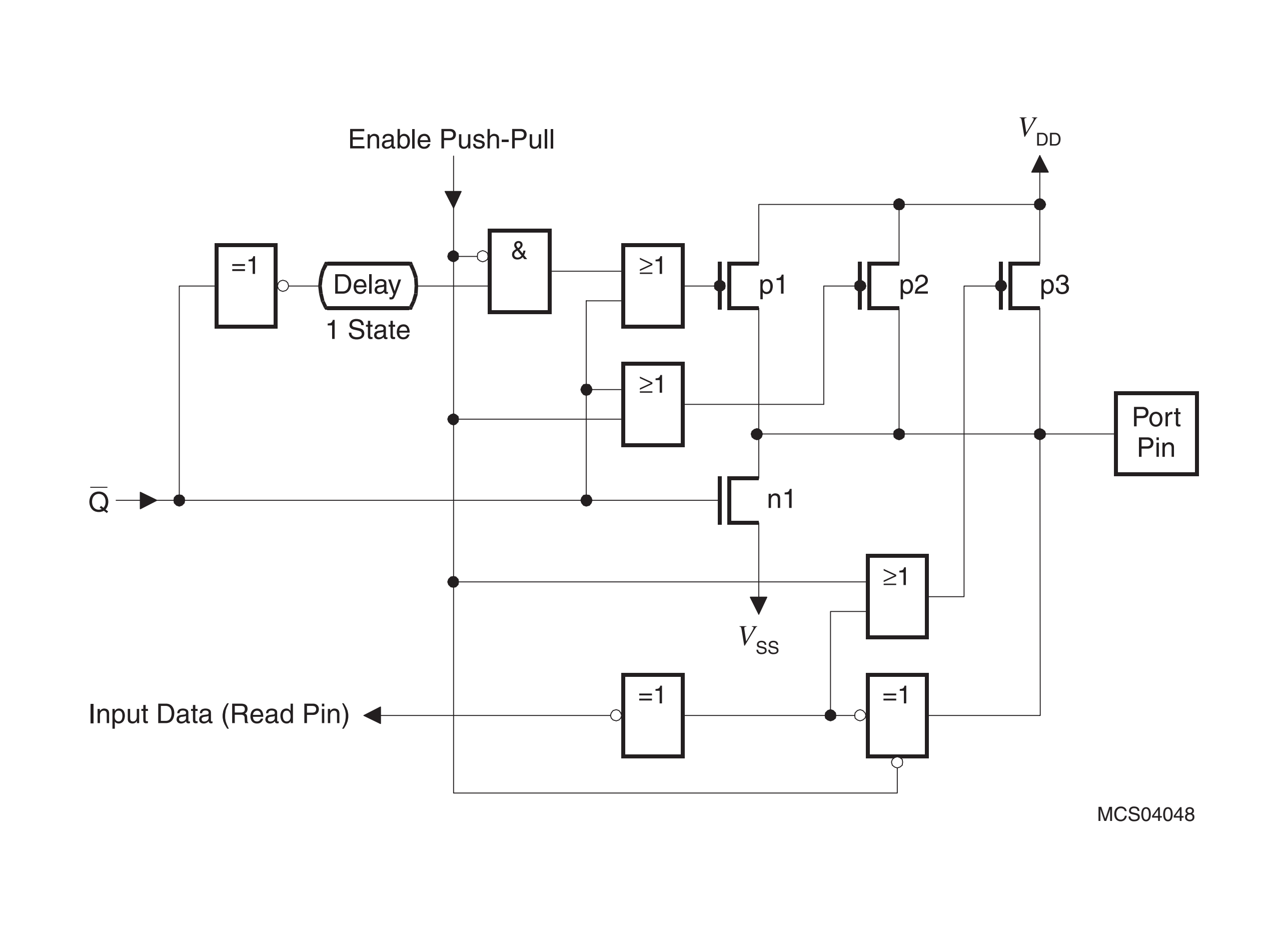


Figure 6-7 Driver Circuit of Type D Port Pins

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 (that is: MOV A, P1), the port pin is actually sampled in State 5 Phase 1 or Phase 2, depending on the port and the alternative functions. **Figure 6-8** illustrates this port timing. It must be noted that this mechanism of sampling once per machine cycle is also used if a port pin is to detect an “edge”. For example, when used as 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 on the pulse length of signals in order to avoid signal “edges” not being detected. The minimum time period of high and low level is one machine cycle, which guarantees that this logic level is noticed by the port at least once.

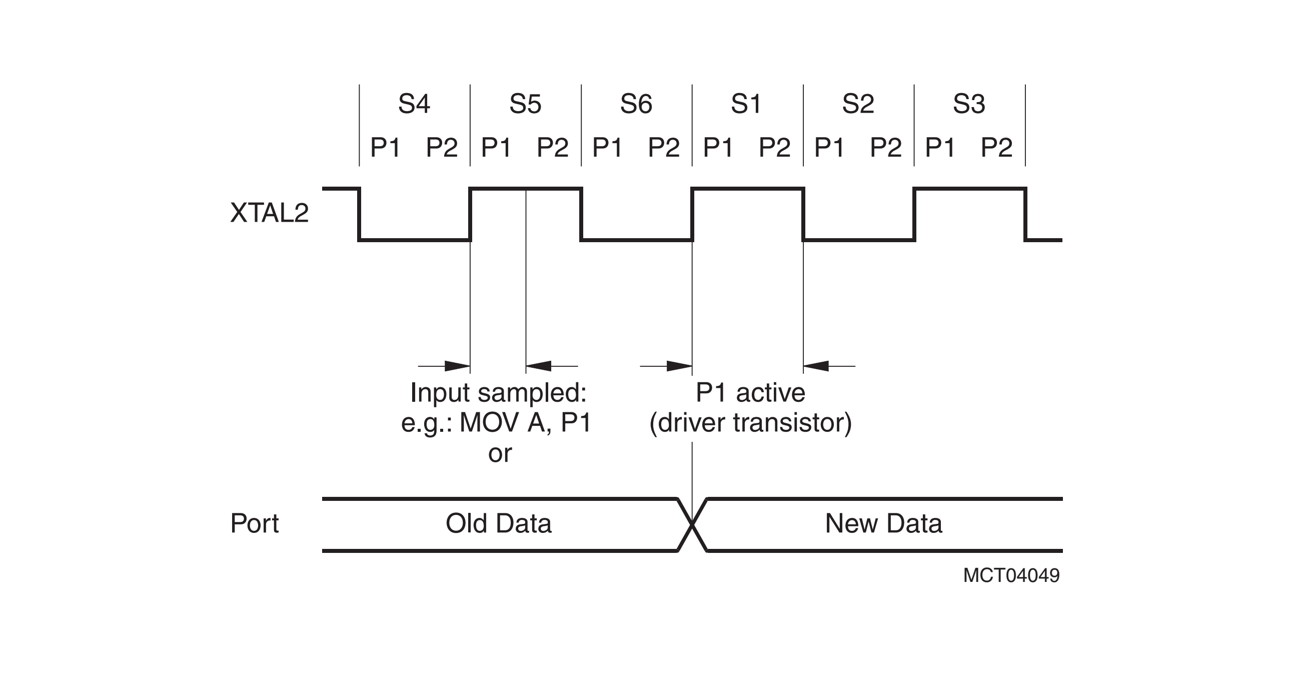


Figure 6-8 Port Timing

6.1.5 Port Loading and Interfacing

The output buffers of Ports 1 to 5, except Port 4 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can be looked up in the DC characteristics in the Data Sheet of the C508. The corresponding parameters are V_{OL} and V_{OH}.

The same 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, it must be noted that the Ports 1 to 5 (except Port 4) 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 be applied to the port pin (the parameters I_{TL} and I_{IL} in the DC characteristics specify these currents). Port 0 however, has floating inputs when used for digital input.

6.1.6 Read-Modify-Write Feature of Ports 0 to 5 (Except Port 4)

Some port-reading instructions read the latch and others read the pin. The instructions reading the latch read a value, possibly change it, and then rewrite it to the latch. These are called “read-modify-write” instructions. They 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 which can be used to read a port, exclusively read the port pin. In any case, reading from latch or pin, respectively, 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 may not be obvious that the last three instructions in **Table 6-3** are read-modify-write instructions, but they are. This is because they read the port byte, all 8 bits, 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; for example: ANL P1, A
ORL	Logic OR; for example: ORL P2, A
XRL	Logic exclusive OR; for example: XRL P3, A
JBC	Jump if bit is set and clear bit; for example: JBC P1.1, LABEL
CPL	Complement bit; for example: CPL P3.0
INC	Increment byte; for example: INC P4
DEC	Decrement byte; for example: DEC P5
DJNZ	Decrement and jump if not zero; for example: 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 to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a ‘1’ is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor (approx. 0.7 V, that is, a logic low level!) and interpret it as ‘0’. For example, when modifying a port bit by a SETB or CLR instruction, another bit in this port with the above mentioned configuration might be changed if the value read from the pin were written back to the latch. However, reading the latch rater than the pin will return the correct value of ‘1’.

6.2 Timers/Counters

The C508 contains three 16-bit timers/counters, Timer 0, Timer 1 and Timer 2. They are useful in many applications for timing and counting.

In its “timer” function, the timer register is incremented every machine cycle. Thus, it can be thought of as counting machine cycles. Since a machine cycle consists of three oscillator periods, the counter rate is 1/3 of the oscillator frequency.

In its “counter” function, the timer register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 or P3.5 respectively). In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (six oscillator periods) to recognize a transition from 1-to-0, the maximum count rate is 1/6 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

6.2.1 Timer/Counter 0 and 1

Timer/counter 0 and 1 of the C508 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. These may be combined into one timer configuration depending on the mode that is established. The functions of the timers are controlled by two Special Function Registers TCON and TMOD.

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

6.2.1.1 Timer/Counter 0 and 1 Registers

Seven special function registers control the operation of timer/counter 0 and 1:

- TL0/TH0 and TL1/TH1 – counter registers, low and high part
- TCON and IEN0 – control and interrupt enable
- TMOD – mode select

Special Function Register TL0 (Address 8A$_H$) **Reset Value: 00$_H$**
Special Function Register TH0 (Address 8C$_H$) **Reset Value: 00$_H$**
Special Function Register TL1 (Address 8B$_H$) **Reset Value: 00$_H$**
Special Function Register TH1 (Address 8D$_H$) **Reset Value: 00$_H$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
8A$_H$	.7	.6	.5	.4	.3	.2	.1	.0	TL0
8C$_H$	.7	.6	.5	.4	.3	.2	.1	.0	TH0
8B$_H$	.7	.6	.5	.4	.3	.2	.1	.0	TL1
8D$_H$	.7	.6	.5	.4	.3	.2	.1	.0	TH1

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

Special Function Register TCON (Address 88_H) **Reset Value: 00_H**
Special Function Register IEN0 (Address $A8_H$) **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
	AF_H	AE_H	AD_H	AC_H	AB_H	AA_H	$A9_H$	$A8_H$	
$A8_H$	▨ EA	▨ WDT	▨ ET2	▨ ES0	ET1	▨ EX1	ET0	▨ EX0	IEN0

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

Bit	Function
TF1	**Timer 1 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.
TF0	**Timer 0 overflow flag** Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	**Timer 0 run control bit** Set/cleared by software to turn timer/counter 0 ON/OFF.
ET1	**Timer 1 overflow interrupt enable** If ET1 = 0, the timer 1 interrupt is disabled.
ET0	**Timer 0 overflow interrupt enable** If ET0 = 0, the timer 0 interrupt is disabled.

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

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

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>GATE</td><td><b>Gating control</b>
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.</td></tr>
<tr><td>C/$\overline{T}$</td><td><b>Counter or timer select bit</b>
Set for counter operation (input from “Tx” input pin).
Cleared for timer operation (input from internal system clock).</td></tr>
<tr><td>M1
M0</td><td><b>Mode select bits</b>
<table>
<tr><th>M1</th><th>M0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>8-bit timer/counter:
“THx” operates as 8-bit timer/counter
“TLx” serves as 5-bit prescaler</td></tr>
<tr><td>0</td><td>1</td><td>16-bit timer/counter.
“THx” and “TLx” are cascaded; there is no prescaler</td></tr>
<tr><td>1</td><td>0</td><td>8-bit auto-reload timer/counter.
“THx” holds a value which is to be reloaded into “TLx” each time it overflows</td></tr>
<tr><td>1</td><td>1</td><td>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</td></tr>
</table>
</td></tr>
</table>

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 6-9** shows the mode 0 operation.

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

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

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0, TH0, TL0 and $\overline{\text{INT0}}$ for the corresponding Timer 1 signals in **Figure 6-9**. There are two different gate bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

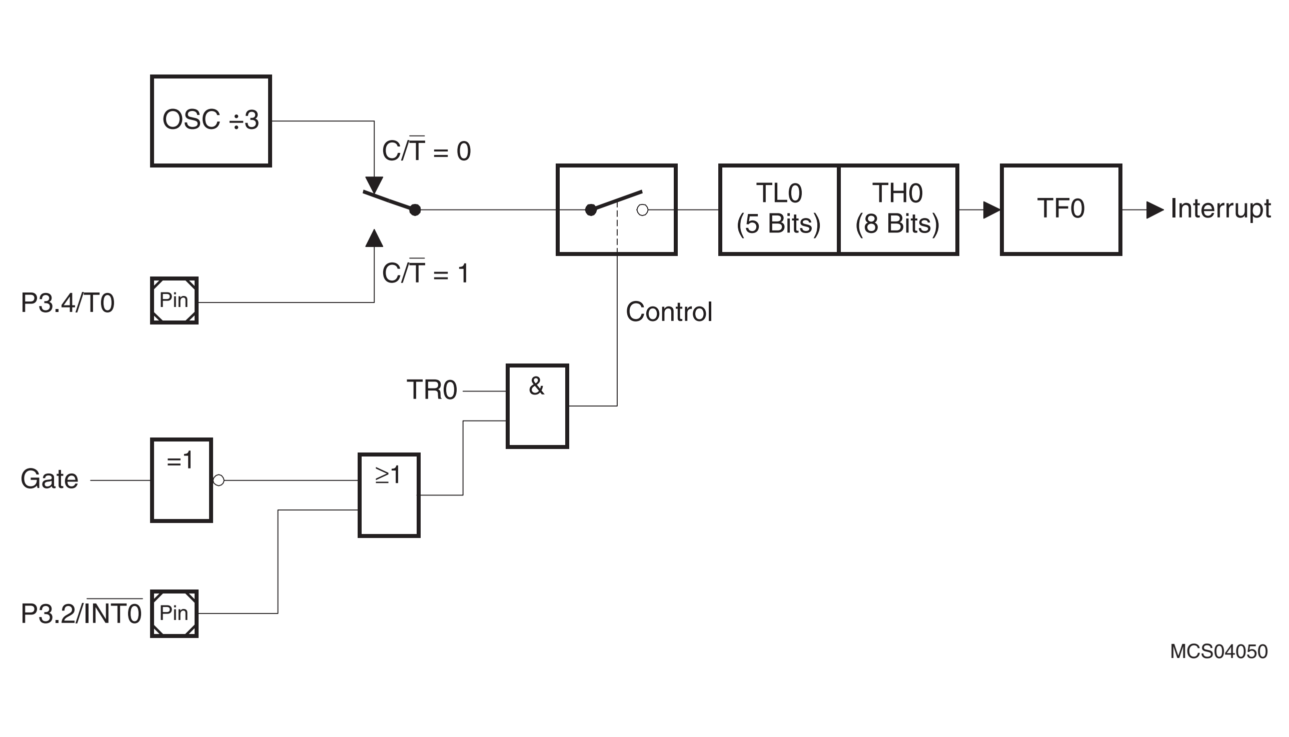


Figure 6-9 Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

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 6-10**.

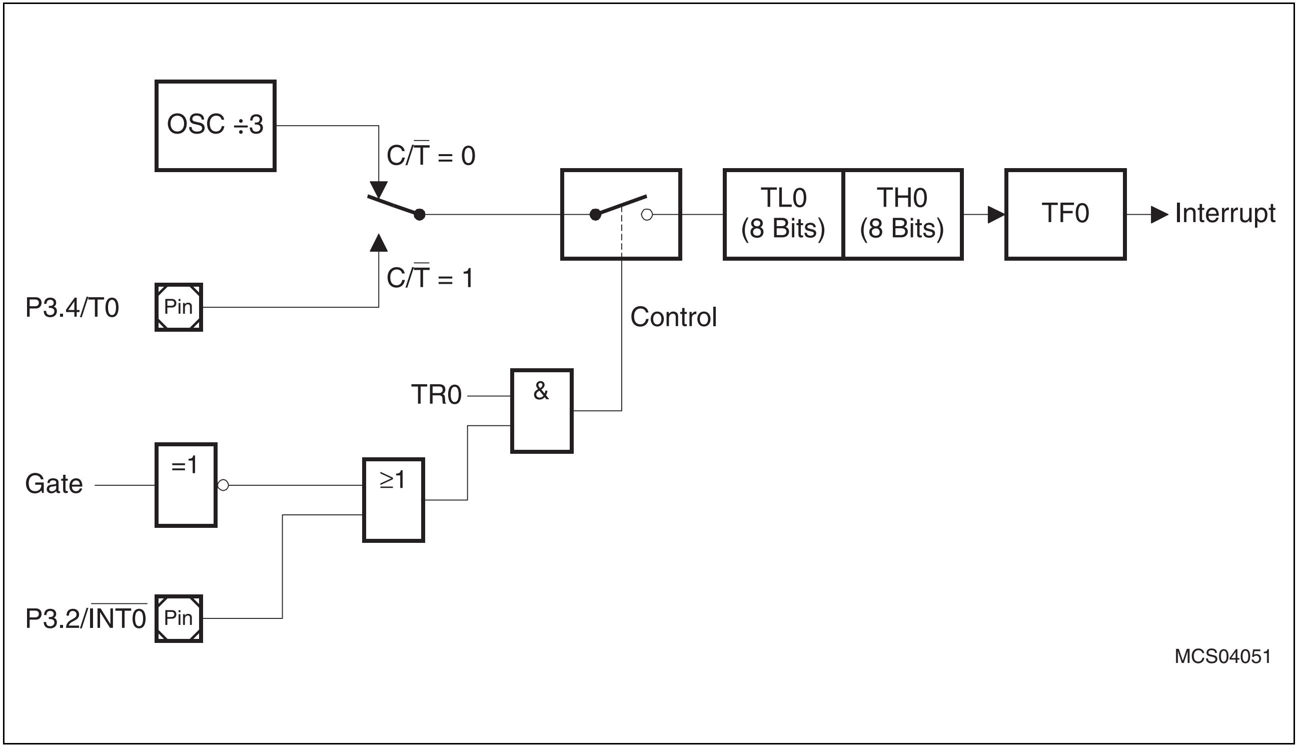


Figure 6-10 Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

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 6-11**. 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.

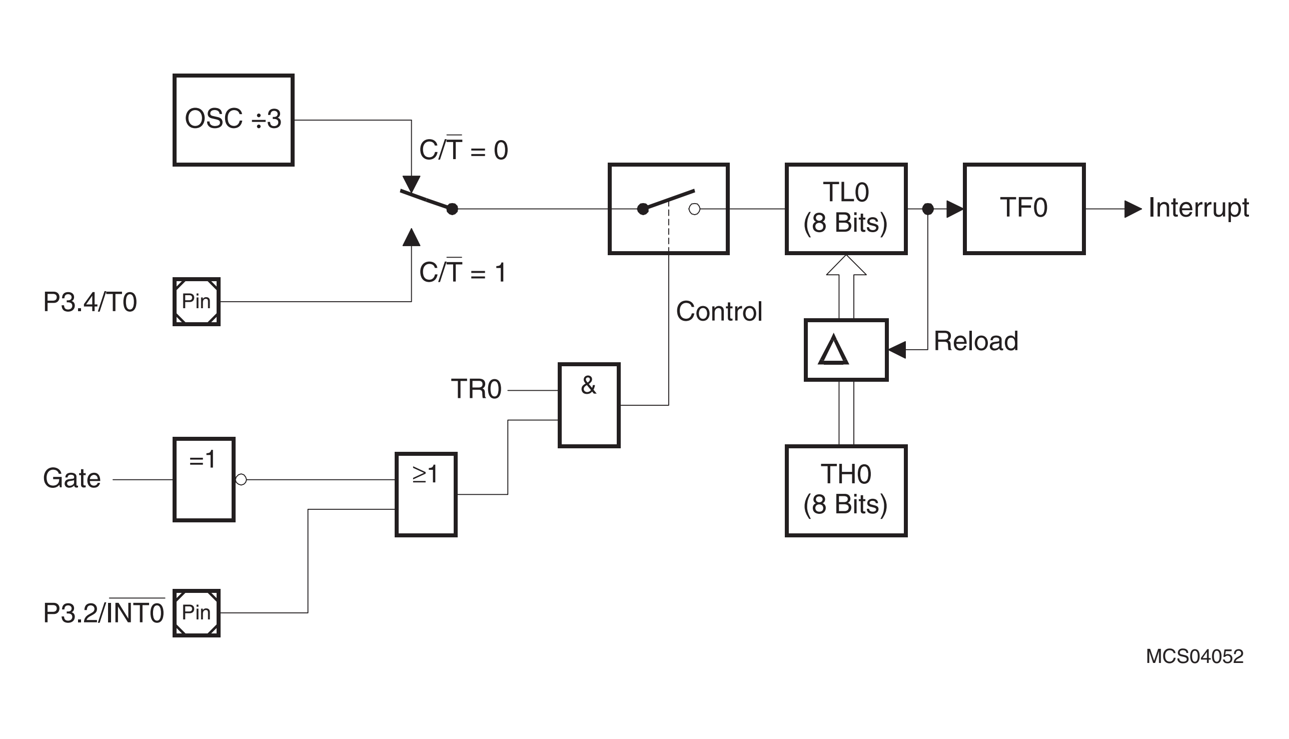


Figure 6-11 Timer/Counter 0, 1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

6.2.1.5 Mode 3

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in mode 3 establishes TL0 and TH0 as two separate counters. The logic for mode 3 on Timer 0 is shown in **Figure 6-12**. TL0 uses the Timer 0 control bits: C/$\overline{\text{T}}$, Gate, TR0, $\overline{\text{INT0}}$ 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 and when TR1 is set, Timer 1 can be turned on by switching it to any mode other than 3 and off by switching it into its own mode 3, or can still be used by the serial channel as a baudrate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

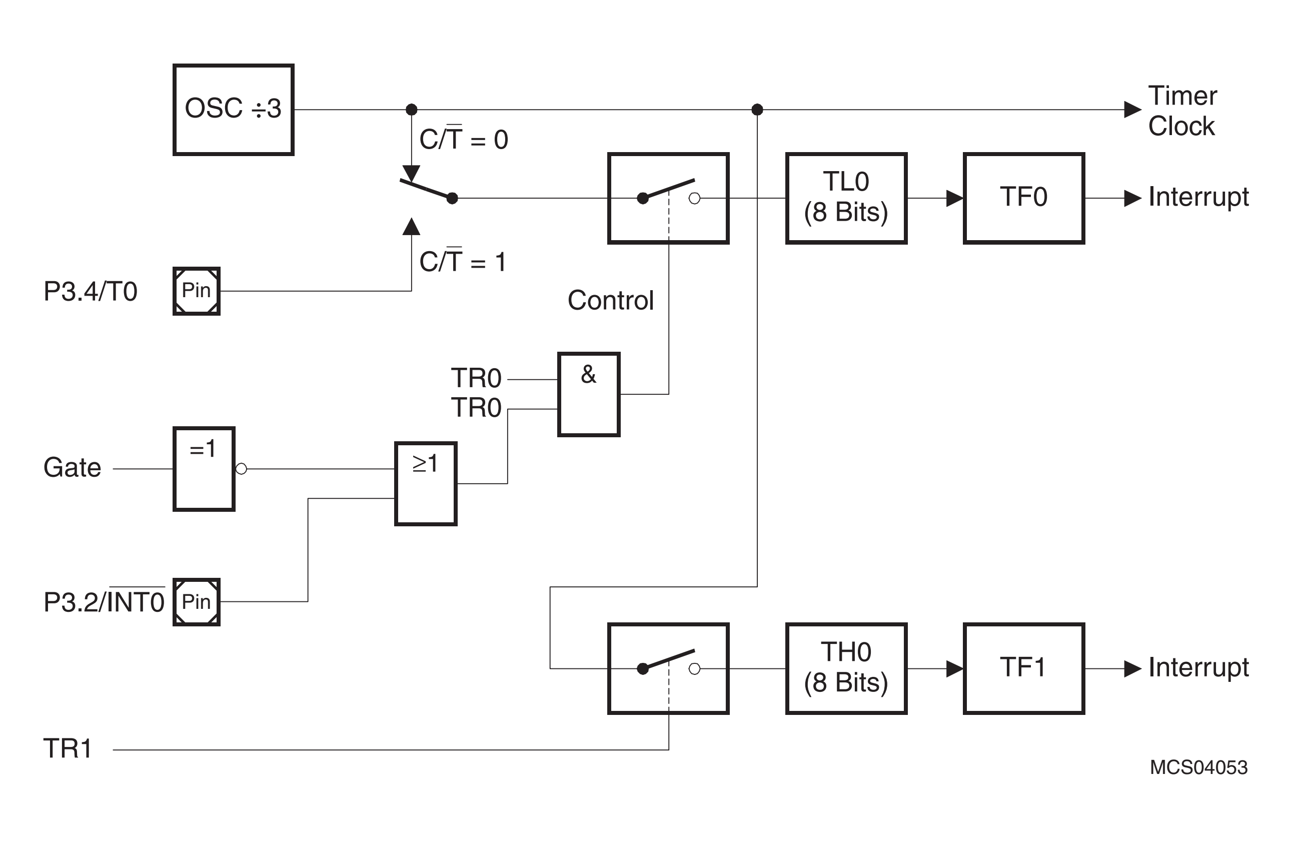


Figure 6-12 Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

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

Timer 2 with additional Compare/Capture/reload features is one of the most powerful peripheral units of the C508. It can be used for various digital signal generation and event capturing like pulse generation, pulse width modulation, pulse width measuring etc.

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

The C508 Timer 2 used in combination with the Compare/Capture/reload registers allows the following operating modes:

– 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 6-13** shows the general configuration of Timer 2 with the additional Compare/Capture/reload registers. The I/O pins which can be used for Timer 2 control are located as multifunctional port functions at Port 5 (see **Figure 6-4**).

Table 6-4 Alternate Port Functions of Timer 2

Pin Symbol	Function
P5.0 / T2CC0 / $\overline{\text{INT3}}$	Compare output/Capture input for CRC Register
P5.1 / T2CC1 / INT4	Compare output/Capture input for CC Register 1
P5.2 / T2CC2 / INT5	Compare output/Capture input for CC Register 2
P5.3 / T2CC3 / INT6	Compare output/Capture input for CC Register 3

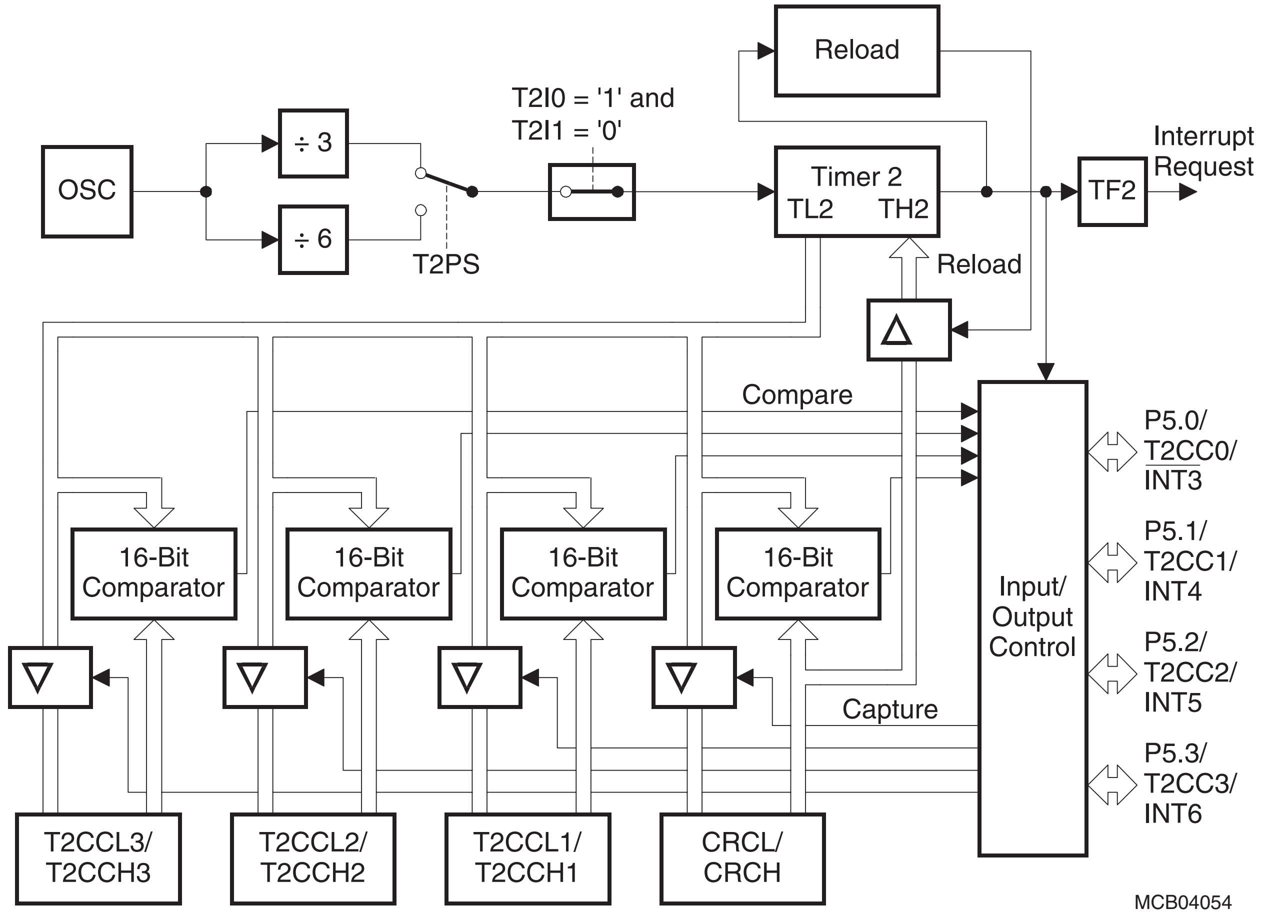


Figure 6-13 Timer 2 Block Diagram

6.2.2.1 Timer 2 Registers

This section describes all Timer 2 related special function registers. The interrupt related SFRs are also included in this section. **Table 6-5** summarizes the Timer 2 SFRs.

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

Symbol	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
T2CCL1	Compare/Capture Register 1, Low Byte	$C2_H$
T2CCH1	Compare/Capture Register 1, High Byte	$C3_H$
T2CCL2	Compare/Capture Register 2, Low Byte	$C4_H$
T2CCH2	Compare/Capture Register 2, High Byte	$C5_H$
T2CCL3	Compare/Capture Register 3, Low Byte	$C6_H$
T2CCH3	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, T2CC1 to T2CC3.

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

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

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

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>T2PS</td><td><b>Prescaler select bit</b>
When set, Timer 2 is clocked with 1/6 of the oscillator frequency. When cleared, Timer 2 is clocked with 1/3 of the oscillator frequency.</td></tr>
<tr><td>I3FR</td><td><b>External interrupt 3 falling / rising edge flag</b>
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 P5.0/T2CC0/$\overline{\text{INT3}}$.</td></tr>
<tr><td>T2R1
T2R0</td><td><b>Timer 2 Reload enable</b>
<table>
<tr><th>T2R0</th><th>T2R1</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Reload disabled</td></tr>
<tr><td>0</td><td>1</td><td>Auto-Reload upon Timer 2 overflow (TF2)</td></tr>
<tr><td>1</td><td>X</td><td>Prohibited. Do not use this combination.</td></tr>
</table></td></tr>
<tr><td>T2CM</td><td><b>Compare mode bit for registers CRC, T2CC1 through T2CC3</b>
<table>
<tr><th>T2CM</th><th>Function</th></tr>
<tr><td>0</td><td>Compare mode 0 is selected</td></tr>
<tr><td>1</td><td>Compare mode 1 is selected</td></tr>
</table></td></tr>
<tr><td>T2I1
T2I0</td><td><b>Timer 2 input selection</b>
<table>
<tr><th>T2I0</th><th>T2I1</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>No input selected, Timer 2 stops</td></tr>
<tr><td>1</td><td>0</td><td>Timer function:
input frequency = $f_{osc}/3$ (T2PS = 0)
or $f_{osc}/6$ (T2PS = 1)</td></tr>
<tr><td>X</td><td>1</td><td>Prohibited. Do not use this combination.</td></tr>
</table></td></tr>
</table>

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							LSB	
	7	6	5	4	3	2	1	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 low byte of the 16-bit Timer 2 count value.
TH2.7-0	**Timer 2 Value High Byte** The TH2 register holds the high byte of the 16-bit Timer 2 count value.
CRCL.7-0	**Reload Register Low Byte** CRCL is the 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 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 IRCON (Address C0$_H$) **Reset Value: X0000000$_B$**

	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

Bit No.	C7$_H$	C6$_H$	C5$_H$	C4$_H$	C3$_H$	C2$_H$	C1$_H$	C0$_H$	
C0$_H$	–	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

▨ The shaded bits are not used in Timer/counter 2 interrupt control.

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.
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

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>COCAH3
COCAL3</td><td><b>Compare/Capture mode for CC register 3</b>
<table>
<tr><th>COCAH3</th><th>COCAL3</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare/Capture disabled</td></tr>
<tr><td>0</td><td>1</td><td>Capture on rising edge at pin P5.3 / T2CC3 / INT6</td></tr>
<tr><td>1</td><td>0</td><td>Compare enabled</td></tr>
<tr><td>1</td><td>1</td><td>Capture on write operation into register CCL3</td></tr>
</table></td></tr>
<tr><td>COCAH2
COCAL2</td><td><b>Compare/Capture mode for CC register 2</b>
<table>
<tr><th>COCAH2</th><th>COCAL2</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare/Capture disabled</td></tr>
<tr><td>0</td><td>1</td><td>Capture on rising edge at pin P5.2 / T2CC2 / INT5</td></tr>
<tr><td>1</td><td>0</td><td>Compare enabled</td></tr>
<tr><td>1</td><td>1</td><td>Capture on write operation into register CCL2</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>COCAH1
COCAL1</td><td><b>Compare/Capture mode for CC register 1</b>
<table>
<tr><th>COCAH1</th><th>COCAL1</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare/Capture disabled</td></tr>
<tr><td>0</td><td>1</td><td>Capture on rising edge at pin P5.1 / T2CC1 / INT4</td></tr>
<tr><td>1</td><td>0</td><td>Compare enabled</td></tr>
<tr><td>1</td><td>1</td><td>Capture on write operation into register CCL1</td></tr>
</table></td></tr>
<tr><td>COCAH0
COCAL0</td><td><b>Compare/Capture mode for CRC register</b>
<table>
<tr><th>COCAH0</th><th>COCAL0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare/Capture disabled</td></tr>
<tr><td>0</td><td>1</td><td>Capture on falling/rising edge at pin P5.0 / T2CC0 / $\overline{\text{INT3}}$</td></tr>
<tr><td>1</td><td>0</td><td>Compare enabled</td></tr>
<tr><td>1</td><td>1</td><td>Capture on write operation into register CRCL</td></tr>
</table></td></tr>
</table>

6.2.2.2 Timer 2 Operation

Timer 2, which is a 16-bit-wide register, operates as a timer with its count rate derived from the oscillator frequency. A prescaler offers the possibility of selecting a count rate of 1/3 or 1/6 of the oscillator frequency. Thus, the 16-bit timer register (consisting of TH2 and TL2) is either incremented in every machine cycle or in every second machine cycle. The prescaler is selected by bit T2PS in special function register T2CON. If T2PS is cleared, the input frequency is 1/3 of the oscillator frequency. If T2PS is set, the 2:1 prescaler gates 1/6 of the oscillator frequency to the timer.

The timer overflow flag TF2 in SFR IRCON is set when there is a roll-over of the count from all ‘1’s to all ‘0’s. The flag TF2 can generate an interrupt and it must be cleared by the interrupt service routine.

Reload of Timer 2

The reload mode for Timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON. To enable the reload mode, bit T2R0 must be cleared and bit T2R1 set. Figure 6-14 shows the configuration of Timer 2 in reload mode.

When Timer 2 rolls over from all ‘1’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.

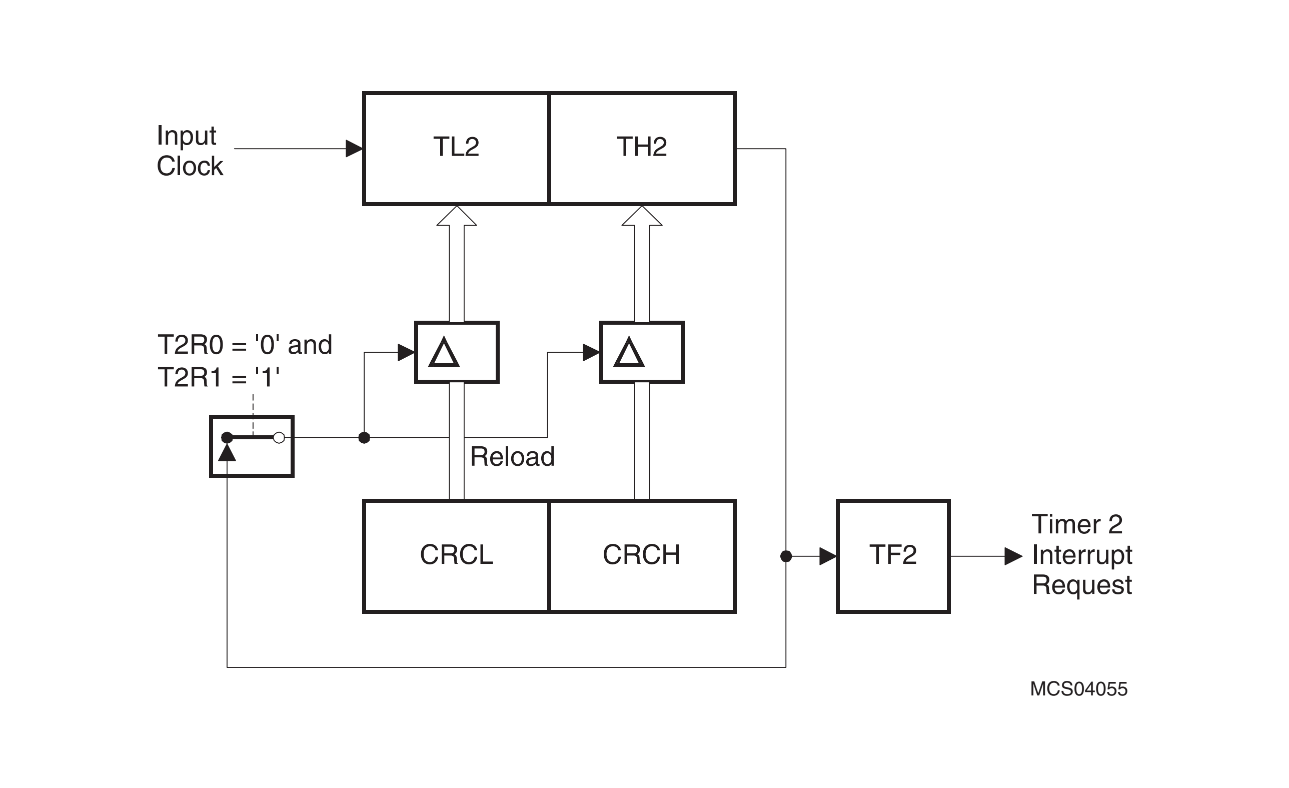


Figure 6-14 Timer 2 in Reload Mode

6.2.2.3 Compare Function of Registers CRC, T2CC1 to T2CC3

The compare function of a timer/register combination can be described as follows. The 16-bit value stored in a Compare/Capture register is compared with 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 a “time stamp” at which a dedicated output reacts in a predefined way (with either a positive or negative transition). Variation of 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 pulse width modulation as well as for a continually controlled generation of any kind of square waveforms. Two Compare modes are implemented to cover a wide range of possible applications.

The compare modes 0 and 1 are selected by bit T2CM in special function register 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, T2CC1 to T2CC3 are multifunctional as they additonally provide a capture, compare or reload capability (reload capability for CRC register only). A general selection of the function is done in register CCEN. Please note that the compare interrupt CC0 can be programmed to be negative or positive transition activated. The internal compare signal (not the output signal at the port pin!) is active as long as the timer 2 contents is equal to the one of the appropriate compare registers, and it has a rising and a falling edge. Thus, when using the CRC register, it can be selected whether an interrupt should be 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 6-16**).

6.2.2.3.1 Compare Mode 0

In mode 0, when the timer and compare register contents match, 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 6-15** shows a functional diagram of a port latch in compare mode 0. The port latch is directly controlled by the two signals timer overflow and compare. 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 initially defined period and duty cycle. This is the mode which needs the least CPU time. Once set up, the output goes on oscillating without any

CPU intervention. Figure 6-16 and Figure 6-17 illustrate the function of compare mode 0.

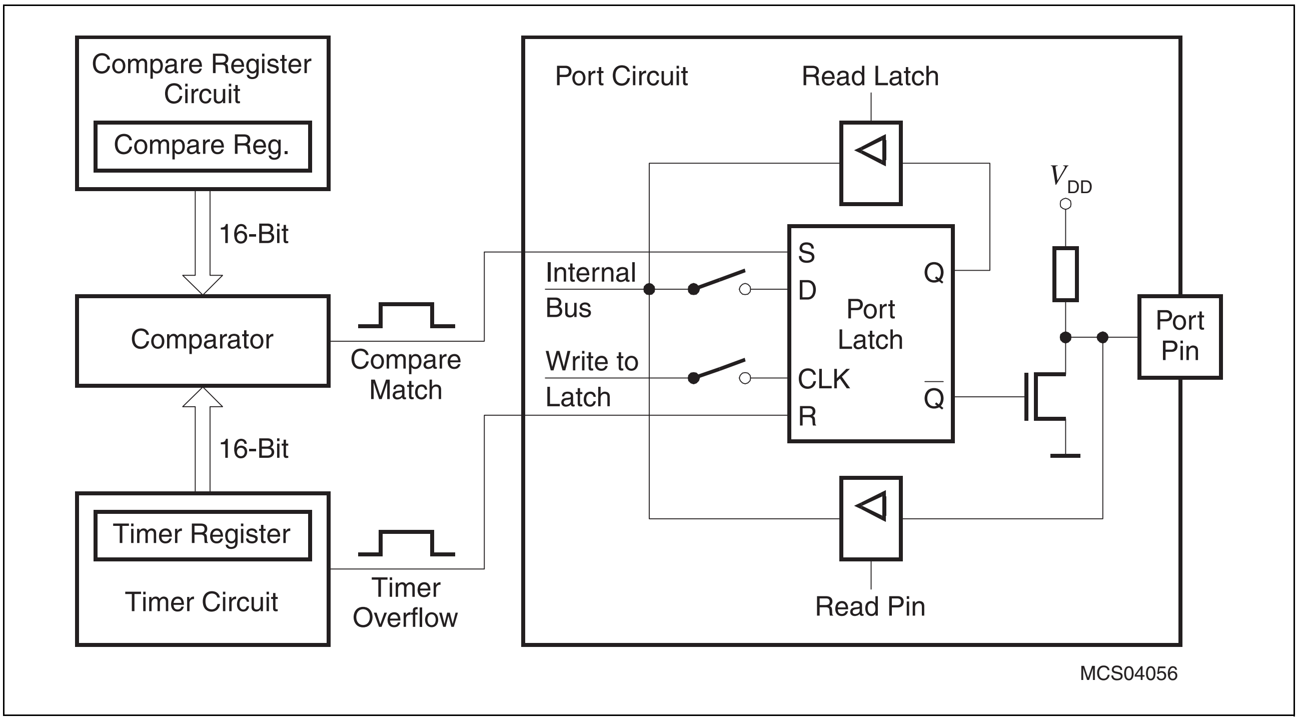


Figure 6-15 Port Latch in Compare Mode 0

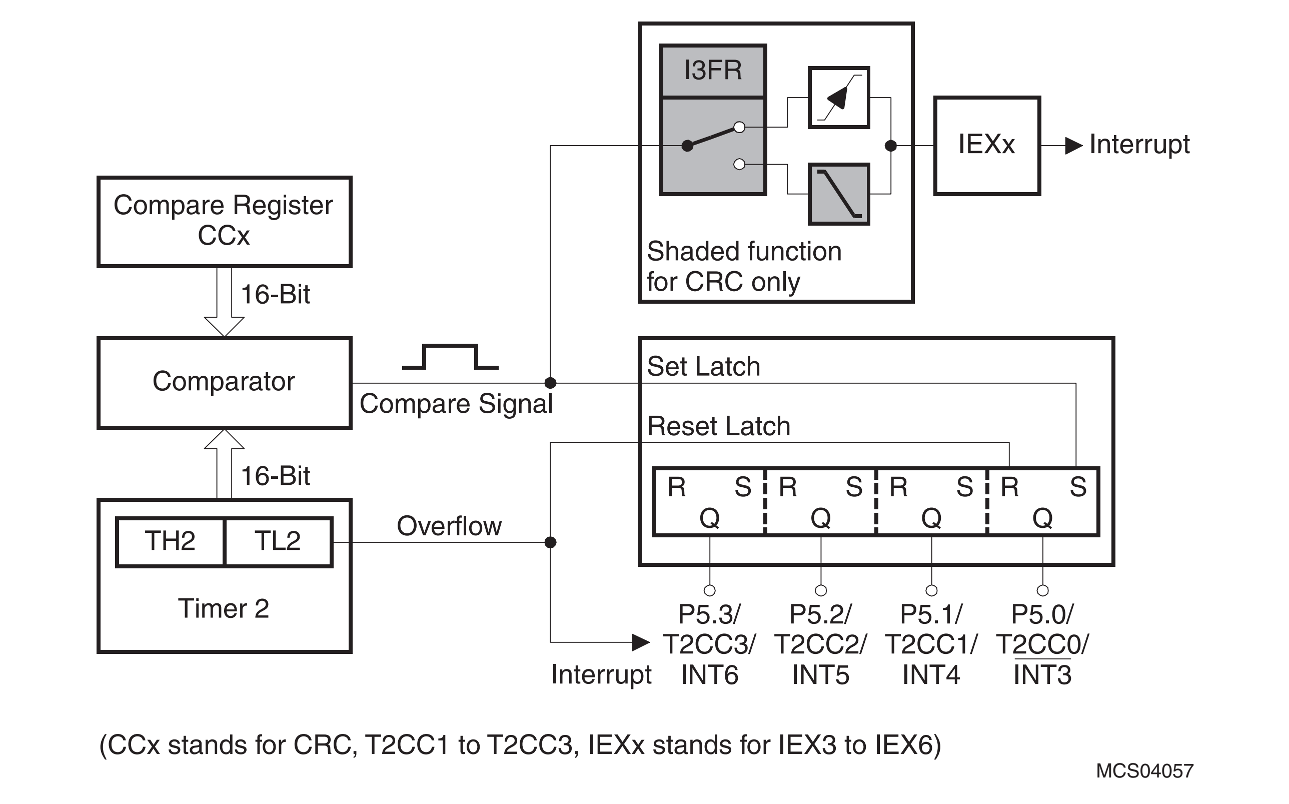


Figure 6-16 Timer 2 with Registers CCx in Compare Mode 0

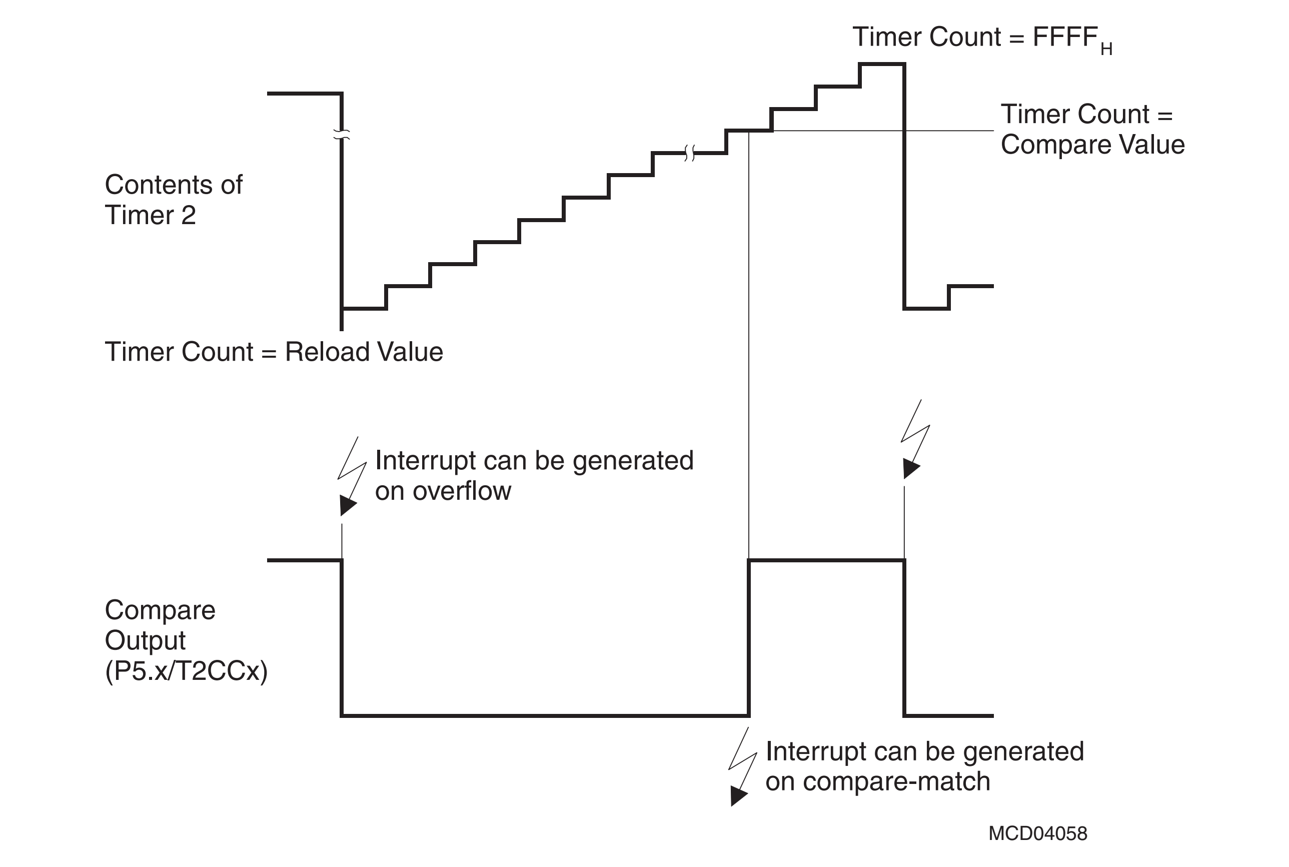

Timer Count = FFFF$_H$
Timer Count = Compare Value
Contents of Timer 2
Timer Count = Reload Value
Interrupt can be generated on overflow
Compare Output (P5.x/T2CCx)
Interrupt can be generated on compare-match
MCD04058

Figure 6-17 Function of Compare Mode 0

6.2.2.3.2 Modulation Range of a PWM Signal 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 which is as long as the timer clock period.

This “spike” may either appear 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. This spike in CCx register configuration of timer 2 in compare mode 0 is divided into two halves. One half is at the beginning when the contents of the compare register are equal to the reload value of the timer and the other half is when the compare register is equal to the maximum value of the timer register (that is, FFFF$_H$). Refer to **Figure 6-18** where the maximum and minimum duty cycles of a compare output signal are illustrated.

Timer 2 is incremented with every machine clock ($f_{OSC}/6$); thus, both these spikes are approximately 150 ns long at 20-MHz operational frequency.

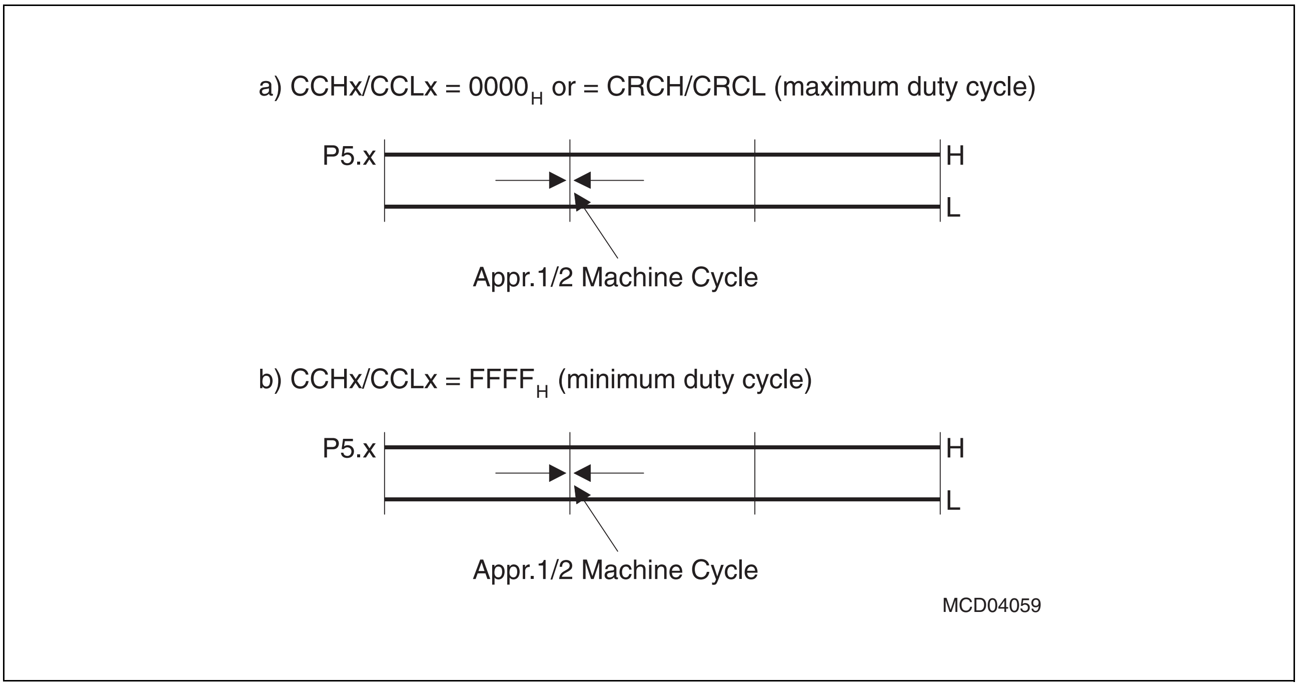

a) CCHx/CCLx = 0000_H or = CRCH/CRCL (maximum duty cycle)
P5.x
H
L
Appr.1/2 Machine Cycle
b) CCHx/CCLx = $FFFF_H$ (minimum duty cycle)
P5.x
H
L
Appr.1/2 Machine Cycle
MCD04059

Figure 6-18 PWM Signal Modulation Range (generated with a Timer 2/CCx Register Combination in Compare Mode 0*)

The following illustrates the calculation of 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-bit) the modulation range would be so severely limited that it would be negligible.

Example:

Case: Timer 2 in auto-reload mode.

Contents of reload register CRC = $FF00_H$

Restriction of modulation range = 1 / (256 × 2) × 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 the 16 bits are used.

6.2.2.3.3 Compare Mode 1

In compare mode 1, the software adoptively 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 which 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 pin level) or should keep its old value at the time the Timer 2 count matches the stored compare value.

Figure 6-19 and **Figure 6-20** show 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 only be transferred to the output latch (and thus to the port pin) 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 in which the CPU could unintentionally 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.

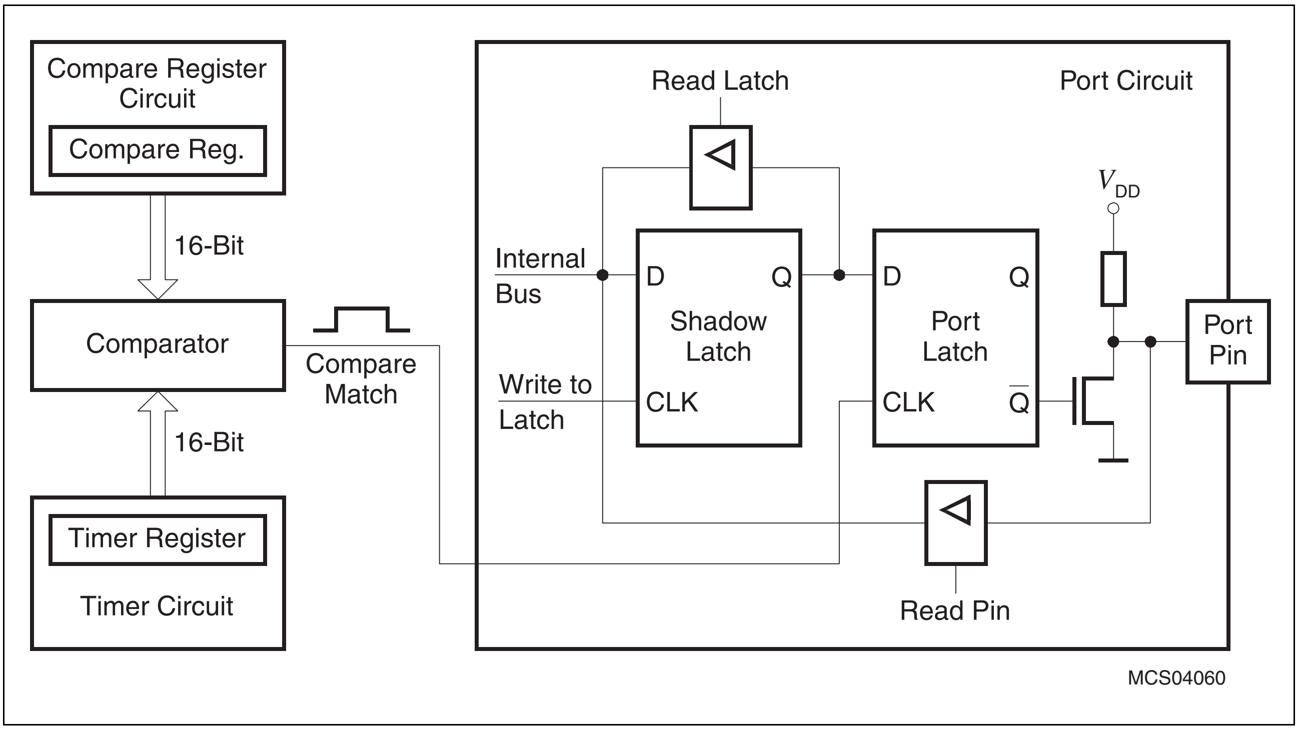
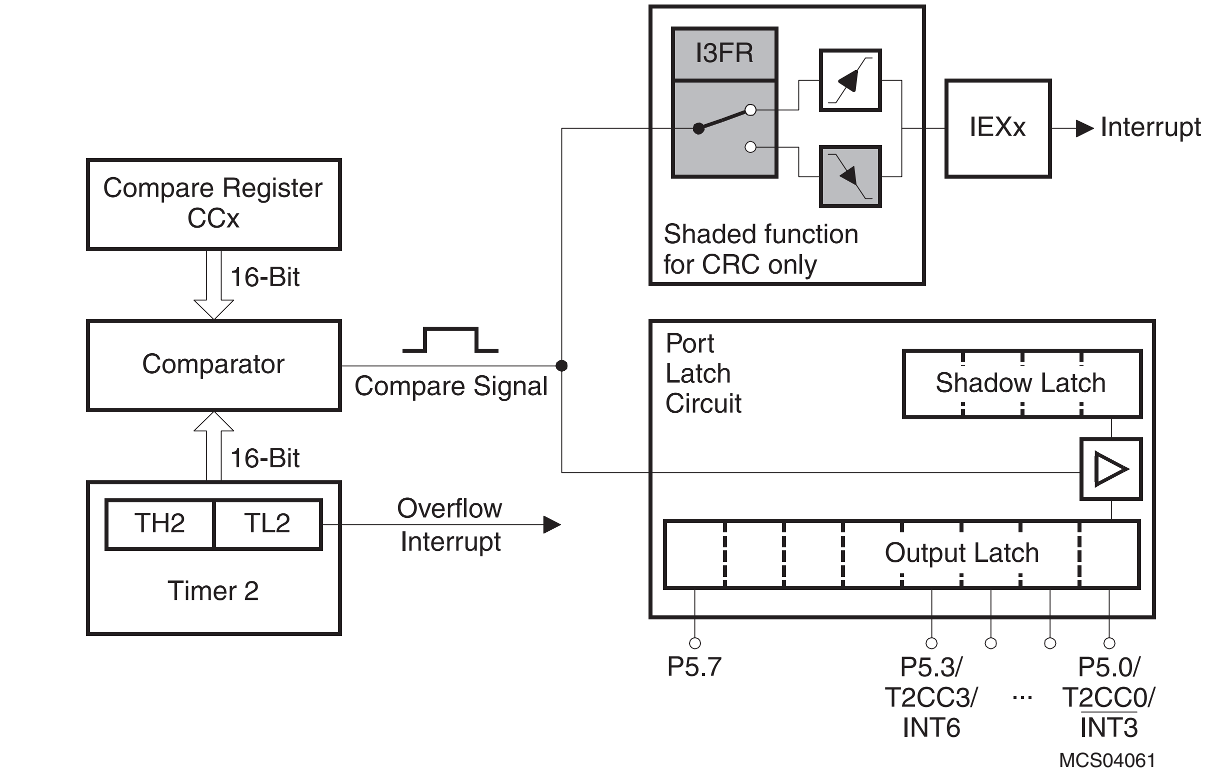


Figure 6-19 Port Latch in Compare Mode 1



(CCx stands for CRC, T2CC1 to T2CC3, IEXx stands for IEX3 to IEX6)

Figure 6-20 Timer 2 with Registers CCx in Compare Mode 1

6.2.2.4 Using Interrupts in Combination with the Compare Function

The compare service of registers CRC, T2CC1, T2CC2 and T2CC3 are assigned to alternate output functions at port pins P5.0 to P5.3. Another option of these pins is that they can be used as external interrupt inputs. However, when using the port lines as compare outputs then 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 a setting of the corresponding interrupt flag. 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 at a match between timer count and register contents) not only manipulates the compare output but also sets the corresponding interrupt request flag. Thus, the current task of the CPU is interrupted if the priority of the Compare interrupt is higher than the present task 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 when Using Compare Interrupts

First, there is no danger of unintentional overwriting a Compare register before a match has been reached. This could happen when the CPU writes to the compare register without knowing about the actual Timer 2 count.

Second, and the most interesting advantage of the compare feature, is that the output pin is exclusively controlled by hardware; therefore, it is completely independent from any service delay which in real time applications could be disastrous. 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.

Please note the following special case where a program using compare interrupts could show a "surprising" behavior.

The configuration has already been mentioned in the description of 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 latches" used in compare mode 1 are transparent while the compare signal is active.

Thus, with a slow input clock for Timer 2, the comparator signal is active for a long time (i.e. high number of machine cycles) and therefore a fast interrupt controlled reload of the compare register could not only change the "shadow latch" - as probably intended - but also the output buffer.

When using the CRC, 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 negative transition triggered is advisable in the above case. Then the compare signal is already inactive and any write access to the port latch changes only the contents of the “shadow-latch”.

Note that for T2CC1 to T2CC3 registers, an interrupt is always requested when the compare signal goes active.

The second configuration which should be noted is the compare function combined with negative transition activated interrupts. If the port latch of Port P5.0 contains a ‘1’, the interrupt request flags IEX3 will immediately be set 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. So, 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 T2CC1 to T2CC3 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 allows the software to read the contents of Timer 2 “on-the-fly”.

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

- For T2CC registers 1 to 3: a positive transition at pins T2CC1 to T2CC3 of Port 5
- 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 P5.0 / T2CC0 / $\overline{\text{INT3}}$.

In both cases, the appropriate Port 5 pin is used as input and the port latch must be programmed to contain a one (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 are 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 T2CC1 to T2CC3 will also set the corresponding external interrupt request flags IEX4 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 (example, 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 6-21 illustrates the operation of the CRC register, while **Figure 6-22** shows the operation of the Compare/Capture registers 1 to 3.

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

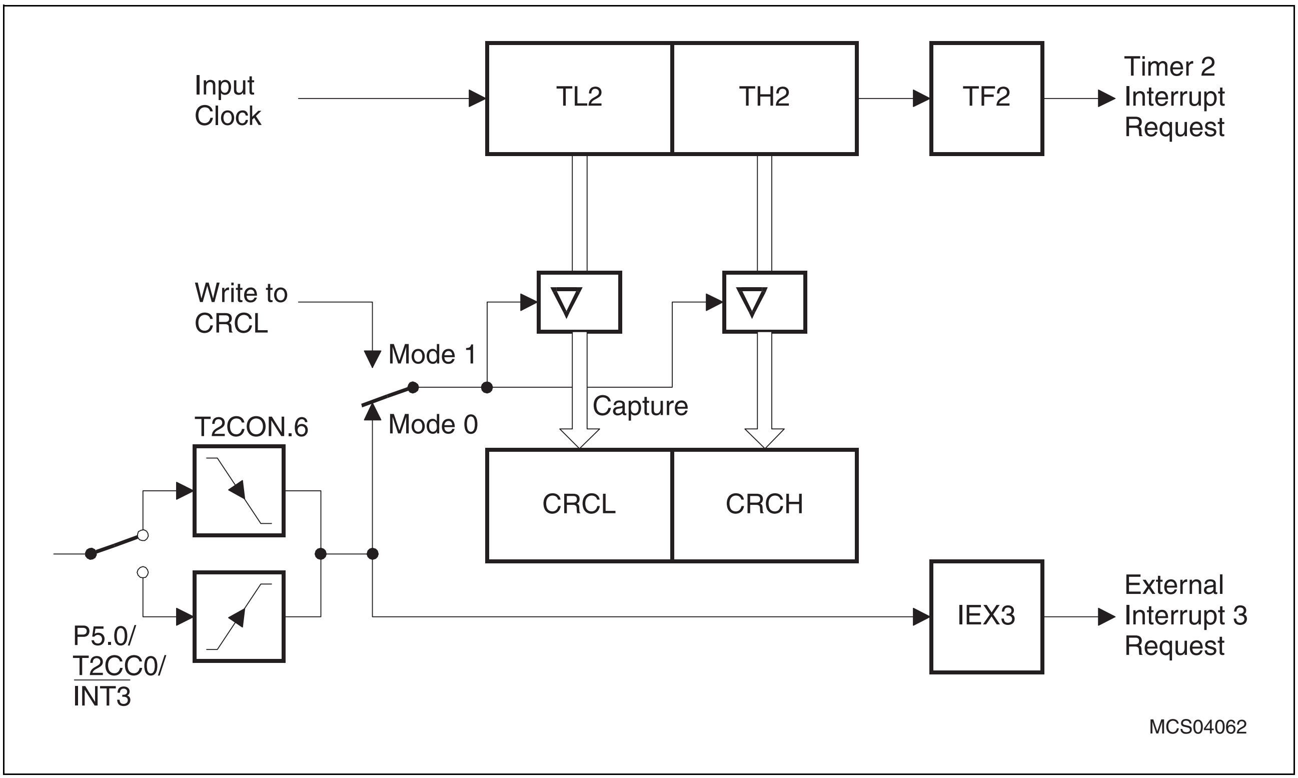


Figure 6-21 Timer 2 – Capture with Register CRC

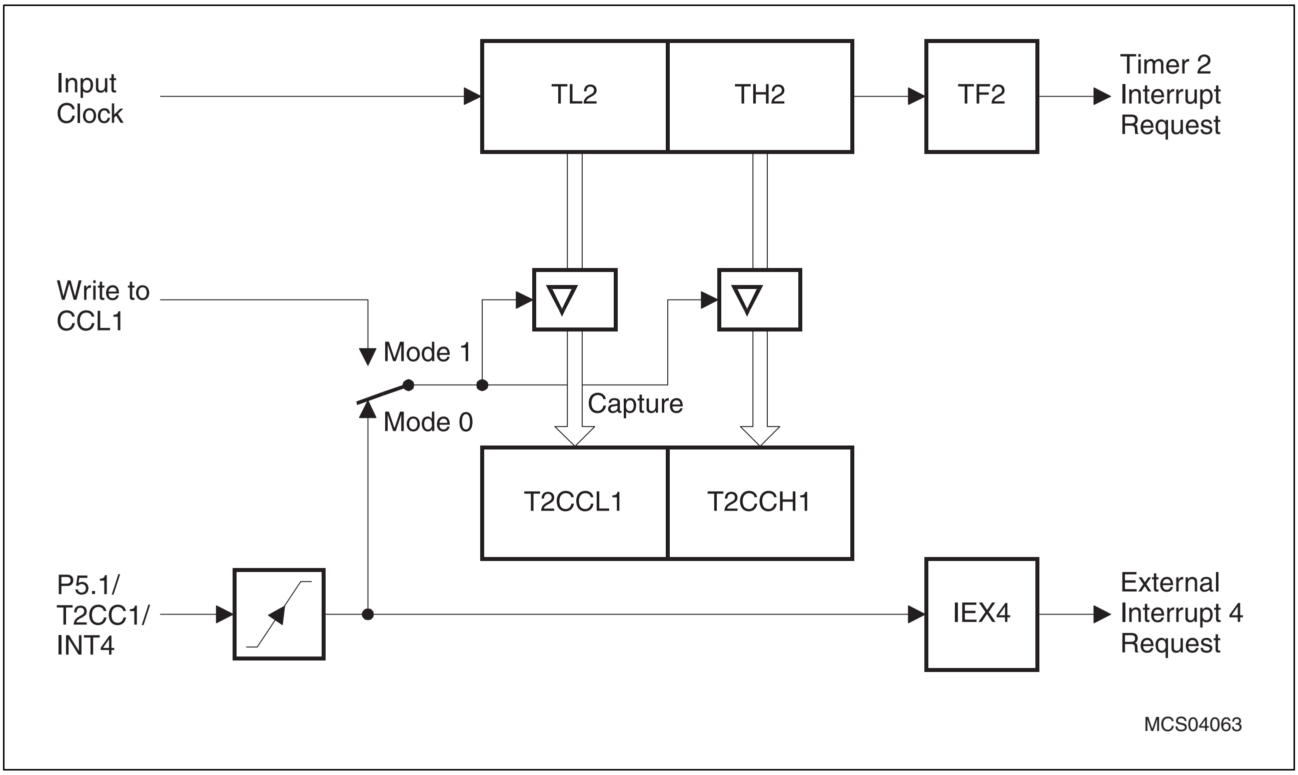


Figure 6-22 Timer 2 – Capture with Registers T2CC1 to T2CC3

6.3 Capture/Compare Unit (CCU)

The Capture/Compare Unit (CCU) of the C508 has been designed for applications which demand digital signal generation and/or event capturing (such as pulse width modulation or pulse width measuring). It consists of a 16-bit three-channel Capture/Compare unit (CAPCOM) and a 10-bit one-channel Compare unit (COMP).

In compare mode, the CAPCOM unit provides two output signals per channel, which can have inverted signal polarity and non-overlapping pulse transitions. The COMP unit can generate a single PWM output signal and is further used to modulate the CAPCOM output signals. For motor control applications, both units (CAPCOM and COMP) may generate versatile multichannel PWM signals. For brushless DC motors, dedicated control modes are supported which are controllable by either software or hardware (hall sensors).

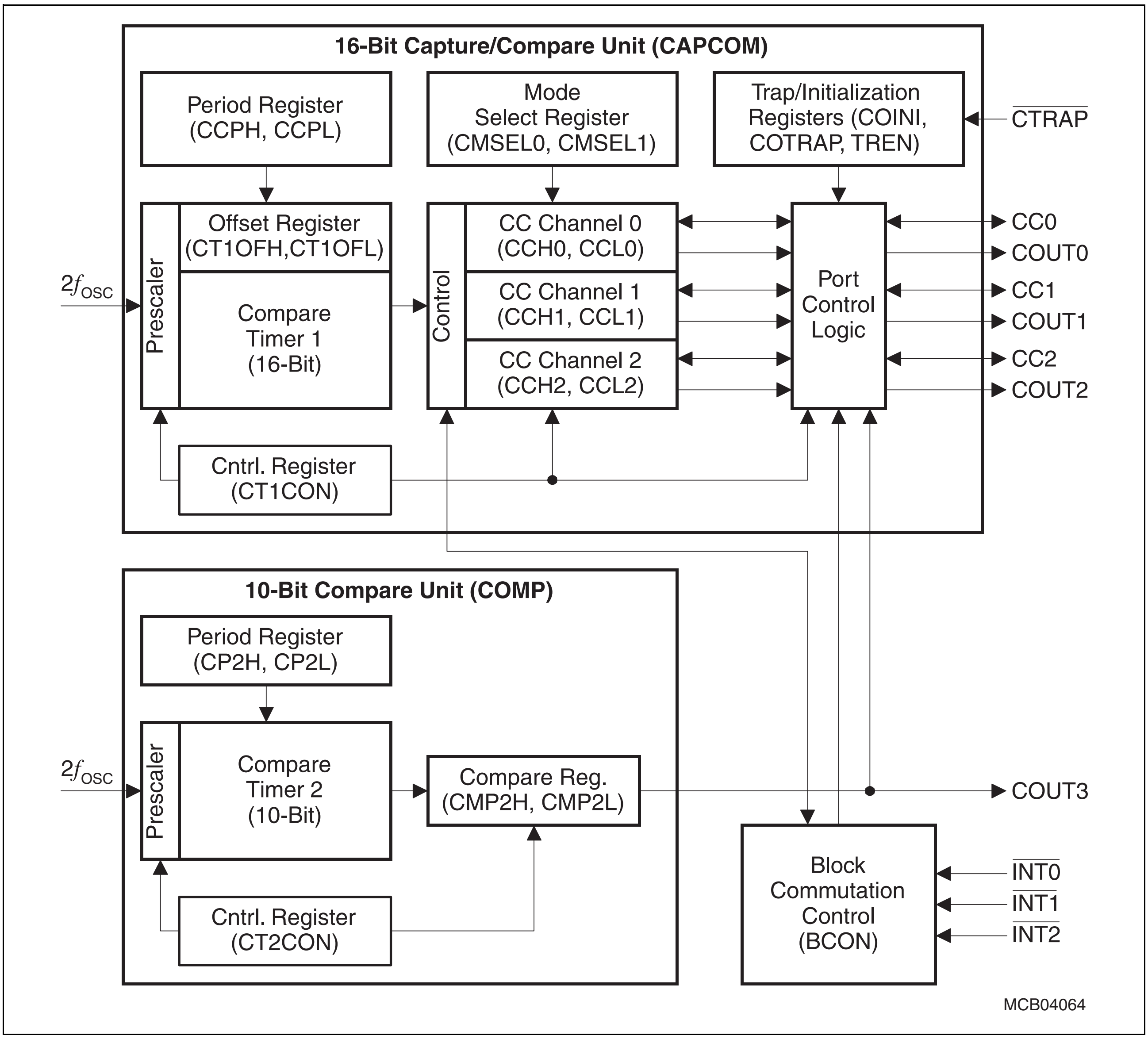

16-Bit Capture/Compare Unit (CAPCOM)
Period Register (CCPH, CCPL)
Mode Select Register (CMSEL0, CMSEL1)
Trap/Initialization Registers (COINI, COTRAP, TREN)
CTRAP
Prescaler
Offset Register (CT1OFH,CT1OFL)
Compare Timer 1 (16-Bit)
$2f_{OSC}$
Control
CC Channel 0 (CCH0, CCL0)
CC Channel 1 (CCH1, CCL1)
CC Channel 2 (CCH2, CCL2)
Port Control Logic
CC0
COUT0
CC1
COUT1
CC2
COUT2
Cntrl. Register (CT1CON)
10-Bit Compare Unit (COMP)
Period Register (CP2H, CP2L)
$2f_{OSC}$
Prescaler
Compare Timer 2 (10-Bit)
Compare Reg. (CMP2H, CMP2L)
COUT3
Cntrl. Register (CT2CON)
Block Commutation Control (BCON)
INT0
INT1
INT2
MCB04064

Figure 6-23 Capture/Compare Unit Block Diagram

6.3.1 General Capture/Compare Unit Operation

The Compare Timer 1 and 2 are free running, processor clock coupled 16-bit/10-bit timers; each of which has a count rate with a maximum of 2 f_{OSC} up to f_{OSC}/64. The compare timer operations with its possible compare output signal waveforms are shown in **Figure 6-24**.

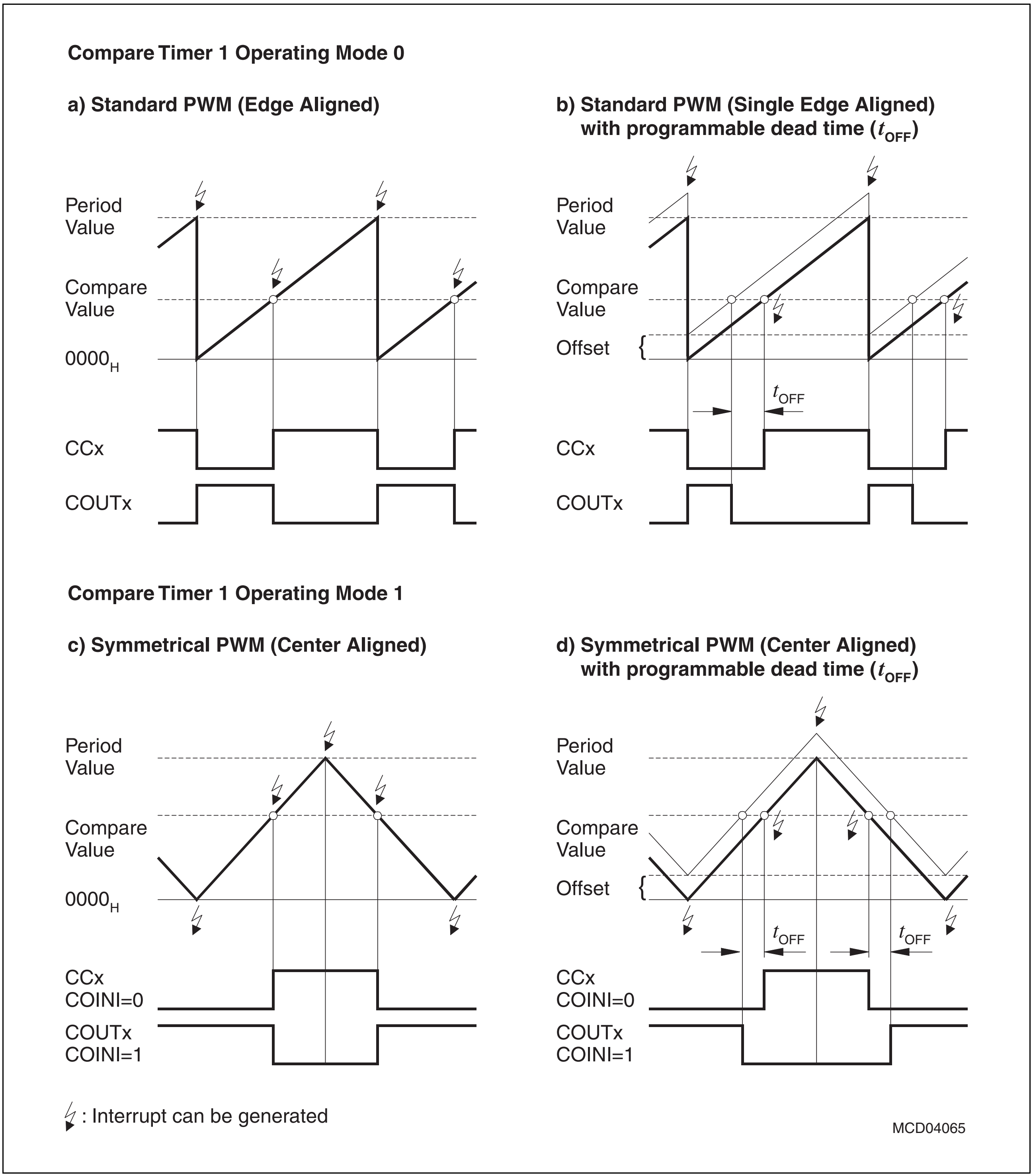

Compare Timer 1 Operating Mode 0
a) Standard PWM (Edge Aligned)
b) Standard PWM (Single Edge Aligned) with programmable dead time (t_{OFF})
Period Value
Compare Value
0000H
Offset
t_{OFF}
CCx
COUTx
Compare Timer 1 Operating Mode 1
c) Symmetrical PWM (Center Aligned)
d) Symmetrical PWM (Center Aligned) with programmable dead time (t_{OFF})
CCx COINI=0
COUTx COINI=1
: Interrupt can be generated
MCD04065

Figure 6-24 CAPCOM Unit Basic Operating Modes

Both compare timers start counting from 0000_H upwards to a count value stored in the period registers. If the value stored in the period register is reached, they are reset (operating mode 0, both compare timers) or the count direction is changed from up-counting to down-counting (operating mode 1, only Compare Timer 1) Using operating mode 0, edge aligned PWM signals can be generated. Using operating mode 1, center aligned PWM signals can be generated. Compare Timer 1 can be programmed for both operating modes while Compare Timer 2 always works in operating mode 0 with one output signal COUT3. **Figure 6-24 a)** and **c)** show the function of these basic operating modes.

Compare Timer 1 has an additional 16-bit offset register, which consists of the high byte stored in CT1OFH and the low byte stored in CT1OFL. If the value stored in CT1OFF is 0, the compare timer operates as shown in **Figure 6-24 a)** and **c)**. If the value stored in CT1OFF is not zero, the compare timer operates as shown in **Figure 6-24 b)** and **d)**. In operating mode 0, Compare Timer 1 is always reset after its value has been equal to the value stored in period register. In operating mode 1, the count direction of the compare timer is changed from up- to down-counting when its value has reached the value stored in the period register. The count direction is changed from down- to up-counting when the compare timer value has reached 0000_H. Generally, the compare outputs CCx are always assigned to a match condition with the compare timer value directly, where as the compare outputs COUTx are assigned to a match condition with the compare timer value plus the offset value. Therefore, signal waveforms with non-overlapping signal transitions as shown in **Figure 6-24 b)** and **d)** can be generated.

Further, the initial logic output level of the CAPCOM channel outputs can be selected in compare mode. This allows waveforms to be generated with inverting signal polarities.

In capture mode of the CAPCOM unit, the value of Compare Timer 1 is stored in the capture registers on a signal transition at pins CCx.

The compare unit COMP is a 10-bit compare unit which can be used to generate a Pulse Width Modulated signal. This PWM output signal drives the output pin COUT3. In burst mode and in the PWM modes, the output of the COMP unit can be switched to the COUTx outputs.

The block commutation control logic allows to generate versatile multi-channel PWM output signals. In one of these modes, the block commutation mode, signal transitions at the three external interrupt inputs are used to trigger the PWM signal generation logic. Depending on these signal transitions, the six I/O lines of the CAPCOM unit, which are decoupled in block commutation mode from the three Capture/Compare channels, are driven as static or PWM modulated outputs. CAPCOM channel 0 can be used in block commutation mode for a capture operation (speed measurement) which is triggered by each transition at the external interrupt inputs.
Further, the multi-channel PWM mode signal generation can be also triggered by the period of Compare Timer 1. These operating modes are referenced as multi-channel PWM modes.

Using the $\overline{\text{CTRAP}}$ input signal of the C508, the compare outputs can be put immediately into their state as defined in COTRAP register.

The CCU unit has four main interrupt sources with their specific interrupt vectors. Interrupts can be generated at the Compare Timer 1 period match or count-change events, at the Compare Timer 2 period match event, at a CAPCOM Compare match or Capture event, and at a CAPCOM emergency event. An emergency event occurs if an active $\overline{\text{CTRAP}}$ signal is detected or if an error condition in block commutation mode is detected. All interrupt sources can be enabled/disabled individually.

6.3.2 CAPCOM Unit Operation

6.3.2.1 CAPCOM Unit Clocking Scheme

The CAPCOM unit is controlled by the 16-bit Compare Timer 1. Compare Timer 1 is the timing base for all compare and capture capabilities of the CAPCOM unit. The input clock for Compare Timer 1 is directly coupled to the system clock of the C508. Its frequency can be selected via three bits of the CT1CON register in a range of 2 f_{OSC} up to $f_{OSC}/64$.

For the understanding of the following timing diagrams, **Figure 6-25** shows the internal clocking scheme of the CAPCOM unit. The internal input clock of the CAPCOM unit is a symmetrical clock with 50% duty cycle. The clock transitions (edges) of the CAPCOM internal input clock are used for different actions. At clock edge 1, the Compare Timer 1 is clocked to the next count value and with clock edge 2, the compare outputs CCx and COUTx are toggled/set to the new logic level if required.

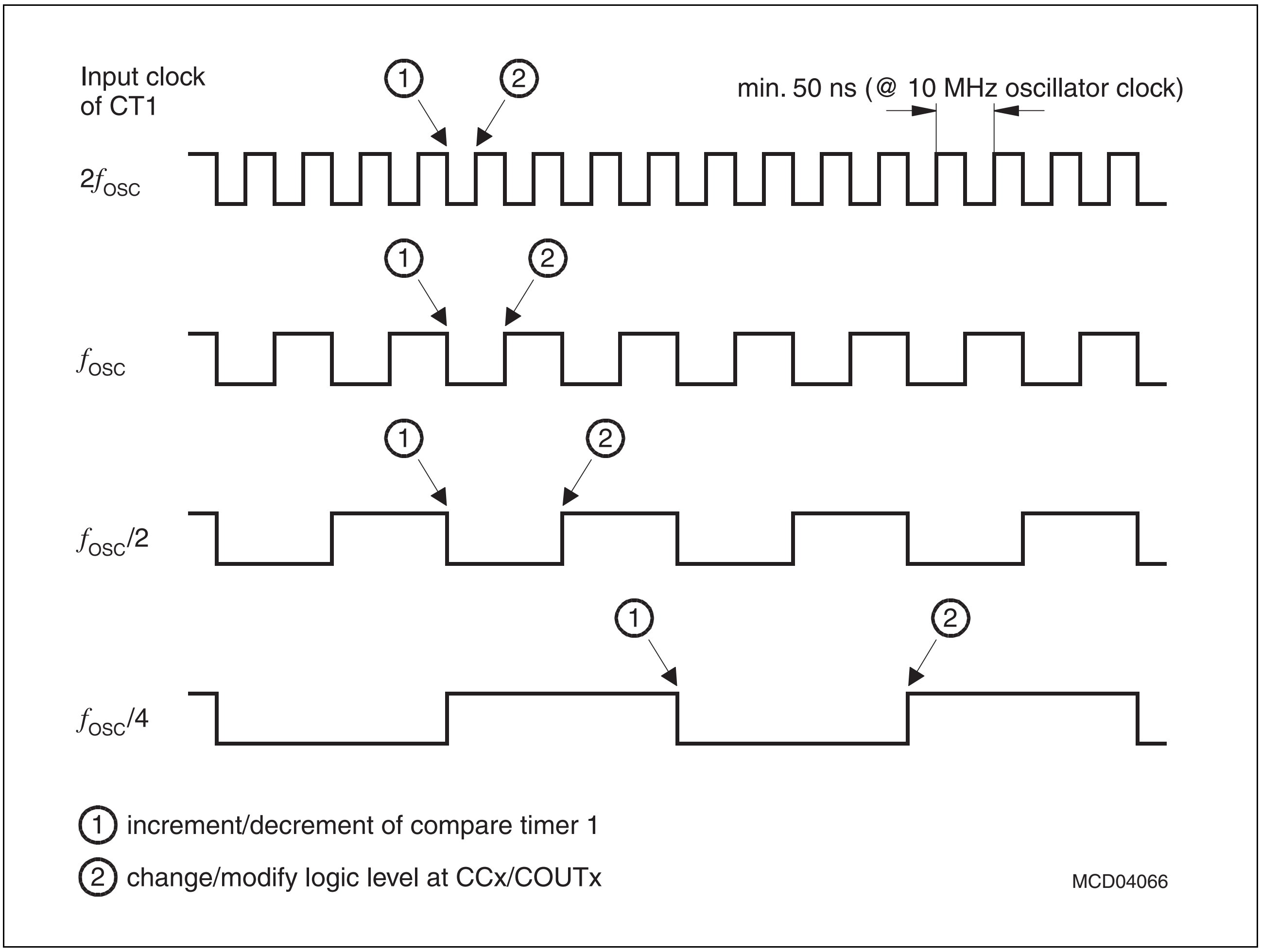
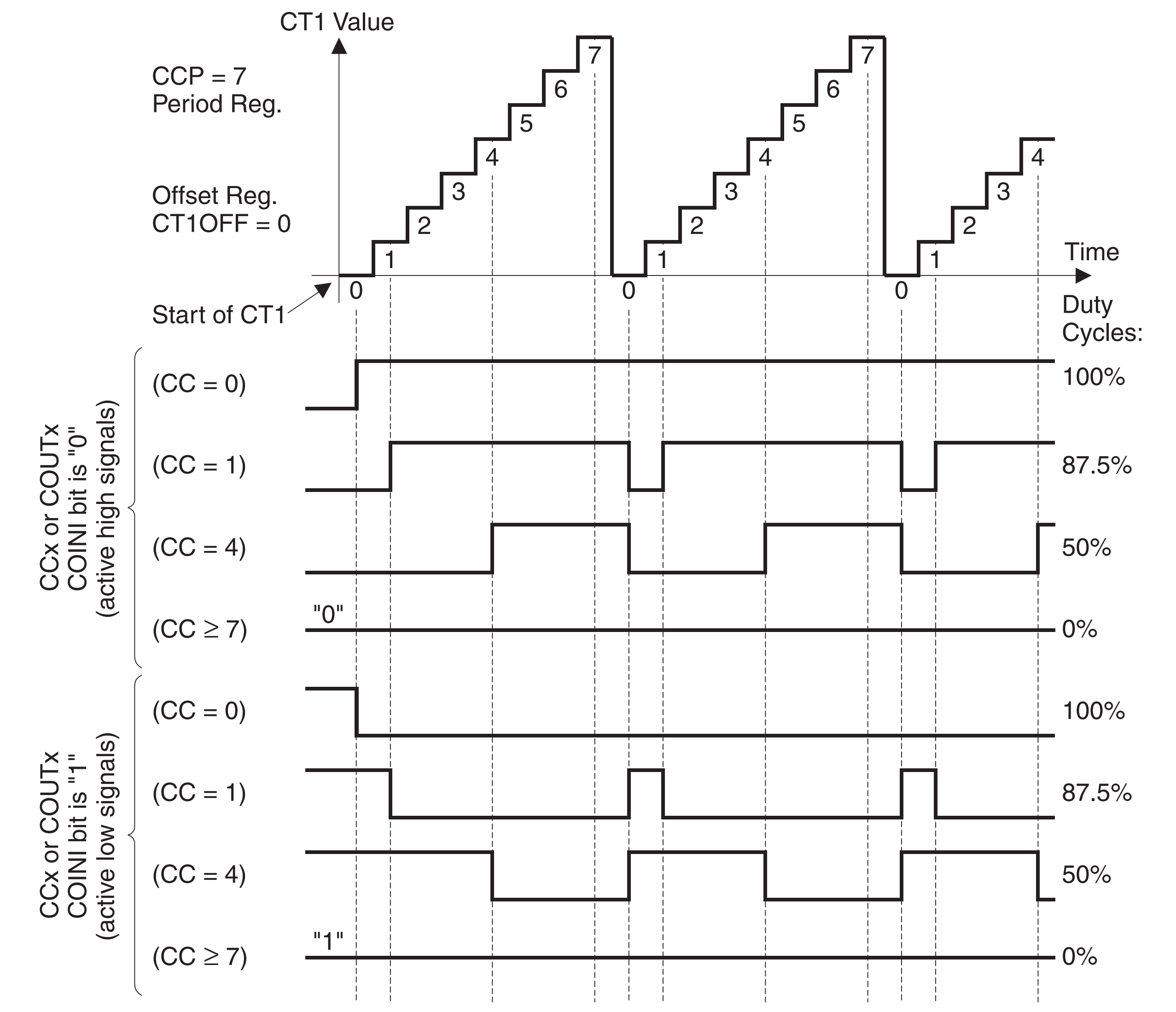

Input clock
of CT1
min. 50 ns (@ 10 MHz oscillator clock)
2fOSC
fOSC
fOSC/2
fOSC/4
1
2
1 increment/decrement of compare timer 1
2 change/modify logic level at CCx/COUTx
MCD04066

Figure 6-25 CAPCOM Unit Clocking Scheme

Generally, the CAPCOM clocking scheme shown above is also valid for the COMP (Compare Timer 2) unit.

6.3.2.2 CAPCOM Unit Operating Mode 0

Figure 6-26 shows the details of the CAPCOM unit timing in operating mode 0.



CC : content of the CCxH/CCxL compare registers
CCP : content of the CCPH/CCPL period register
CC1O : content of the CT1OFH/CT1OFL offset register

MCT04067

Figure 6-26 Compare Timer 1 Mode 0

In the example above, Compare Timer 1 counts from 0000_H up to 0007_H (value stored in CCPH/CCPL). The offset registers CT1OFH/CT1OFL have a value of 0000_H. If programmed in compare mode, two output signals (CCx and COUTx) are assigned to the related CAPCOM channel x. The mode select bits in the SFRs CMSEL0 and

CMSEL1 define which of these two outputs will be controlled by the CAPCOM channel. In **Figure 6-26** only the CCx signal is shown, but the same or the inverted waveform can be generated at the COUTx outputs.

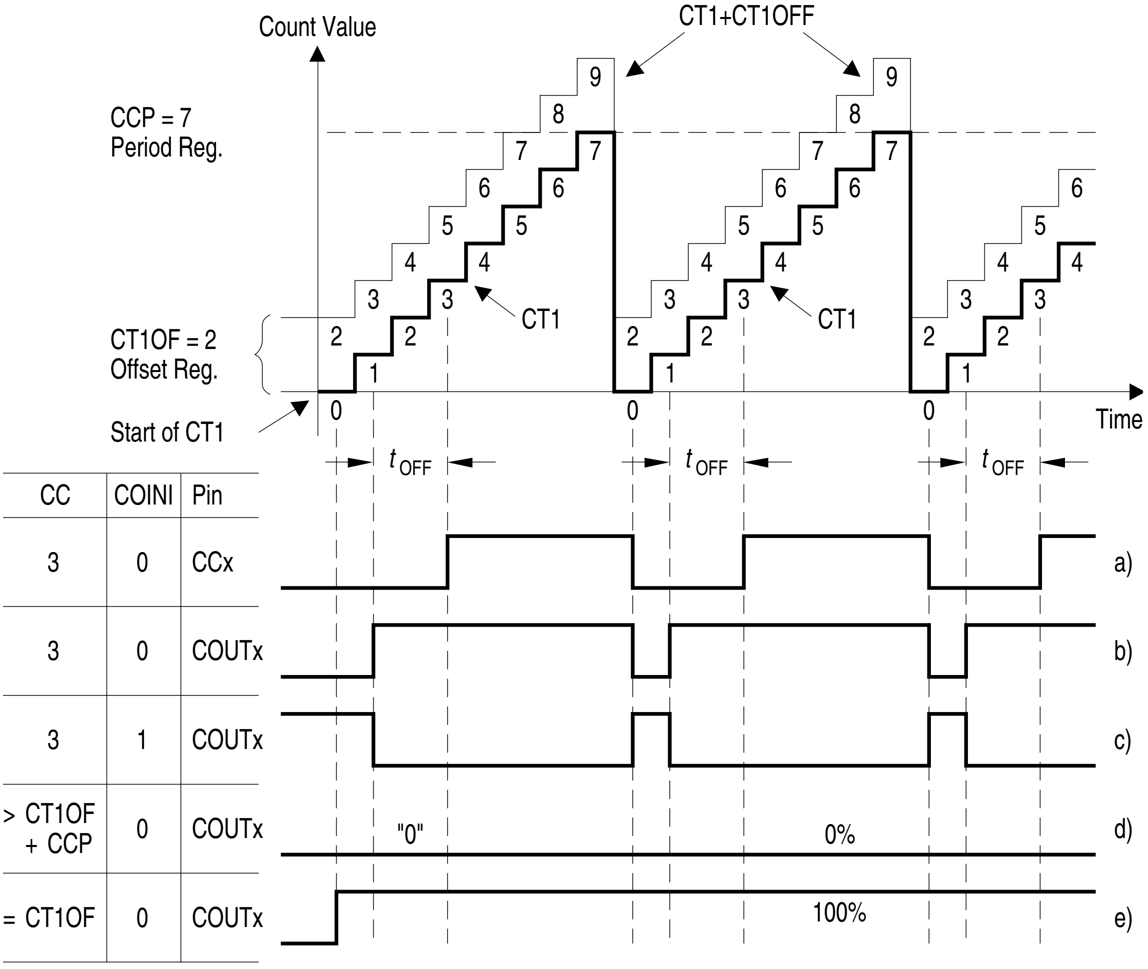
After reset all CCx/COUTx pins are at high level, driven by a weak pull-up. With the programming of the CMSEL1 or CMSEL0 registers, all affected compare outputs are switched to push-pull mode and start driving an initial level which is defined by the bits in SFR COINI. In **Figure 6-26**, the upper five waveforms are assigned to a CCx pin with the appropriate bit in COINI cleared while the lower five waveforms are assigned to a CCx pin with the appropriate bit in COINI set.

When the count value of the Compare Timer 1 is incremented and the new value matches the value stored in the corresponding compare register, the related compare output changes its logic state. When the compare timer is reset to 0000_H the related compare output changes its logic state again. With the scheme shown in **Figure 6-26**, output waveforms with duty cycles between 0% and 100% can be generated. For a compare register value of 0000_H, the output will remain at high level (COINI bit = 0) or low level (COINI bit = 1), representing a duty cycle of 100%. If the value stored in the compare register is greater than or equals to the value of the period register, a low level (COINI bit = 0) or high level (COINI bit = 1) corresponds to a duty cycle of 0%.

Figure 6-27 shows the waveform generation in operating mode 0 when the offset register has a value which is not equal 0000_H (example: CT1OFH/CT1OFL = 0002_H). Using Compare Timer 1 with an offset value not equal 0 is used to generate single edge aligned signals with a constant delay between one of the two signal transitions.

Compare Timer 1 always counts from 0000_H up to the value stored in CCP, if the value in the offset register is not equal 0. With reset (count value 0000_H) of the Compare Timer 1, the CCx and COUTx will always change their logic state. During the up-counting phase, CCx will change the logic state when the compare timer value is equal to the compare register value; and COUTx will change the logic state when the compare timer value plus the offset value matches the value stored in the compare register.

In **Figure 6-27** the waveforms **a)** and **b)** show an example for a waveform of two signals with a constant delay of their rising edge. A compare register value of 3 is assumed. Using inverted signal polarity (SFR COINI), signal **c)** can be generated at COUTx. If the value in the offset register plus the value of the period register is less than or equal to the value stored in the compare register, a static ‘1’ or a static ‘0’ (depending on COINI content) will be generated at COUTx (see **Figure 6-27** **d)** and **e)**). Therefore, CCx will also stay at a static level if the compare register value is greater than the value stored in the period register.



CC	COINI	Pin
3	0	CCx
3	0	COUTx
3	1	COUTx
> CT1OF + CCP	0	COUTx
= CT1OF	0	COUTx

CC : content of the CCxH/CCxL compare registers
CCP : content of the CCPH/CCPL period register
CT1OF : content of the CT1OFH/CT1OFL offset registers

Figure 6-27 Compare Timer 1 with Offset not equal 0 – Mode 0

6.3.2.3 CAPCOM Unit Operating Mode 1

Using Compare Timer 1 in operating mode 1, two symmetric output signals with constant dead time t_{OFF} at each signal transition can be generated per channel. **Figure 6-28** shows the operating mode 1 timing in detail.

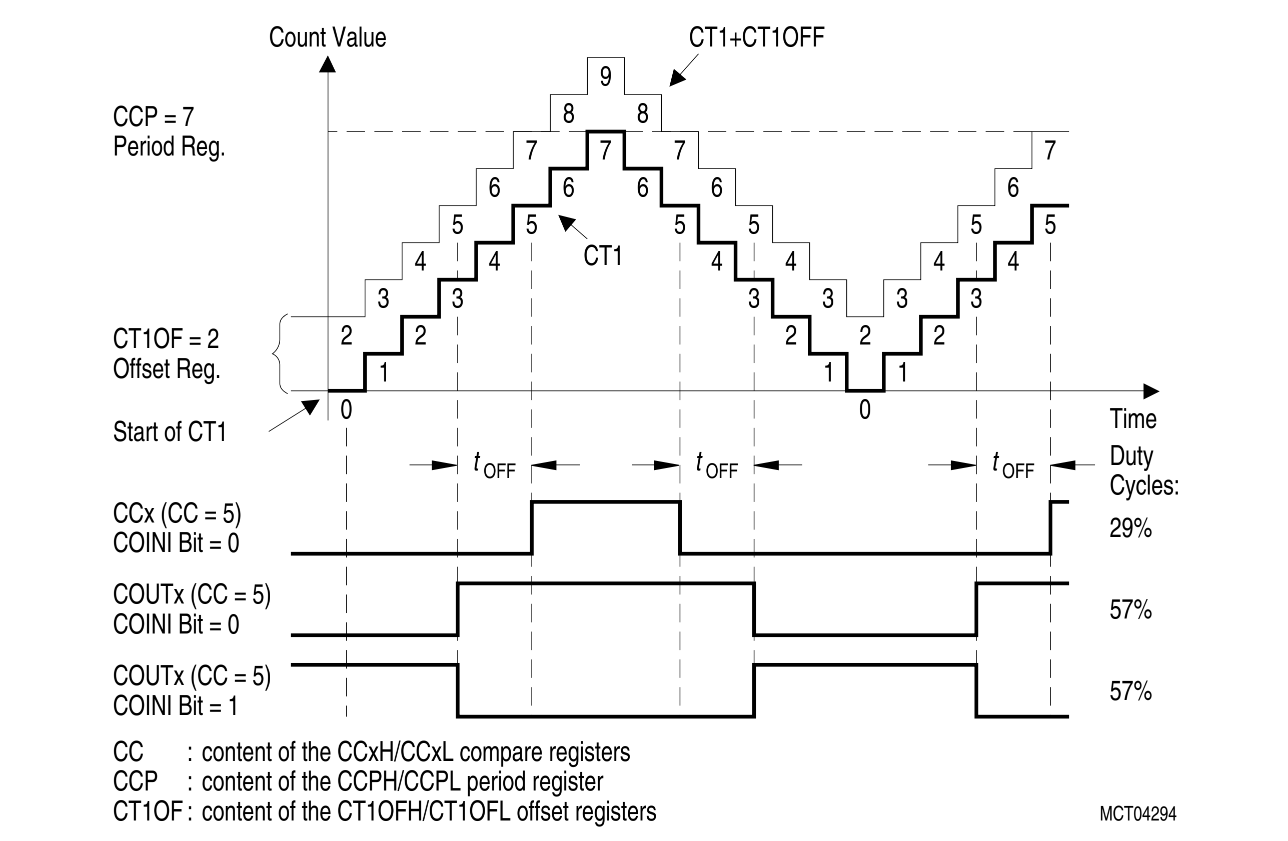


Figure 6-28 Compare Timer 1 with Offset not equal to 0 - Mode 1

In the example above, Compare Timer 1 counts from 0000_H up to 0007_H (value stored in period register CCPH/CCPL) and then counts down again to 0000_H. The maximum and minimum (0000_H) values of the Compare Timer 1 always occur once in the count value sequence. In the example shown in **Figure 6-28**, the offset registers have a value of 0002_H.

With the programming of the CMSEL1 or CMSEL0 registers, all affected compare outputs are switched to push-pull mode and start driving an initial level defined by the bits in SFR COINI.

In operating mode 0, two compare output signals, CCx and COUTx, are assigned to the related CAPCOM channel. The compare outputs CCx change their state if a match of Compare Timer 1 content and the corresponding compare register occurs. The compare outputs COUTx change their state when a match of Compare Timer 1 content plus the value stored in the offset registers and the corresponding compare register has occurred. If the value in the offset register plus the value of the period register is less than or equal to the value stored in the compare register, a static ‘1’ or a static ‘0’ (depending on COINI content) will be generated at COUTx. In the same way, CCx will also stay at a static level if the compare register value is greater than the value stored in the period register.

6.3.2.4 CAPCOM Unit Timing Relationships

Depending on the operating mode of the Compare Timer 1, compare output signals can be generated with a maximum period and resolution as shown in **Figure 6-29.** This example also demonstrates the reloading of the compare and period registers which occurs when Compare Timer 1 reaches the count value 0000_H.

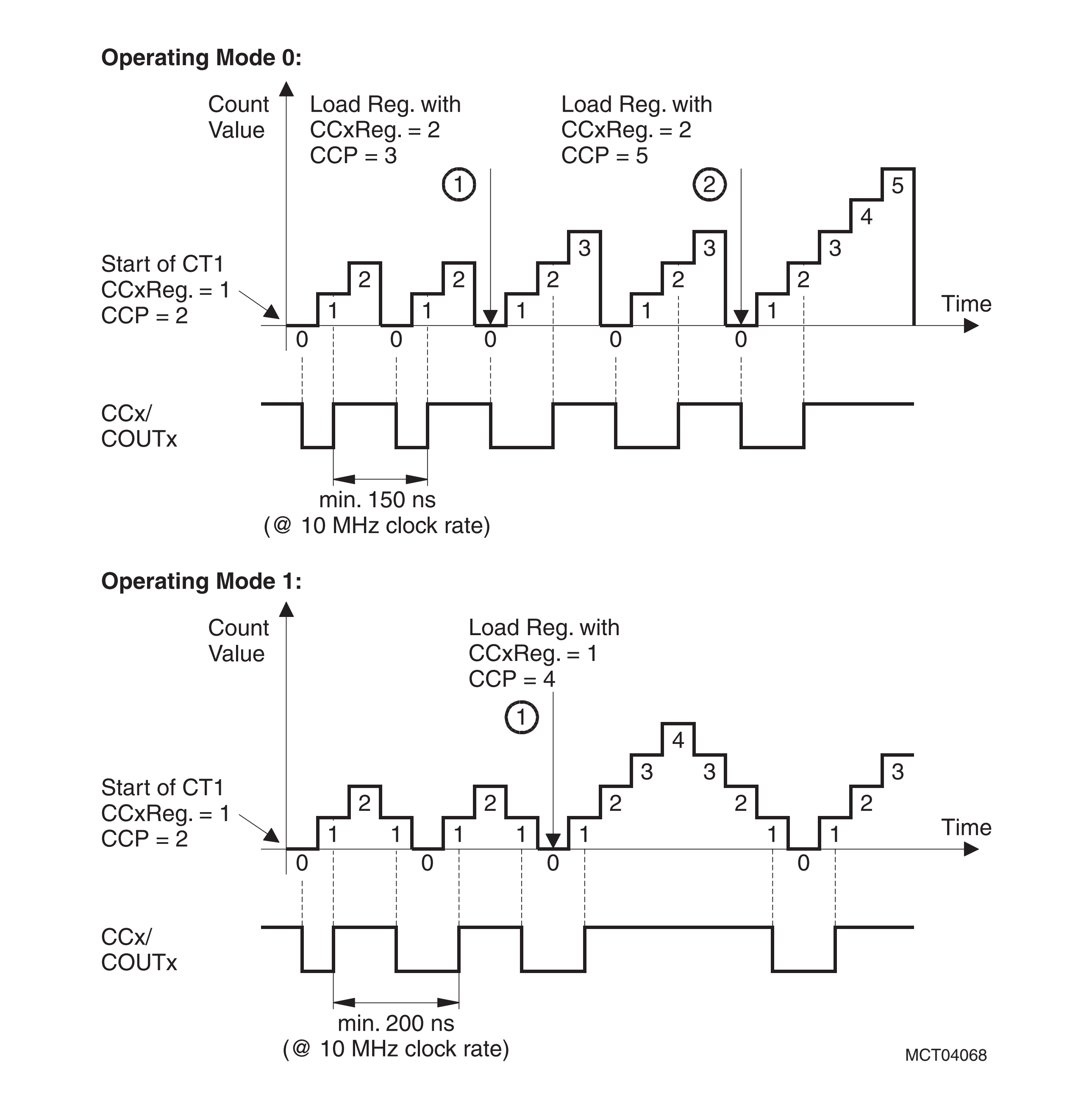

Operating Mode 0:
Count Value
Load Reg. with CCxReg. = 2 CCP = 3
Load Reg. with CCxReg. = 2 CCP = 5
1
2
Start of CT1 CCxReg. = 1 CCP = 2
0 1 2 0 1 2 0 1 2 3 0 1 2 3 0 1 2 3 4 5
Time
CCx/ COUTx
min. 150 ns (@ 10 MHz clock rate)
Operating Mode 1:
Count Value
Load Reg. with CCxReg. = 1 CCP = 4
1
Start of CT1 CCxReg. = 1 CCP = 2
0 1 2 1 0 1 2 1 0 1 2 3 4 3 2 1 0 1 2 3
Time
CCx/ COUTx
min. 200 ns (@ 10 MHz clock rate)
MCT04068

Figure 6-29 Maximum Period and Resolution of the Compare Timer 1 Unit

Figure 6-29 shows the resolution and the period value range which depends on the selected Compare Timer 1 input clock prescaler ratio.

Table 6-6 Resolution and Period of the Compare Timer 1 (at f_{OSC} = 10 MHz)

Compare Timer 1 Input Clock	Operating Mode 0		Operating Mode 1	
	Resolution	Period	Resolution	Period
2 f_{OSC}	50 ns	100ns - 3.28 ms	50 ns	200 ns - 6.55 ms
f_{OSC}	100 ns	200 ns - 6.55 ms	100 ns	400 ns - 13.11 ms
f_{OSC} / 2	200 ns	400 ns - 13.11 ms	200 ns	800 ns - 26.21 ms
f_{OSC} / 4	400 ns	800 ns - 26.21 ms	400 ns	1.6 µs - 52.43 ms
f_{OSC} / 8	800 ns	1.6 µs - 52.43 ms	800 ns	3.2 µs - 104.86 ms
f_{OSC} / 16	1.6 µs	3.2 µs - 104.86 ms	1.6 µs	6.4 µs - 209.71 ms
f_{OSC} / 32	3.2 µs	6.4 µs - 209.72 ms	3.2 µs	12.8 µs - 419.42 ms
f_{OSC} / 64	6.4 µs	12.8 µs - 419.43 ms	6.4 µs	25.6 µs - 838.85 ms

Compare Timer 1 period and duty cycle values can be calculated using the formulas given below. Following abbreviations are used.

pv = period value, stored in the period registers CCPH/CCPL
ov = offset value, stored in the offset registers CT1OFH/CT1OFL
cv = compare value, stored in the Capture/Compare registers CCHx/CCLx

Operating Mode 0:

$$\text{Period value} = \text{pv} + 1$$

$$\text{Duty cycle of CCx outputs} = \left(1 - \frac{\text{cv}}{\text{pv} + 1}\right) \times 100\%$$

$$\text{Duty cycle of COUTx outputs} = \left(1 - \frac{\text{cv - ov}}{\text{pv} + 1}\right) \times 100\%$$

Operating Mode 1:

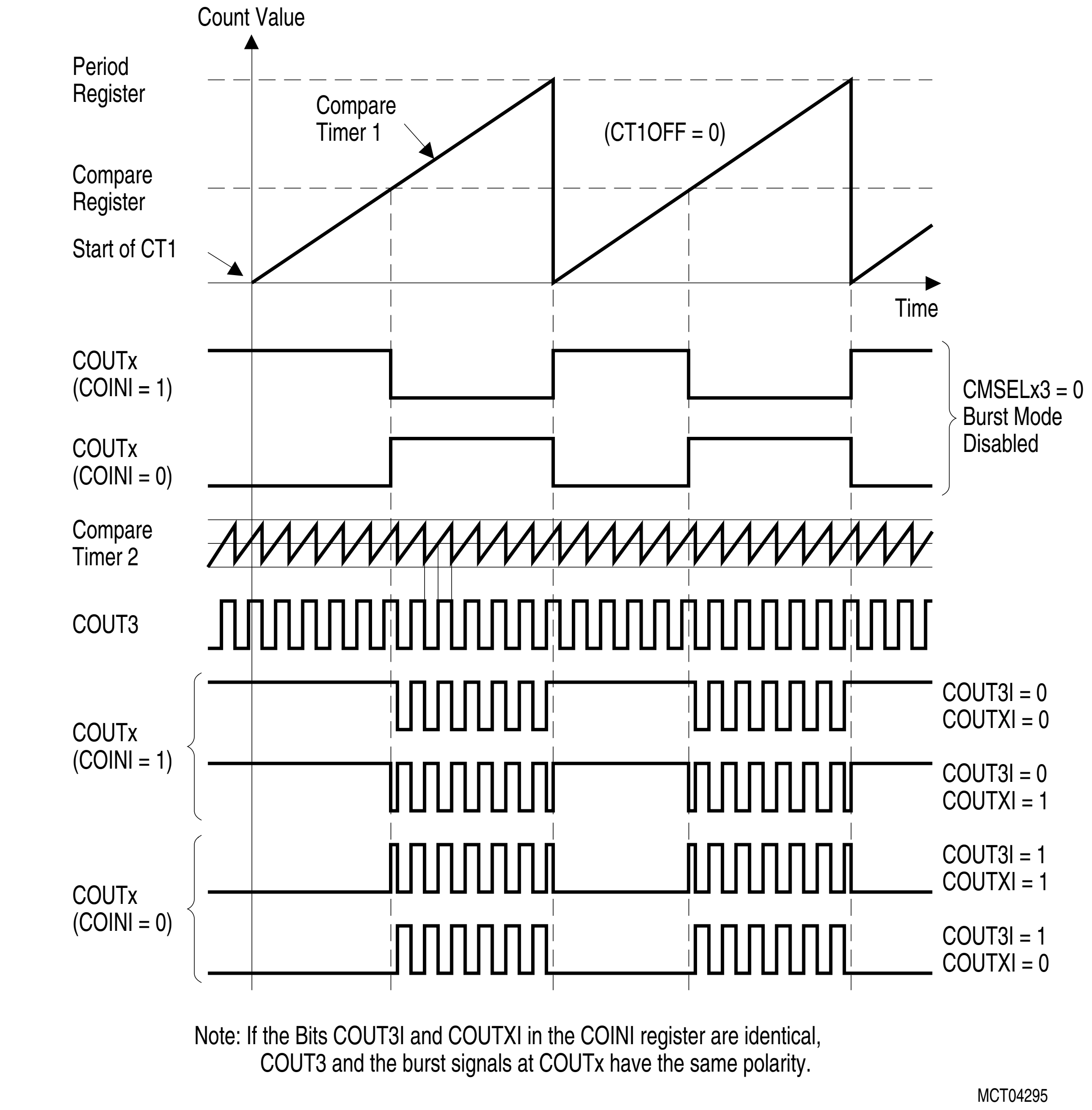
$$\text{Period value} = 2 \times \text{pv}$$

$$\text{Duty cycle of CCx outputs} = \left(1 - \frac{\text{cv}}{\text{pv}}\right) \times 100\%$$

$$\text{Duty cycle of COUTx outputs} = \left(1 - \frac{\text{cv - ov}}{\text{pv}}\right) \times 100\%$$

6.3.2.5 Burst Mode of CAPCOM / COMP Unit

In burst mode, both units of the CCU are combined in a way that the CAPCOM outputs COUTx or CCx **and** COUTx (controlled by bit BCMP in SFR BCON) are modulated by the output signal of the COMP unit. Using the burst mode, the CAPCOM unit operates in compare mode and the COMP unit provides a PWM signal which is switched to the COUTx outputs. This PWM signal typically has a higher frequency than the compare output signal of the CAPCOM unit. **Figure 6-30** shows the waveform generation using the burst mode.



Note: If the Bits COUT3I and COUTXI in the COINI register are identical, COUT3 and the burst signals at COUTx have the same polarity.

Figure 6-30 Burst Mode Operation

Burst mode of a COUTx output is enabled by the bit CMSELx3, located in the mode select registers CMSEL0 and CMSEL1. **Figure 6-30** shows four CAPCOM output signals with different initial logic states with burst mode disabled (CMSELx3 = 0) and burst mode enabled (CMSELx3 = 1). Generally, the CCx outputs cannot operate in burst mode. Optionally, the signal at COUTx may have inverted polarity than the PWM signal which is available at pin COUT3.

Depending on the corresponding initial compare output level bit in COINI, either a low or high level for the non-modulated state at the COUTx pins can be selected. Burst mode can be enabled in both operating modes of the Compare Timer 1. The burst mode as shown in **Figure 6-30** is only valid if the block commutation mode of the CCU is disabled (bit BCEN of SFR BCON cleared).

Modulation of the compare output signals at COUTx is switched on (COUT3 signal is switched to COUTx) when the Compare Timer 1 contents plus the value stored in the Compare Timer 1 offset register are equal to or greater than the value stored in the compare register of CAPCOM channel x.

6.3.2.6 CAPCOM Unit in Capture Mode

The three channels of the CAPCOM unit can be individually programmed to operate in capture mode. In capture mode, each CAPCOM channel offers one capture input at pin CCx. Compare Timer 1 runs either in operating mode 0 or 1. A rising or/and falling edge at CCx will copy the actual value of the Compare Timer 1 into the Compare/Capture registers. Interrupts can be generated selectively at each transition of the capture input signal.

Capture mode is selected by writing the mode select registers CMSEL1 and CMSEL0 with the appropriate values. The bit combinations in CMSEL0 and CMSEL1 also define the signal transition type (falling/rising edge) which generates a capture event. If a CAPCOM channel is enabled for capture mode, its CCx input is sampled with 1/(4 TCL) (i.e. $2 f_{OSC}$ = twice external oscillator clock rate).

Consecutive capture events, generated through signal transitions at a CCx capture input, overwrite the corresponding 16-bit Compare/Capture register contents. This must be considered when successive signal transitions are processed.

6.3.2.7 Trap Function of the CAPCOM Unit in Compare Mode

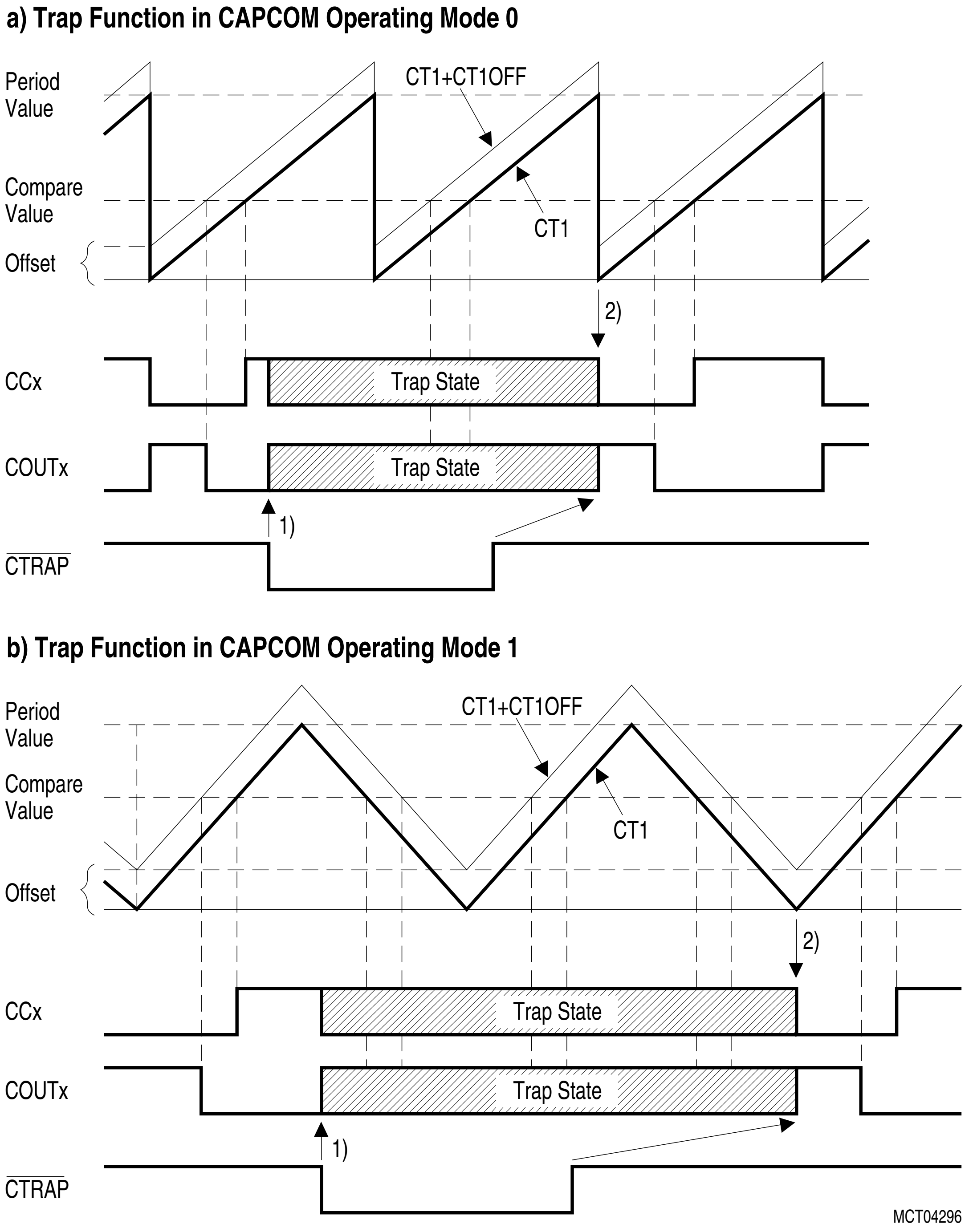
When a channel of the CAPCOM unit operates in compare mode, its output lines can be decoupled in trap mode from the CAPCOM pulse generation. The trap mode is controlled by the external signal $\overline{\text{CTRAP}}$. The $\overline{\text{CTRAP}}$ signal is sampled at each phase of the oscillator clock cycle. If a low is detected, the trap flag TRF of register TRCON is set and CCx or COUTx compare outputs are switched immediately to the logic state as defined by the bits in COTRAP if that particular channel has been enabled for trap function. The compare outputs of the channels which are not enabled for trap function will have their last output levels maintained. For safety reasons, it is recommended that trap function be enabled. If CT1RES = 0, Compare Timer 1 continues its operation but no compare output signal will be generated. If CT1RES = 1, Compare Timer 1 is reset when $\overline{\text{CTRAP}}$ becomes active. When $\overline{\text{CTRAP}}$ is sampled inactive (high) again, the compare channel outputs are synchronously switched to the compare channel output signal generation when Compare Timer 1 has reached the count value 0000_H.

The trap function is controlled by bits in the TRCON register. The general enable function of the external $\overline{\text{CTRAP}}$ signal is controlled by one bit (TRPEN). Further, each CAPCOM compare channel output can be enabled/disabled selectively for trap function.

Figure 6-31 shows the trap function for the two outputs CCx and COUTx of one compare channel x. The timing diagram implies that the trap function is enabled at the CCx and COUTx outputs.

At reference point 1) in **Figure 6-31** $\overline{\text{CTRAP}}$ becomes active and at reference point 2) the trap state is released again synchronously to the Compare Timer 1 count state 0000_H. If the trap function is enabled and $\overline{\text{CTRAP}}$ becomes active, bit TRF (trap flag) in SFR TRCON is set and a CCU emergency interrupt will be generated if the related interrupt enable bits are set. The flag TRF is level sensitive and must be cleared by software.

The trap function used in block commutation mode differs from the trap function described above. In particular, the synchronization scheme is different (see **Chapter 6.3.4.6**).



Note: The state of the CCx and COUTx signals in trap state is defined by the corresponding bits in COTRAP

Figure 6-31 Trap Function of the CAPCOM Unit

6.3.2.8 CAPCOM Registers

The CAPCOM unit of the C508 contains several special function registers. Table 6-7 provides an overview of the CAPCOM related registers.

Table 6-7 Special Function Registers of the CAPCOM Unit

Unit	Symbol	Description	Address
CAPCOM Capture / Compare Unit	CT1CON	Compare Timer 1 Control Register	$E1_H$
	CCPL	Compare Timer 1 Period Register, Low Byte	DE_H
	CCPH	Compare Timer 1 Period Register, High Byte	DF_H
	CT1OFL	Compare Timer 1 Offset Register, Low Byte	$E6_H$
	CT1OFH	Compare Timer 1 Offset Register, High Byte	$E7_H$
	CMSEL0	Capture/Compare Mode Select Register 0	$E3_H$
	CMSEL1	Capture/Compare Mode Select Register 1	$E4_H$
	CCL0	Capture/Compare Register 0, Low Byte	$F2_H$
	CCH0	Capture/Compare Register 0, High Byte	$F3_H$
	CCL1	Capture/Compare Register 1, Low Byte	$F4_H$
	CCH1	Capture/Compare Register 1, High Byte	$F5_H$
	CCL2	Capture/Compare Register 2, Low Byte	$F6_H$
	CCH2	Capture/Compare Register 2, High Byte	$F7_H$
	CCIR	Capture/Compare Interrupt Request Flag Register	$E5_H$
	CCIE	Capture/Compare Interrupt Enable Register	$D6_H$
	COINI	Compare output initialization register	$E2_H$
	TRCON	Trap Enable Register	FF_H
	COTRAP	Compare Output in Trap State Register	$F9_H$

The following sections describe the CAPCOM registers in detail.

Writing the CAPCOM Period/Offset/Compare Registers on-the-Fly

If Compare Timer 1 is running, then the period, offset or compare registers can be written with modified values for generating new periods or duty cycles of the compare output signals. For proper synchronization purposes, a special mechanism for updating of the 16-bit offset, period, and compare registers is implemented in the C508. This mechanism is based on shadow latches. When new values for offset, period, or compare registers have been written into the shadow latches, the real register update operation must be initiated by setting bit STE1 (shadow transfer enable) in SFR CT1CON. When this bit is set, the content of the shadow latches is transferred to the real registers when Compare Timer 1 has reached its period value or zero value. This applies to both operating modes 0 and 1.

When the register transfer has been executed, STE1 is reset by hardware. So the software can recognize when the register transfer has occurred.
When Compare Timer 1 is started by setting the run bit CT1R the first time after reset, a shadow register transfer into the real registers is automatically executed. In this case STE1 must not be set.

Care must be taken when programming a new compare value. If the new compare value is greater than or equal to the period value, the reload should be delayed till the next zero match (Compare Timer 1 reaches 0000_H) instead of the approaching period match (Compare Timer 1 reaches period value). This can be achieved by setting bit STE1 only in the period match interrupt service routine.

If the desired compare value is less than the offset value, the COUT bits in COINI register must be inverted first, before the reload is allowed.

Compare Timer 1 Control Register

The 16-bit Compare Timer 1 is controlled by the bits of the CT1CON register. With this register the count mode, the trap interrupt enable, the compare timer start/stop and reset, and the timer input clock rate is controlled.

Special Function Register CT1CON (Address $E1_H$) **Reset Value: 00010000_B**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$E1_H$	CTM	ETRP	STE1	CT1RES	CT1R	CLK2	CLK1	CLK0	CT1CON

Bit	Function
CTM	**Compare Timer 1 operating mode selection** CTM = 0 selects operating mode 0 (up count) and CTM = 1 selects operating mode 1 (up/down count) for Compare Timer 1.
ETRP	**CCU emergency trap interrupt enable** If ETRP = 1, the emergency interrupt for the CCU trap signal is enabled.
STE1	**CAPCOM unit shadow latch transfer enable** When STE1 is set, the content of the Compare Timer 1 period, Compare and offset registers (CCPH, CCPL, CCHx, CCLx, CT1OFH, CT1OFL) is transferred to its real registers when Compare Timer 1 reaches the next time the period value or value 0000_H. After the shadow transfer event, STE1 is reset by hardware.
CLK2 CLK1 CLK0	**Compare Timer 1 input clock selection** The input clock for the Compare Timer 1 is derived from the clock rate f_{OSC} of the C508 via a programmable prescaler. The following table shows the programmable prescaler ratios.

CLK2	CLK1	CLK0	Function	
0	0	0	Compare timer 1 input clock is	$2\,f_{OSC}$
0	0	1	Compare timer 1 input clock is	f_{OSC}
0	1	0	Compare timer 1 input clock is	$f_{OSC}/2$
0	1	1	Compare timer 1 input clock is	$f_{OSC}/4$
1	0	0	Compare timer 1 input clock is	$f_{OSC}/8$
1	0	1	Compare timer 1 input clock is	$f_{OSC}/16$
1	1	0	Compare timer 1 input clock is	$f_{OSC}/32$
1	1	1	Compare timer 1 input clock is	$f_{OSC}/64$

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>CT1RES
CT1R</td><td><b>Compare timer 1 reset control</b>
<b>Compare timer 1 run/stop control</b>
These two bits control the start, stop, and reset function of Compare Timer 1. CT1RES is used to reset Compare Timer 1 and CT1R is used to start and stop the Compare Timer 1. The following table shows the functions of these two bits:
<table>
<tr><th>CT1RES</th><th>CT1R</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare Timer 1 is stopped and holds its value; the compare outputs stay in the logic state as they are.</td></tr>
<tr><td>1</td><td>0</td><td>Compare Timer 1 is stopped and reset; compare outputs are set to the logic state as defined in SFR COINI (default after reset).</td></tr>
<tr><td>0</td><td>0 → 1</td><td>Compare Timer 1 starts. Before CT1R is set the first time, the CMSEL register should be programmed (enable Capture/Compare functions).</td></tr>
<tr><td>1</td><td>0 → 1</td><td>Compare Timer 1 starts running from count value 0000_H; compare outputs are set to the logic state as defined in SFR COINI.</td></tr>
<tr><td>0</td><td>1 → 0</td><td>Compare Timer 1 is stopped and holds its value; the Compare outputs drive their actual logic state.</td></tr>
<tr><td>1</td><td>1 → 0</td><td>Compare Timer 1 is stopped and reset to 0000_H; Compare outputs are set to the logic state as defined in SFR COINI.</td></tr>
</table>
<i>Note for Capture mode:
Setting CT1R = 0 and CT1RES = 1 after a capture event will destroy the value stored in the capture register CCx. Therefore, CT1RES should be set to 0 in capture mode. Reason: if CT1R = 0 and CT1RES = 1 all shadow registers are transparent (switched directly) to the real registers.</i></td></tr>
</table>

*Note: When software power-down mode is entered with CT1RES bit of SFR CT1CON set, the Compare Timer 1 is reset after the execution of a wake-up from power-down mode procedure. When CT1RES is **cleared** before software power-down mode is entered and a wake-up from power-down mode procedure has been executed, the Compare Timer 1 is not reset. Depending on the state of bit CT1R at power-down mode entry, the Compare Timer 1 either stops (CT1R = 0) or continues (CT1R = 1) counting after a wake-up from power-down mode procedure. Further details of the power-down mode are provided in **Chapter 9.2**.*

Compare Timer 1 Period Registers

The Compare Timer 1 period registers CCPH and CCPL store the 16-bit value for the Compare Timer 1 count period. CCPH holds the high byte of the 16-bit period value and CCPL holds the low byte. If CCPH/CCPL is written, shadow latches are always loaded. The contents of these shadow latches are transferred to the real registers when STE1 is set and the Compare Timer 1 reaches its period value (operating mode 0) or count value 0000_H (operating mode 1). When the Compare Timer 1 period registers are read, shadow latches are always accessed.

Special Function Register CCPL (Address DE_H) **Reset Value: 00_H**
Special Function Register CCPH (Address DF_H) **Reset Value: 00_H**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
DE_H	.7	.6	.5	.4	.3	.2	.1	LSB	CCPL
DF_H	MSB	.6	.5	.4	.3	.2	.1	.0	CCPH

Bit	Function
CCPL.7 - 0	**Compare Timer 1 period value, low byte** The 8-bit value in the CCPL register is the low byte of the 16-bit period value of Compare Timer 1 (shadow latch).
CCPH.7 - 0	**Compare Timer 1 period value, high byte** The 8-bit value in the CCPH register is the high byte of the 16-bit period value of Compare Timer 1 (shadow latch).

Compare Timer 1 Offset Registers

The CT1OFH and CT1OFL registers contain the value for the Compare Timer 1. CT1OFH holds the high byte of the 16-bit offset value and CT1OFL holds the low byte. For the detection of a compare match event, which results in changing polarity of a COUTx compare output signal, the content of CT1OFH/CT1OFL is always added to the actual value of the Compare Timer 1. The value stored in the offset registers has no effect on the signal generation at the CCx compare outputs.

If the Compare Timer 1 offset registers are written, shadow latches are always loaded. The content of these shadow latches is transferred to the real registers when STE1 is set and the Compare Timer 1 reaches its period value or count value 0000$_H$. When the Compare Timer 1 offset registers are read, shadow latches are always accessed.

Special Function Register CT1OFL (Address E6$_H$) Reset Value: 00$_H$
Special Function Register CT1OFH (Address E7$_H$) Reset Value: 00$_H$

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
E6$_H$	.7	.6	.5	.4	.3	.2	.1	LSB	CT1OFL
E7$_H$	MSB	.6	.5	.4	.3	.2	.1	.0	CT1OFH

Bit	Function
CT1OFL.7 - 0	**8-bit Compare Timer 1 offset value, low byte** The 8-bit value in the CT1OFL register is the low byte of the offset value for Compare Timer 1 (shadow latch).
CT1OFH.7 - 0	**8-bit Compare Timer 1 offset value, high byte** The 8-bit value in the CT1OFH register is the high byte of the offset value for Compare Timer 1 (shadow latch).

To generate correct dead times for PWM signals, the offset value stored in CT1OFH/CT1OFL must be lower than the values stored in the compare registers.

Capture/Compare Channel Mode Select Registers

The capture/compare channels of the CAPCOM unit can operate individually either in compare mode or in capture mode. The CMSEL0 and CMSEL1 registers contain the mode select bits for the CAPCOM unit.

Special Function Register CMSEL0 (Address E3$_H$) Reset Value: 00$_H$
Special Function Register CMSEL1 (Address E4$_H$) Reset Value: 00$_H$

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
E3$_H$	CMSEL 13	CMSEL 12	CMSEL 11	CMSEL 10	CMSEL 03	CMSEL 02	CMSEL 01	CMSEL 00	CMSEL0
	CAPCOM Channel 1				CAPCOM Channel 0				

	7	6	5	4	3	2	1	0	
E4$_H$	ESMC	NMCS	0	0	CMSEL 23	CMSEL 22	CMSEL 21	CMSEL 20	CMSEL1
					CAPCOM Channel 2				

Bit	Function
ESMC	**Enable software controlled multi-channel PWM modes** If ESMC = 0, switching of the follower state in 4-/5-/6-phase multi-channel PWM mode is controlled by Compare Timer 1 reaching its period value. If ESMC = 1, switching of the follower state in 4-/5-/6-phase multi-channel PWM mode is controlled by bit NMCS.
NMCS	**Next multi-channel PWM state** Setting bit NMCS (with ESMC set) will select the follower state in the 4/5/6-phase multi-channel PWM mode, which is taken into account at the output pins, when Compare Timer 1 is 0. Bit NMCS is reset by hardware in the next clock cycle after it has been set.
CMSELx3 x = 0-2	**Switching Compare Timer 2 output signal to COUTx** If CMSELx3 is set and compare mode is selected for the outputs COUTx, the output signal of the 10-bit Compare unit, typically a higher frequency signal, is switched (modulated) to the COUTx pin. The state of the corresponding COINI bit at the start of Compare Timer 1 defines the logic level of the CAPCOM channel output signal at which the COMP output signal is output to COUTx. COINI is set: The COMP output is switched to COUTx during the low phase of the CAPCOM channel X signal. COINI is cleared: The COMP output is switched to COUTx during the high phase of the CAPCOM channel X signal.

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>CMSELx2- 0
x = 0-2</td><td><b>CAPCOM Capture/Compare mode enable bits</b>
The CMSEL registers are used to select/enable the operating mode and the output/input pin configuration of the capture/compare channels. Each CAPCOM channel can be programmed individually for either compare or capture operation.
<table>
<tr><th>CMSEL x2</th><th>CMSEL x1</th><th>CMSEL x0</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Compare outputs disabled;
No compare output signal is generated; CCx and COUTx are normal I/O pins.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Compare output on pin CCx enabled;
COUTx is normal I/O pin.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Compare output on pin COUTx enabled;
CCx is normal I/O pin.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Compare outputs on pins CCx and COUTx enabled.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Capture mode enabled; signal transitions at CCx do not generate a capture event. COUTx is a normal I/O pin or analog input pin.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Capture mode enabled; CCx is configured as a Capture input and a rising edge at CCx transfers compare timer 1 content into the capture register. COUTx is a normal I/O pin or analog input pin.</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Capture mode enabled; CCx is configured as a Capture input and a falling edge at CCx transfers compare timer 1 content into the capture register. COUTx is a normal I/O pin or analog input pin.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Capture mode enabled; CCx is configured as a capture input. Rising and falling edge at CCx transfer the compare timer 1 content into the capture register. COUTx is a normal I/O pin or analog input pin.</td></tr>
</table>
<i>Note: Only CC0/COUT0 can be analog inputs if not selected as compare output.</i></td></tr>
</table>

In compare mode, the two output signals of a CAPCOM channel can be enabled selectively. In capture mode, the type of signal transition which will generate a capture event can be chosen.

Capture/Compare Registers of CAPCOM Unit

The capture/compare registers are 16-bit registers, organized as two 8-bit byte-wide registers. Each of the three CAPCOM channels has one capture/compare register. In compare mode, they hold a compare value which typically defines the duty cycle of the output signals. In capture mode, the actual Compare Timer 1 value is transferred into the Capture/Compare registers at a Capture event.

If CCLx/CCHx is written, shadow latches are always loaded. The content of these shadow latches is transferred to the real registers when STE1 is set and the Compare Timer 1 reaches its period value (operating mode 0) or count value 0000$_H$ (operating mode 1). When the Capture/Compare registers are read, the real registers are always accessed because of capture mode.

Special Function Registers CCL0/CCH0 (Addresses F2$_H$ / F3$_H$) Reset Value: 00$_H$
Special Function Registers CCL1/CCH1 (Addresses F4$_H$ / F5$_H$) Reset Value: 00$_H$
Special Function Registers CCL2/CCH2 (Addresses F6$_H$ / F7$_H$) Reset Value: 00$_H$

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
F2$_H$	.7	.6	.5	.4	.3	.2	.1	LSB	CCL0
F3$_H$	MSB	.6	.5	.4	.3	.2	.1	.0	CCH0
F4$_H$	.7	.6	.5	.4	.3	.2	.1	LSB	CCL1
F5$_H$	MSB	.6	.5	.4	.3	.2	.1	.0	CCH1
F6$_H$	.7	.6	.5	.4	.3	.2	.1	LSB	CCL2
F7$_H$	MSB	.6	.5	.4	.3	.2	.1	.0	CCH2

Bit	Function
CCLx.7 - 0 x = 0-2	**Capture/Compare value, low byte** The 8-bit value in the CCLx register is the low byte of the 16-bit capture/compare value of channel x.
CCHx.7 - 0 x = 0-2	**Capture/Compare value, high byte** The 8-bit value in the CCHx register is the low byte of the 16-bit capture/compare value of channel x.

Capture/Compare Interrupt Request Flags/Register

The interrupt flags of the CAPCOM capture/compare match and Compare Timer 1 interrupt are located in the register CCIR. All CAPCOM capture/compare match interrupt flags are set by hardware and must be cleared by software. A capture/compare match interrupt is generated by setting of a CCxR bit (x = 0-2) if the corresponding enable bits are set. The Compare Timer 1 interrupt is triggered by the CT1FP or CT1FC bits of SFR CCIR.

Special Function Register CCIR (Address E5$_H$) **Reset Value: 00$_H$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
E5$_H$	CT1FP	CT1FC	CC2F	CC2R	CC1F	CC1R	CC0F	CC0R	CCIR
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

Bit	Function
CT1FP	**Compare Timer 1 period flag** Compare Timer 1 operating mode 0: CT1FP is set if Compare Timer 1 reaches the period value. Compare Timer 1 operating mode 1: CT1FP is set if Compare Timer 1 reaches the period value and changes the count direction from up- to down counting Bit CT1FP must be cleared by software. If Compare Timer 1 interrupt is enabled, the setting of CT1FP will generate a Compare Timer 1 interrupt.
CT1FC	**Compare Timer 1 count direction change flag** This flag can only be set if Compare Timer 1 runs in operating mode 1 (CTM = 1). CT1FC is set when Compare Timer 1 reaches count value 0000$_H$ and changes the count direction from down- to up-counting. If Compare Timer 1 interrupt is enabled, the setting of CT1FC will generate a Compare Timer 1 interrupt. Bit CT1FC must be cleared by software.

Bit	Function
CCxR x = 0-2	**Capture/Compare match on up-count flag** **Capture Mode:** CCxR is set at a low-to-high transition (rising edge) of the corresponding CCx Capture input signal. **Compare Mode:** CCxR is set if the Compare timer 1 value matches the Compare register CCx value during the up-count phase.
CCxF x = 0-2	**Capture/Compare match on down-count flag** **Capture Mode:** CCxF is set at a high-to-low transition (falling edge) of the corresponding CCx capture input signal. **Compare Mode:** CCxF is set if the Compare Timer 1 value matches the compare register CCx value during the down-count phase (only in Compare Timer 1 operating mode 1).

Capture/Compare Interrupt Enable Register

The bits of the interrupt enable register CCIE control the specific interrupt enable/disable functions of the CAPCOM part of the Capture/Compare unit.

The bits ECTP and ECTC control the Compare Timer 1 period/count change interrupt. Depending on the mode in which Compare Timer 1 is running, interrupts can be generated at a period match or a count direction change event.

The lower 6 bits of CCIE are the CAPCOM channel specific interrupt enable/disable control bits for the capture or compare match interrupt. The functions of these bits depend on the selected mode (capture or compare) of a capture/compare channel. In compare mode, compare channel specific interrupts can be generated at a match event between compare register content and compare timer 1 count value during the up- or down-counting phase of Compare Timer 1. In capture mode, capture channel specific interrupts can be generated selectively at rising or falling or both edges of the capture input signals at CCx.

Special Function Registers CCIE (Address $D6_H$) Reset Value: 00_H

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$D6_H$	ECTP	ECTC	CC2FEN	CC2REN	CC1FEN	CC1REN	CC0FEN	CC0REN	CCIE

Bit	Function
ECTP	**Enable Compare Timer 1 period interrupt** If ECTP = 0, the Compare Timer 1 period interrupt is disabled. Compare Timer 1 operating mode 0: If ECTP = 1, an interrupt is generated when Compare Timer 1 reaches the period value. Compare timer 1 operating mode 1: If ECTP = 1, an interrupt is generated when Compare Timer 1 reaches the period value and changes the count direction from up- to down-counting.

Bit	Function
ECTC	**Enable Compare Timer 1 count direction change interrupt status** If ECTC = 0, the Compare Timer 1 count change interrupt is disabled. Compare timer 1 operating mode 0: Bit has no effect on the interrupt generation. Compare timer 1 operating mode 1: If ECTC = 1, an interrupt is generated when Compare Timer 1 reaches count value 0000_H and changes its count direction from down- to up-counting.
CCxREN (x = 0-2)	**Capture/Compare rising edge interrupt enable** Capture Mode: If CCxREN is set, an interrupt is generated at a low-to-high transition (rising edge) of the corresponding CCx input signal. Compare Mode: If CCxREN is set, an interrupt is generated if the Compare Timer 1 value matches the compare register CCx value during the up-counting phase of the Compare Timer 1. This function is available in both Compare Timer 1 operating modes.
CCxFEN (x = 0-2)	**Capture/Compare falling edge interrupt enable** Capture Mode: If CCxFEN is set, an interrupt is generated at a high-to-low transition (falling edge) of the corresponding CCx input signal. Compare Mode: If CCxFEN is set, an interrupt is generated only in Compare timer mode 1 if the Compare Timer 1 value matches the Compare register CCx value during the down-counting phase of the Compare Timer 1. This function is available only in Compare Timer 1 operating mode 1.

Compare Output Initialization Register COINI

The six lower bits of the COINI register define the initial values (passive levels) of the Port 1 lines, which are programmed to be used as a compare output. If an output of the CAPCOM unit is enabled for compare mode operation by writing the corresponding bit combination into the CMSEL0/CMSEL1 registers, the compare output is switched into push-pull mode and starts driving an initial logic level as defined by the bits of the COINI register.

Bit COUTXI controls an inverter for the COMP unit output signal, when it is wired to the CCx and COUTx outputs in burst or multi-channel PWM mode. COUT3I defines the initial logic level at COUT3 before Compare Timer 2 is started as well as the logic state when COUT3 is disabled by setting bit ECT2O in SFR CT2CON (see **Figure 6-32**).

The COINI register should be written prior to the starting of the compare timers. Any write operation to the COINI register when the compare timer is running will affect the compare output signals immediately and drive the logic value as defined by the bits of COINI.

A PWM output signal of the C508 basically consists of two phases, an inactive phase and an active phase. The inactive phase of a PWM output signal is defined by the bit in the register COINI. A ‘1’ in bit location 0 to 5 of COINI defines the high level of the corresponding PWM compare output signal as its inactive phase. With a ‘0’, a low level is selected as the inactive phase.

Special Function Register COINI (Address E2$_H$) **Reset Value: FF$_H$**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
E2H	COUT3I	COUTXI	COUT2I	CC2I	COUT1I	CC1I	COUT0I	CC0I	COINI
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

Bit	Function
COUT3I	**COUT3 initial logic level** This bit defines the initial logic state of the output COUT3 before Compare Timer 2 is started the first time. Further, COUT3I defines the logic state of output COUT3 when bit ECT2O (CT2CON.6) is reset (COUT3 disabled).
COUTXI	**Compare Timer 1 output signal inversion in burst and block commutation** When COUTXI is set, the output signal of Compare Timer 2 which is wired to the compare outputs COUTx (x = 0-2) in burst or block commutation mode is inverted.
CCxI, COUTxI (x = 0-2)	**Compare output initial value** Bits at even bit positions (0, 2, 4) are assigned to the CCx compare outputs. Bits at odd bit positions (1, 3, 5) are assigned to the COUTx compare outputs. CCxI, COUTxI = 0: If Compare Timer 1 is not running (after reset), an output CCx/COUTx (x = 0-2) is switched into push-pull mode and starts driving an initial value of 0 when this CCx/COUTx output is programmed as compare output by writing the corresponding bit combination into the CMSEL0/CMSEL1 registers. CCxI, COUTxI = 1: If Compare Timer 1 is not running (after reset), an output CCx/COUTx (x = 0-2) is switched into push-pull mode and starts driving an initial value of 1 when this CCx/COUTx output is programmed as compare output by writing the corresponding bit combination into the CMSEL0/CMSEL1 registers. The COINI values are valid only for capture/compare outputs enabled for compare mode operation.

Trap Enable Register

The trap enable register TREN is used to enable selectively the compare outputs of the three CAPCOM channels for switching it into high or low level in the trap state as defined by the bits of the COTRAP register. Additionally, for a general enable of the trap function, bit TRPEN must be set. The TRF flag indicates when a low level is detected at the $\overline{\text{CTRAP}}$ input signal.

Special Function Register TRCON (Address FF_H) **Reset Value: 00_H**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
FF_H	TRPEN	TRF	TREN5	TREN4	TREN3	TREN2	TREN1	TREN0	TRCON
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

Bit	Function

TRPEN	**External $\overline{\text{CTRAP}}$ trap function enable bit**
	This bit is a general enable bit for the trap function of the $\overline{\text{CTRAP}}$ input signal.
	TRPEN = 0: External trap input $\overline{\text{CTRAP}}$ is disabled (default after reset).
	TRPEN = 1: External trap input $\overline{\text{CTRAP}}$ is enabled;

TRF	**Trap flag** TRF is set by hardware if the trap function is enabled (TRPEN = 1) and the $\overline{\text{CTRAP}}$ level becomes active (low). If enabled, an interrupt is generated when TRF is set. TRF must be reset by software.
TREN5-0	**Trap enable control bits** Bits at even bit positions (0, 2, 4) are assigned to the CCx Compare outputs. Bits at odd bit positions (1, 3, 5) are assigned to the COUTx Compare outputs. TRENx = 0: Compare channel output provides CAPCOM output signal in trap state. TRENx = 1: Compare channel output is enabled to set the logic level of the compare output CCx or COUTx in the trap state to a logic state as defined by the corresponding bits of the COTRAP register. When writing TREN0-5, bit TRF should be reset to 0. Otherwise, setting TREN0-5 will generate a software trap interrupt.

Compare Output in Trap State Register

The six lower bits of the COTRAP register define the values of Port 1 pins 2 to 7, which are programmed to be used as compare outputs, when a trap state is entered. Bit 6 is reserved and must always be written with a ‘0’. Bit 7 selects either one of the two block commutation tables for rotate left that is provided.

Special Function Register COTRAP (Address F9$_H$) **Reset Value: 00$_H$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
F9$_H$	BCTSEL	RES	COUT2T	CC2T	COUT1T	CC1T	COUT0T	CC0T	COTRAP
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

Bit	Function
BCTSEL	**Block Commutation Table (Rotate Left) Select** BCTSEL = 0: The table for 60° phase angle will be selected. BCTSEL = 1: The table for 0° phase angle will be selected.
RES	**Reserved** **This bit must always be written with a ‘0’.** Writing a ‘1’ to this bit is prohibited.
CCxT, COUTxT (x = 0-2)	**Compare output level in trap condition** Bits at even positions (0, 2, 4) are assigned to the CCx compare outputs. Bits at odd positions (1, 3, 5) are assigned to the COUTx compare outputs. CCxT, COUTxT = 0: If the compare timer is running, the compare channel output CCx, COUTx (x = 0-2) will be switched to 0 level in trap state if the channel is enabled for trap function. CCxT, COUTxT = 1: If the compare timer is running, the compare channel output CCx, COUTx (x = 0-2) will be switched to 1 level in trap state if the channel is enabled for trap function.

6.3.3 Compare (COMP) Unit Operation

The Capture/Compare Unit of the C508 also provides a 10-bit Compare Unit (COMP) which operates as a single channel pulse generator with a pulse width modulated output signal. This output signal is available at the output pin COUT3 of the C508. In the combined multi-channel PWM modes and in burst mode of the CAPCOM unit the output signal of the COMP unit can also be switched to the output signals COUTx or CCx. Figure 6-32 shows the block diagram and the pulse generation scheme of the COMP unit (for example: the initial value of COUT3 is set to 0).

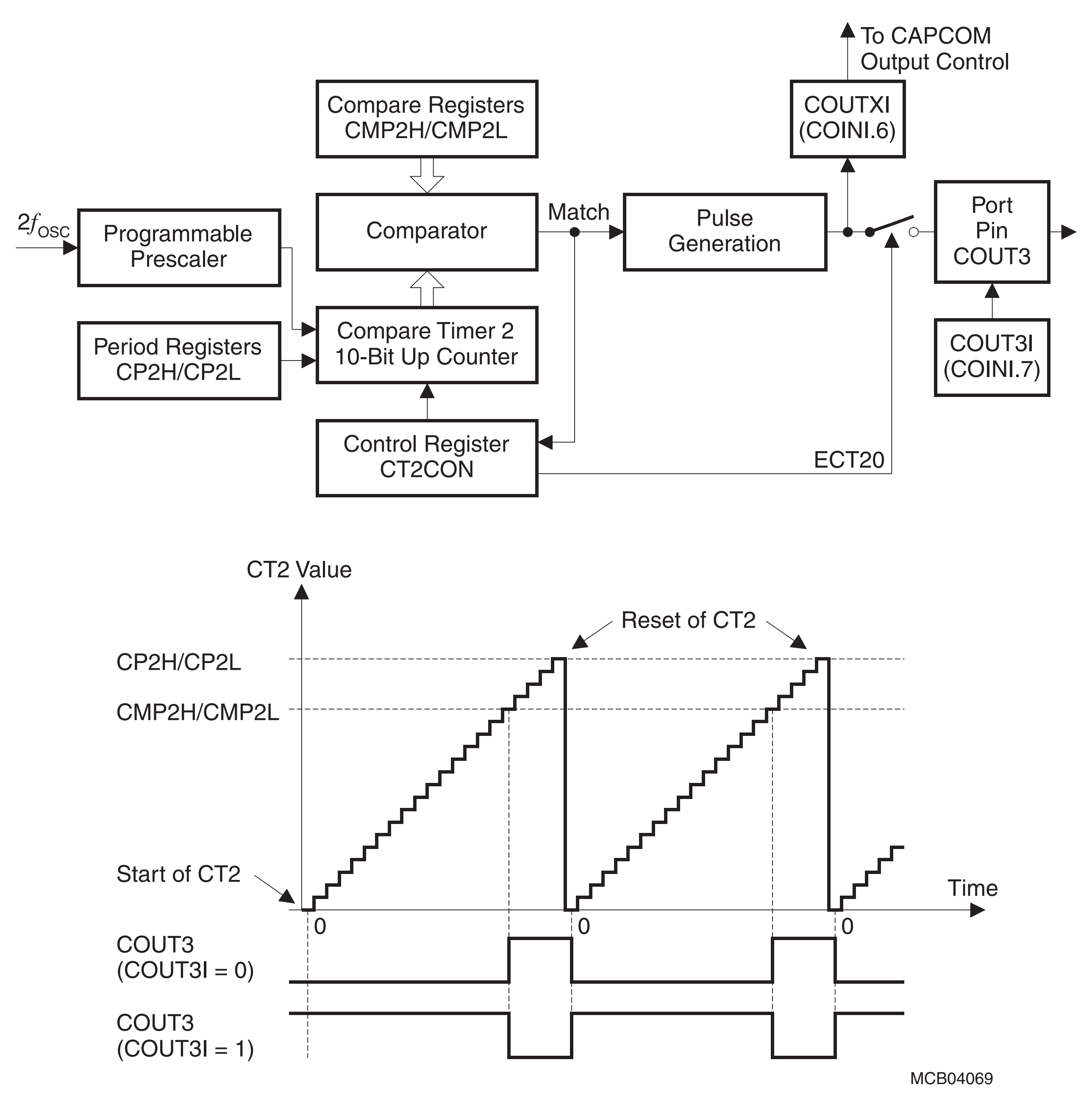


Figure 6-32 COMP Unit: Block Diagram and Pulse Generation Scheme

The COMP unit has a 10-bit up-counter (Compare Timer 2, CT2) which starts counting from 000_H up to the value stored in the period register and then is again reset. This Compare Timer 2 operation is similar to the operating mode 0 of Compare Timer 1. When the count value of CT2 matches the value stored in the compare registers CMP2H/CMP2L, COUT3 toggles its logic state. When Compare Timer 2 is reset to 000_H, COUT3 toggles again its logic state.

COUT3 is only an output pin. After a reset operation COUT3 drives a high level as defined by the reset value (= 1) of bit COUT3I of SFR COINI. When Compare Timer 2 is running (bit CT2R in SFR CT2CON is set), bit ECT2O in SFR CT2CON allows the disconnection of COUT3 from Compare Timer 2 signal generation. In this case, the logic value of COUT3I (bit COINI.7) is put to the COUT3 output. When ECT2O is set thereafter, the Compare Timer 2 output signal is again switched to the COUT3 output.

In the combined multi-channel PWM modes and in the burst mode, the Compare Timer 2 output signal can also be switched to the CAPCOM output pins COUT0, COUT1, and COUT3. In these modes, the polarity of the modulated output signal at COUT2-0 can be inverted by setting bit COUTXI (COINI.6)

6.3.3.1 COMP Registers

The COMP unit has five SFRs which are listed in **Table 6-8**.

Table 6-8 Special Function Registers of the COMP Unit

Unit	Symbol	Description	Address
COMP Compare Unit	CT2CON	Compare Timer 2 control register	$F1_H$
	CP2L	Compare Timer 2 period register, low byte	$D2_H$
	CP2H	Compare Timer 2 period register, high byte	$D3_H$
	CMP2L	Compare Timer 2 Compare register, low byte	$D4_H$
	CMP2H	Compare Timer 2 Compare register, high byte	$D5_H$

The Compare Timer 2 period and compare registers store a 10-bit value, organized in two bytes. For proper synchronization purposes, these registers are not written directly. Each value of a write operation to these registers is stored in shadow latches. The transfer of these shadow latches into the real registers is synchronized with the Compare Timer 2 value 000_H and controlled by bit STE2. When the period or compare value is changed by writing the corresponding SFR, the setting of bit STE2 (CT2CON.5) enables the write transfer of the shadow registers into the real registers. This shadow latch transfer happens when the Compare Timer 2 reaches the count value 000_H the next time after STE2 has been set. With the automatic transfer of the shadow latches to the real registers, bit STE2 is reset by hardware. When the Compare Timer 2 period and compare registers are initialized after reset, bit STE2 must also be set to enable the shadow latch transfer when Compare Timer 2 is started the first time.

Note: Read operations with the Compare Timer 2 period and compare registers always access the shadow registers and not the real registers.

Compare Timer 2 Control Register

The 10-bit Compare Timer 2 is controlled by the bits of the CT2CON register. With this register the count mode, the timer input clock rate, and the compare timer reset function is controlled.

Special Function Register CT2CON (Address $F1_H$) **Reset Value: 00010000_B**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$F1_H$	CT2P	ECT2O	STE2	CT2RES	CT2R	CLK2	CLK1	CLK0	CT2CON

Bit	Function
CT2P	**Compare Timer 2 period flag** When the Compare Timer 2 value matches with the Compare Timer 2 period register value, bit CT2P is set. If the Compare Timer 2 interrupt is enabled, the setting of CT2P will generate a Compare Timer 2 interrupt. Bit CT2P must be cleared by software.
ECT2O	**Enable Compare timer 2 output** When ECT2O is cleared and Compare Timer 2 is running, output COUT3 is put into the logic state as defined by bit COUT3I which is located in SFR COINI.6. When ECT2O is set and Compare Timer 2 is running, the Compare Timer 2 output COUT3 is enabled and outputs the PWM signal of the COMP unit.
STE2	**COMP unit shadow latch transfer enable** When STE2 is set, the content of the Compare Timer 2 period and compare latches (CP2H, CP2L, CMP2H, CMP2L) is transferred to its real registers when Compare Timer 2 reaches the period value. After the shadow transfer event, STE2 is reset by hardware.

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>CT2RES
CT2R</td><td><b>Compare Timer 2 reset control</b>
Compare Timer 2 run/stop control
These two bits controls the start, stop, and reset function of the Compare Timer 2. CT2RES is used to reset Compare Timer 2; and CT2R is used to start or stop Compare Timer 2. The following table shows the functions of these two bits:
<table>
<tr><th>CT2RES</th><th>CT2R</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Compare Timer 2 is stopped; compare output COUT3 stays in the logic state as it is.</td></tr>
<tr><td>0</td><td>1</td><td>Compare Timer 2 is running. If CT2R is set the first time after reset, COUT3 is set to the logic state as defined by bit COUT3I of SFR COINI.</td></tr>
<tr><td>1</td><td>0</td><td>Compare Timer 2 is stopped and reset. The output COUT3 is set to the logic state as defined by bit COUT3I of SFR COINI (default after reset).</td></tr>
<tr><td>1</td><td>1</td><td>Compare Timer 2 is further running.</td></tr>
</table>
ECT2O must be set for COUT3 signal output enable.</td></tr>
<tr><td>CLK2
CLK1
CLK0</td><td><b>Compare Timer 2 input clock selection</b>
The input clock for the Compare Timer 2 is derived from the clock rate f_{OSC} of the C508 via a programmable prescaler. The following table shows the programmable prescaler ratios.
<table>
<tr><th>CLK2</th><th>CLK1</th><th>CLK0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Compare Timer 2 input clock is $2 f_{OSC}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Compare Timer 2 input clock is f_{OSC}</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Compare Timer 2 input clock is $f_{OSC}/2$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Compare Timer 2 input clock is $f_{OSC}/4$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Compare Timer 2 input clock is $f_{OSC}/8$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Compare Timer 2 input clock is $f_{OSC}/16$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Compare Timer 2 input clock is $f_{OSC}/32$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Compare Timer 2 input clock is $f_{OSC}/64$</td></tr>
</table></td></tr>
</table>

Note: With a reset operation (external or internal) Compare Timer 2 is reset (000_H) and stopped. When software power-down mode is entered with CT2RES bit of SFR CT2CON set, the Compare Timer 2 is reset after the execution of a wake-up from *power-down mode procedure. When CT2RES is* ***cleared*** *before software power down mode is entered and a wake-up from power-down mode procedure has been executed, the Compare Timer 2 is not reset. Depending on the state of bit CT2R at power down mode entry, the Compare Timer 2 either stops (CT2R = 0) or continues (CT2R = 1) counting after a wake-up from power-down mode procedure. Further details of the power-down mode are described in* ***Chapter 9.2****.*

Compare Timer 2 Period Registers

The Compare Timer 2 period registers CP2L/CP2H hold the 10-bit value for the Compare Timer 2 period. When the Compare Timer 2 value is equal to the value stored in the period register, the COUT3 signal changes from inactive to active state. If CP2H/ CP2L is written, only shadow latches are written. The content of these latches is transferred to the real registers at compare timer count value 000_H, using bit STE2 of SFR CT2CON.

When the Compare Timer 2 period registers CP2L/CP2H are read, the shadow registers are always accessed.

Special Function Register CP2L (Address $D2_H$) **Reset Value: 00_H**
Special Function Register CP2H (Address $D3_H$) **Reset Value: $XXXXXX00_B$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$D2_H$	.7	.6	.5	.4	.3	.2	.1	.0	CP2L
$D3_H$	–	–	–	–	–	–	.1	.0	CP2H

Bit	Function
CP2L.7 - 0	**Compare Timer 2 period value, low byte** The CP2L register holds the lower 8 bits of the 10-bit period value for Compare Timer 2 (shadow latch).
CP2H.1 - 0	**Compare Timer 2 period value, high bits** The CP2H register holds most significant two bits of the 10-bit period value for Compare Timer 2 (shadow latch).
–	Reserved bits

Compare Timer 2 Compare Registers

The compare registers CMP2H/CMP2L of Compare Timer 2 hold the 10-bit compare value which defines the duty cycle of the output signal at COUT3. When the Compare Timer 2 value is equal to the value stored in the CMP2H/CMP2L register, the COUT3 signal changes from passive to active state. If CMP2H/CMP2L is written, only shadow latches are written. The content of these latches is transferred to the real registers when compare timer count value 000_H is reached and bit STE2 of SFR CT2CON has been set.

When the compare registers CMP2H/CMP2L are read, the shadow registers are always accessed.

Special Function Registers CMP2L (Address $D4_H$) Reset Value: 00_H
Special Function Registers CMP2H (Address $D5_H$) Reset Value: $XXXXXX00_B$

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$D4_H$	.7	.6	.5	.4	.3	.2	.1	.0	CMP2L
$D5_H$	–	–	–	–	–	–	.1	.0	CMP2H

Bit	Function
CMP2L.7 - 0	**Compare Timer 2 compare value, low byte** The CMP2L register holds the lower 8 bits of the 10-bit compare value for Compare Timer 2.
CMP2H.1 - 0	**Compare Timer 2 compare value, high bits** The CMP2H register holds most significant two bits of the 10-bit Compare value for Compare Timer 2.
–	Reserved bits

6.3.4 Combined Multi-Channel PWM Modes

The CCU of the C508 has been designed to support also motor control or inverter applications which have a demand for specific multi-channel PWM signal generation. In these combined multi-channel PWM modes the CAPCOM unit (Compare Timer 1) and the COMP unit (Compare Timer 2) of the C508 CCU are working together.

In the combined multi-channel PWM modes the signal generation of the CCx and COUTx PWM outputs can basically be controlled either by the interrupt inputs $\overline{\text{INT0}}$ to $\overline{\text{INT2}}$ (block commutation mode) or by the operation of Compare Timer 1 or by software (multi-channel PWM mode). In the active phase of a combined multi-channel PWM mode, Compare Timer 1 compare output signal or the Compare Timer 2 output signal or both can be switched selectively to the CCx or COUTx PWM output lines.

The combined multi-channel PWM modes are controlled by the BCON (block commutation control) register. **Figure 6-33** shows the block diagram of the multi-channel PWM mode logic which is integrated in the C508.

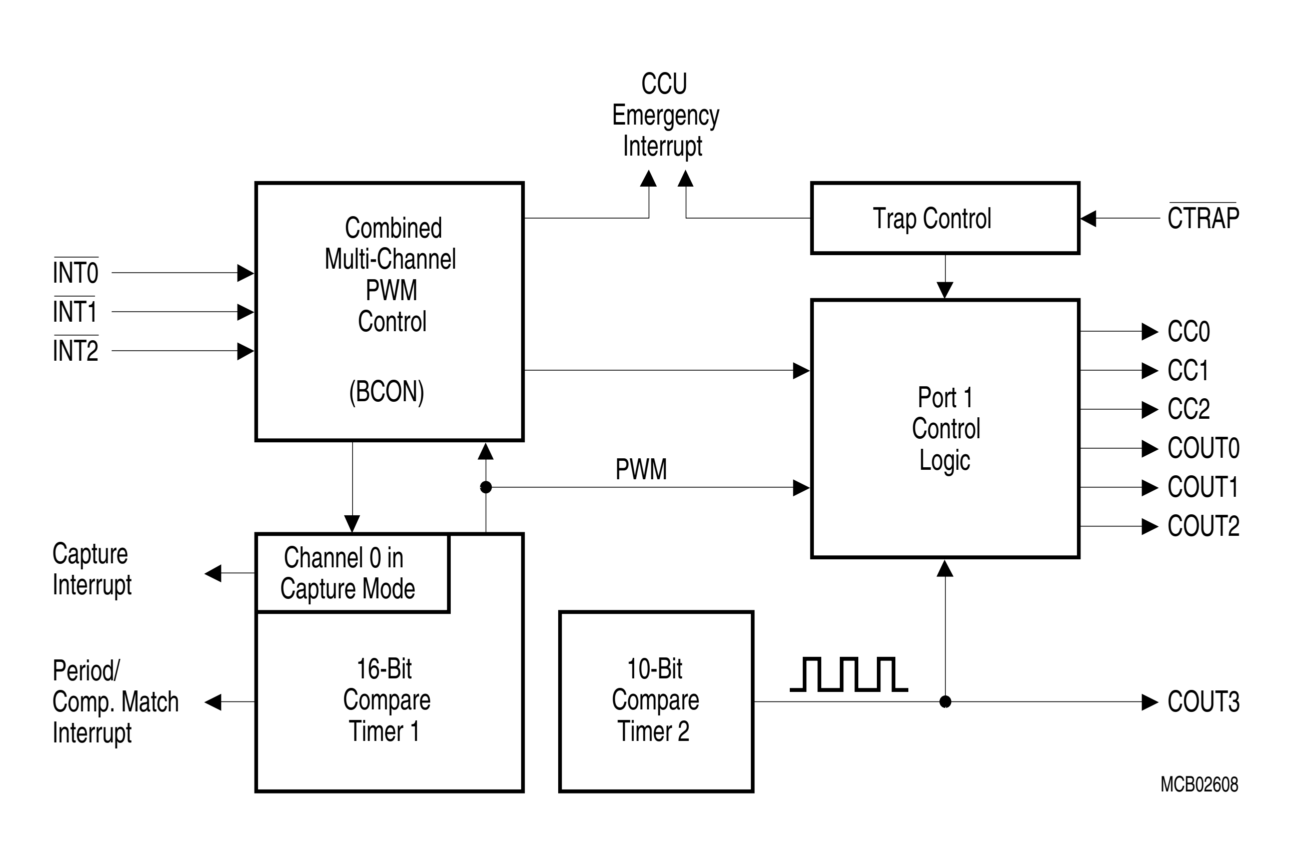


Figure 6-33 Block Diagram of the Combined Multi-Channel PWM Modes

In block commutation mode, a well defined incoming digital signal pattern of e.g. hall sensor signals, which are applied to the $\overline{\text{INT0-2}}$ inputs, is sampled. Each transition at the $\overline{\text{INT0-2}}$ inputs results in a change of the state of the PWM outputs. In block commutation mode, all six PWM output signals CCx and COUTx (x = 0-2) are outputs. According to a block commutation table (**Table 6-10**), the outputs CCx are put either to a low or high state while the outputs COUTx are switched to the PWM signal which is generated by the 10-bit Compare Timer 2 (COMP unit).

For monitoring of sensor input signal timing in block commutation mode, the signal transitions at $\overline{\text{INT0-2}}$ can also generate an interrupt (if enabled) and a capture event at channel 0 of the CAPCOM unit (Compare Timer 1). For emergency cases, (trap function of $\overline{\text{CTRAP}}$ input signal) the six outputs CCx and COUTx can be put selectively to the levels as defined by the first six bits in COTRAP register.

At the multi-channel PWM modes of the C508, a change of the PWM output states (active or inactive) is triggered by Compare Timer 1, which is running either in operating mode 0 or 1. If its count value reaches 0000_H, the PWM output signal changes its state according to a well defined state table. The multi-channel PWM modes consists of three modes:

– 4-phase multi-channel PWM mode (4 PWM output signals)
– 5-phase multi-channel PWM mode (5 PWM output signals)
– 6-phase multi-channel PWM mode (6 PWM output signals)

6.3.4.1 Control Register BCON

The BCON register controls the selection of multi-channel PWM modes. It also contains the block commutation interrupt enable and status bit/flag.

Special Function Register BCON (Address D7$_H$) **Reset Value: 00$_H$**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
D7$_H$	BCMP BCEM	PWM1	PWM0	EBCE	BCERR	BCEN	BCM1	BCM0	BCON

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>BCMP

BCEM</td><td><b>In multi-channel PWM mode: Machine polarity</b>
If BCMP is set and multi-channel PWM mode is selected (PWM1, 0 ≠ 0, 0), all enabled compare outputs COUTx and CCx are switched to the Compare Timer 2 output signal during their active phase. If BCMP is cleared, only the COUTx outputs are switched to the Compare Timer 2 output signal during the active phase in multi-channel PWM mode. CMSELx3 must be set for that functionality.
<b>In block commutation mode: Error mode select bit</b>
If BCEM is set in block commutation mode, in rotate right or rotate left mode additionally a “wrong follower” condition causes the setting of BCERR if EBCE is set.</td></tr>
<tr><td>PWM1
PWM0</td><td><b>Multi-channel PWM mode selection</b>
These bits select the operating mode of the multi-channel PWM modes.
<table>
<tr><th>PWM1</th><th>PWM0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Block commutation mode (for hall sensor inputs)</td></tr>
<tr><td>0</td><td>1</td><td>4-phase multi-channel PWM mode</td></tr>
<tr><td>1</td><td>0</td><td>5-phase multi-channel PWM mode</td></tr>
<tr><td>1</td><td>1</td><td>6-phase multi-channel PWM mode</td></tr>
</table></td></tr>
<tr><td>EBCE</td><td><b>Enable interrupt of block commutation mode error</b>
If EBCE is set, the emergency interrupt for a block commutation mode error condition of the CCU is enabled. In block commutation mode, an emergency error condition occurs if a false signal state at $\overline{\text{INT2}}$ - $\overline{\text{INT0}}$ or a wrong follower state (if selected by bit BCEM) is detected (see also <b>Table 6-10</b>).</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>BCERR</td><td><b>Block commutation mode error flag</b>
In block commutation mode BCERR is set in rotate right or rotate left mode if after a transition at $\overline{\text{INTx}}$ all $\overline{\text{INTx}}$ inputs are at high or low level. Additionally, in rotate right or rotate left mode a “wrong follower” condition according to <b>Table 6-10</b> can cause the setting of BCERR (see description of bit BCEM).
If the block commutation interrupt is enabled (EBCE = 1), the setting of BCERR will generate a CCU emergency interrupt. BCERR must be reset by software.</td></tr>
<tr><td>BCEN</td><td><b>Block commutation enable</b>
If BCEN is set, the multi-channel PWM modes of the CAPCOM unit as selected by the bits PWM1/PWM0 are enabled for operation. Before BCEN bit is set, all required PWM Compare outputs should be programmed to operate as compare outputs by writing the registers CMSEL1/CMSEL0.</td></tr>
<tr><td>BCM1
BCM0</td><td><b>Multi-channel PWM mode output pattern selection</b>
Additionally to bits PWM1 and PWM0, these two control bits select the output signal pattern in all multi-channel PWM modes. The detailed signal pattern information is given in <b>Table 6-10</b> to <b>Table 6-13</b>.
<table>
<tr><th>BCM1</th><th>BCM0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Idle mode</td></tr>
<tr><td>0</td><td>1</td><td>Rotate right mode</td></tr>
<tr><td>1</td><td>0</td><td>Rotate left mode</td></tr>
<tr><td>1</td><td>1</td><td>Slow down mode</td></tr>
</table></td></tr>
</table>

Note: When a multi-channel PWM mode is initiated the first time after reset, BCON must be written twice: first write operation with bit BCEN cleared and all other bits set/ cleared as required (BCM1, 0 must be 0, 0 for idle mode), followed by a second write operation with the same BCON bit pattern of the first write operation ***but with BCEN set.*** *After this second BCON write operation, Compare Timer 1 can be started (setting CT1R in CT1CON) and thereafter BCM1, 0 can be put into another mode than idle mode.*

6.3.4.2 Signal Generation in Multi-Channel PWM Modes

The multi-channel PWM modes of the C508 use the pins CCx and COUTx for compare output signal generation. Before signal generation of a multi-channel PWM mode can be started, the COINI register should be programmed with the logic value of the multi-channel PWM inactive phase. After this, the output pins which are required for the multi-channel PWM signal generation must be programmed to operate as compare outputs by writing the mode select registers CMSEL0 and CMSEL1. **Table 6-9** shows the CMSEL0/ CMSEL1 register bits which are required for the full operation of the multi-channel PWM modes.

Table 6-9 Programming of Multi-Channel PWM Compare Outputs

Multi-Channel PWM Mode	CMSEL1	CMSEL0
Block commutation / 6-phase multi-channel PWM	XXXX Y011_B	Y011 Y011_B
5-phase multi-channel PWM		Y010 Y011_B
4-phase multi-channel PWM		Y010 Y001_B

Note: The abbreviation "X" means don't care. The abbreviation "Y" (bit CMSELx.3) represents the burst mode bit. If Y = 0 the signal generation at the COUTx pins is controlled by Compare Timer 1. If Y = 1 the signal generation at the COUTx pins is also controlled by Compare Timer 1 but modulated by Compare Timer 2.

Output Signals During the Active Phase

An active phase of a compare output signal in multi-channel PWM mode can be controlled either by the CAPCOM unit (Compare Timer 1) and/or modulated by Compare Timer 2. The selection is done by bit CMSELx.3 (see note below **Table 6-9**).

Figure 6-34 shows the different possibilities for controlling the active phase of a compare output signal using Compare Timer 1. Compare Timer 1 may operate either in mode 0 or mode 1. In multi-phase mode, the block commutation logic switches from one state to the next state when Compare Timer 1 reaches the value 0000_H. As an active phase always lasts for two states, the duration of an active phase is determined by Compare Timer 1 reaching 0000_H twice.

As shown in **Figure 6-34a**, a compare output signal CCx or COUTx of a CAPCOM channel is either at low or high level during the whole active phase when the value stored in the Compare Timer 1 offset registers (CT1OFH, CT1OFL) and the value stored in its compare registers (CCHx, CCLx) is equal 0000_H. When the compare value is not equal 0000_H and less or equal the period value, the active phase of the related compare output signal CCx or COUTx is controlled by the CAPCOM unit as shown in **Figure 6-34b**.

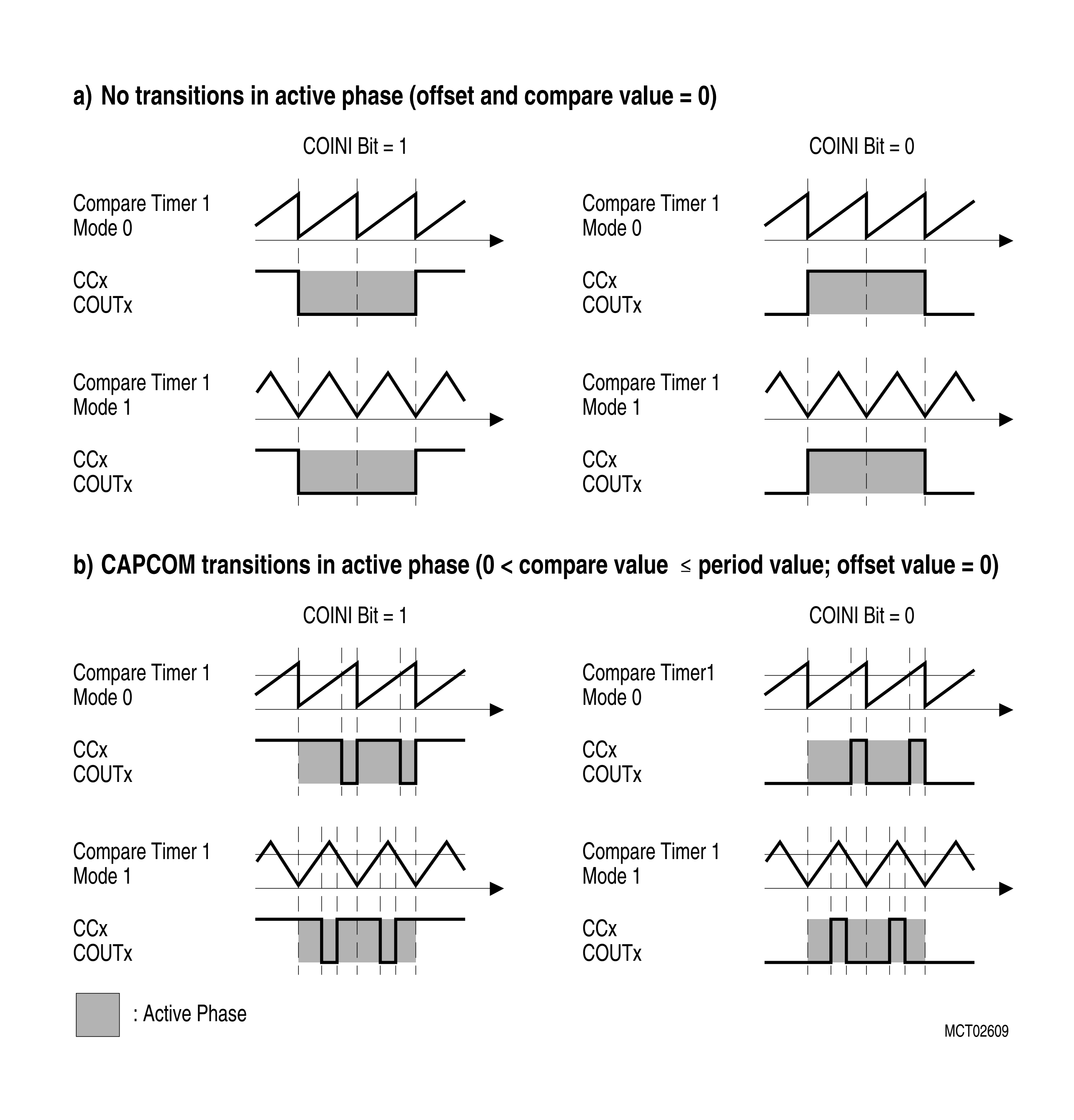


Figure 6-34 Compare Timer 1 Controlled Active Phase of the Multi-Channel PWM Modes (with CMSELx.3 = 0)

Figure 6-35 shows the different possibilities for controlling the active phase of a compare output signal using Compare Timer 2. In this operating mode, which is selected when bit CMSELx.3 is set, the Compare Timer 2 output signal is switched to the COUTx or CCx outputs during the active phase of a multi-channel PWM signal. Bit BCMP (BCON.7) defines whether only COUTx or COUTx and CCx are modulated by the Compare Timer 2 output signal.

Depending on the bits COUT3I and COUTXI of COINI, the polarity of COUT3 and the switched CCx/COUTx active phase signal can be identical or inverted.

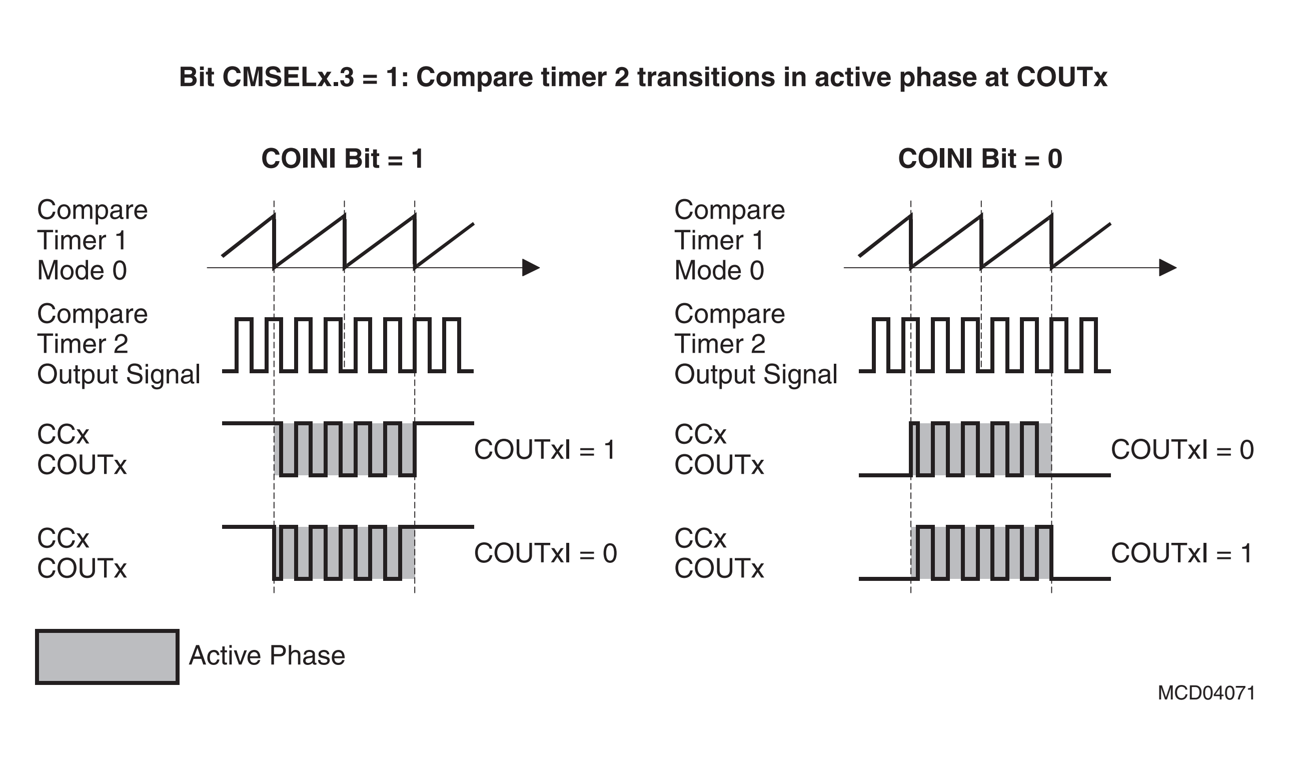


Figure 6-35 Compare Timer 2 Controlled Active Phase of the Multi-Channel PWM Modes (with CMSELx.3 = 1)

6.3.4.3 Block Commutation PWM Mode

In block commutation mode the $\overline{\text{INT0-2}}$ inputs are sampled once each processor cycle. If the input signal combination at $\overline{\text{INT0-2}}$ changes its state, the outputs CCx and COUTx are set to their new state according to **Table 6-10**.

Table 6-10 Block Commutation Control Table

Mode (BCM1, BCM0)	$\overline{\text{INT0}}$ - $\overline{\text{INT2}}$ Inputs			CC0 - CC2 Outputs			COUT0 - COUT2 Outputs		
	$\overline{\text{INT0}}$	$\overline{\text{INT1}}$	$\overline{\text{INT2}}$	CC0	CC1	CC2	COUT0	COUT1	COUT2
Rotate left[1]	0	0	0	inactive	inactive	inactive	inactive	inactive	inactive
Rotate right[1]	1	1	1	inactive	inactive	inactive	inactive	inactive	inactive
Rotate left, 60° phase shift (BCTSEL = 0, default)	1	0	1	inactive	inactive	active	active	inactive	inactive
	1	0	0	inactive	active	inactive	active	inactive	inactive
	1	1	0	inactive	active	inactive	inactive	inactive	active
	0	1	0	active	inactive	inactive	inactive	inactive	active
	0	1	1	active	inactive	inactive	inactive	active	inactive
	0	0	1	inactive	inactive	active	inactive	active	inactive
Rotate left, 0° phase shift (BCTSEL - 1)	1	0	1	inactive	inactive	active	inactive	active	inactive
	1	0	0	inactive	inactive	active	active	inactive	inactive
	1	1	0	inactive	active	inactive	active	inactive	inactive
	0	1	0	inactive	active	inactive	inactive	inactive	active
	0	1	1	active	inactive	inactive	inactive	inactive	active
	0	0	1	active	inactive	inactive	inactive	active	inactive
Rotate right	1	1	0	active	inactive	inactive	inactive	active	inactive
	1	0	0	active	inactive	inactive	inactive	inactive	active
	1	0	1	inactive	active	inactive	inactive	inactive	active
	0	0	1	inactive	active	inactive	active	inactive	inactive
	0	1	1	inactive	inactive	active	active	inactive	inactive
	0	1	0	inactive	inactive	active	inactive	active	inactive
Slow down	X	X	X	inactive	inactive	inactive	active	active	active
Idle[2]	X	X	X	inactive	inactive	inactive	inactive	inactive	inactive

[1] If one of these two combinations of $\overline{\text{INTx}}$ signals is detected in rotate left or rotate right mode, bit BCERR flag is set. If enabled, a CCU emergency interrupt can be generated. When these states (error states) are reached, idle state is entered immediately.

[2] Idle state is also entered when a “wrong follower” is detected (if bit BCON.7 = BCEM is set). When idle state is entered, the BCERR flag is always set. Idle state can only be left when the BCERR flag is reset by software.

Two tables are available for “rotate left” direction. The first table is identical to the one in C504 which has a 60° phase shift. It is selected if bit BCTSEL of SFR COTRAP is cleared. The second table has 0° phase shift, and it is selected if bit BCTSEL is set. After reset, the first table is selected by default.

This option is provided as a feature so that a wider range of motors can be operated at optimum performance.

In block commutation mode, any signal transition at $\overline{\text{INT0-2}}$ generates a capture pulse for CAPCOM channel 0 (CCH0/CCL0), independently from the selected $\overline{\text{INT0-2}}$ signal transition type as defined in the SFR TCON (for $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$) and SFR T2CON (for $\overline{\text{INT2}}$).

Figure 6-36 gives an example of a block commutation mode timing (only COUTx outputs are modulated with Compare Timer 2 output signal). It shows the case for rotate left at 60° phase shift (BCM1, BCM0 = 1, 0; BCTSEL = 0) and the rotate right case (BCM1, BCM0 = 0, 1). For the timing shown in Figure 6-36 the COINI register is set to XX111111$_B$. This means that a high level is defined as inactive phase. The CMSELx.3 bits in the CMSEL0/CMSEL1 registers must also be set (Compare Timer 2 switched to COUTx during active phase). The timing shown below is directly derived from Table 6-10.

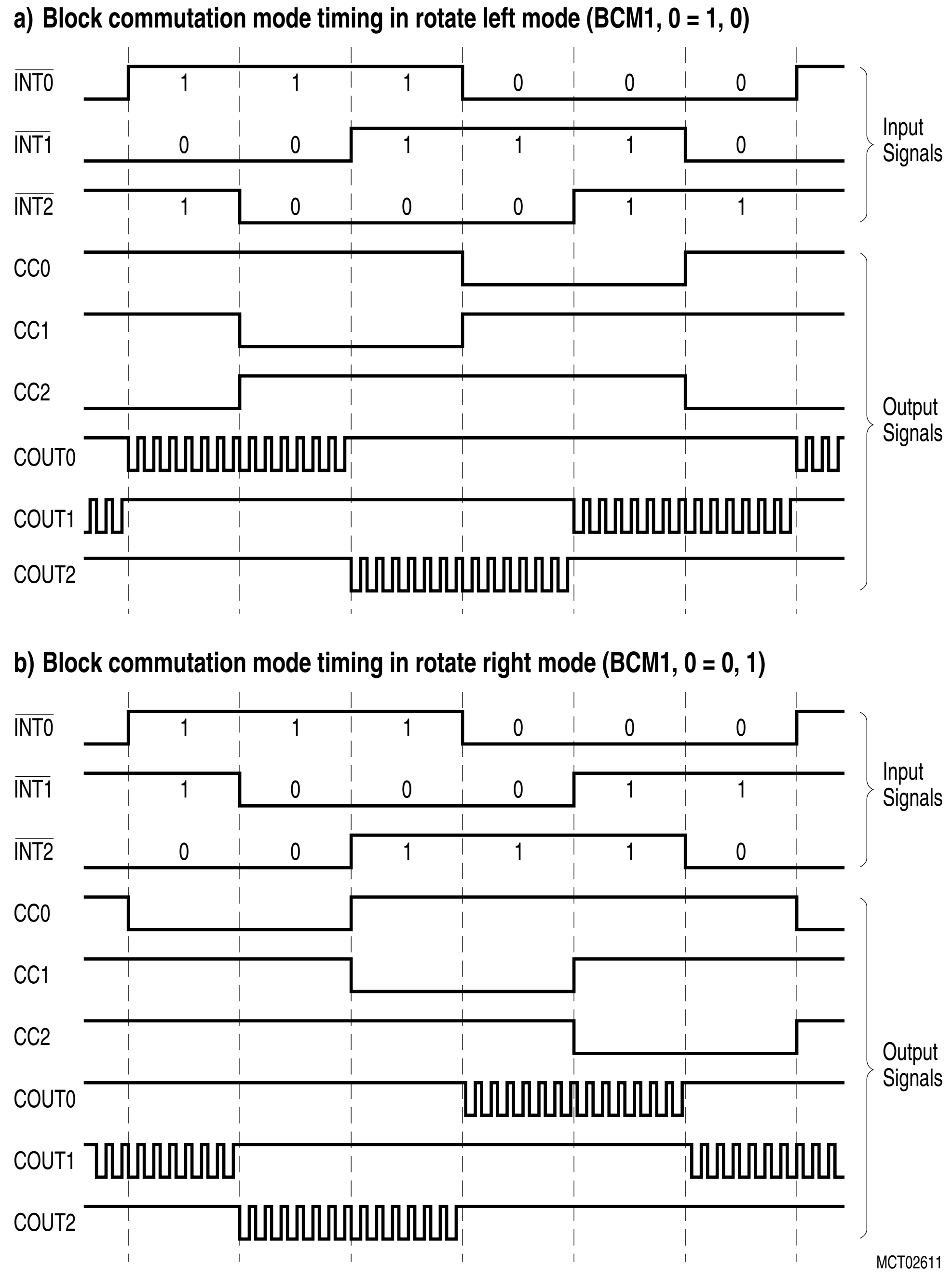


Figure 6-36 Block Commutation Mode Timing

6.3.4.4 Compare Timer 1 Controlled Multi-Channel PWM Modes

Using the multi-channel PWM modes of the C508, several Compare Timer 1 controlled PWM waveforms can be generated:

- 4-phase multi-channel PWM waveforms
- 5-phase multi-channel PWM waveforms
- 6-phase multi-channel PWM waveforms

The basic waveforms of these three Compare Timer 1 controlled PWM modes are shown in the following three figures, **Figure 6-37** to **Figure 6-39**. The figures show waveforms for different COINI values with the resulting active/inactive phases and rotate right / rotate left condition. All three figures assume that Compare Timer 1 operates with 100% duty cycle (compare and offset registers = 0000_H) and without Compare Timer 2 modulation. Compare Timer 1 duty cycles less than 100% or Compare timer 2 modulation in the multi-channel PWM modes are shown in **Figure 6-34** and **Figure 6-35**.

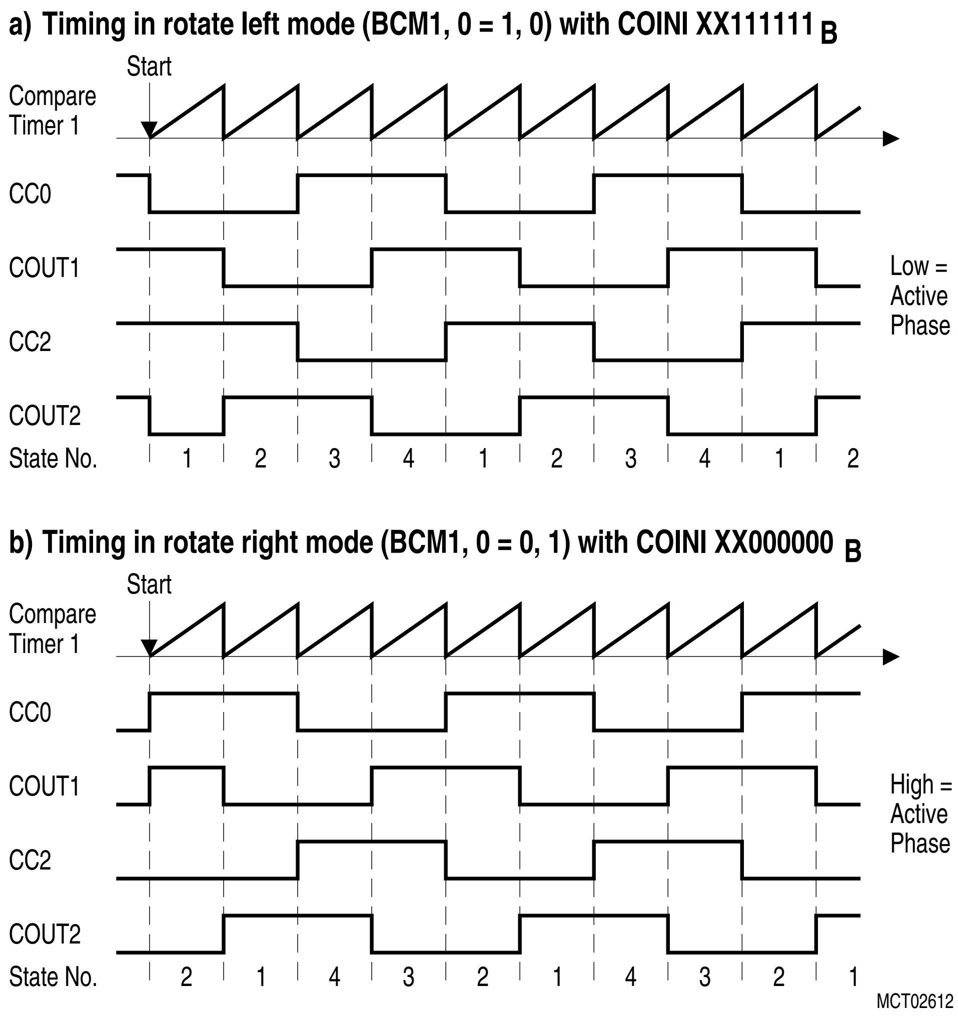


Figure 6-37 Basic Compare Timer 1 Controlled 4-Phase PWM Timing

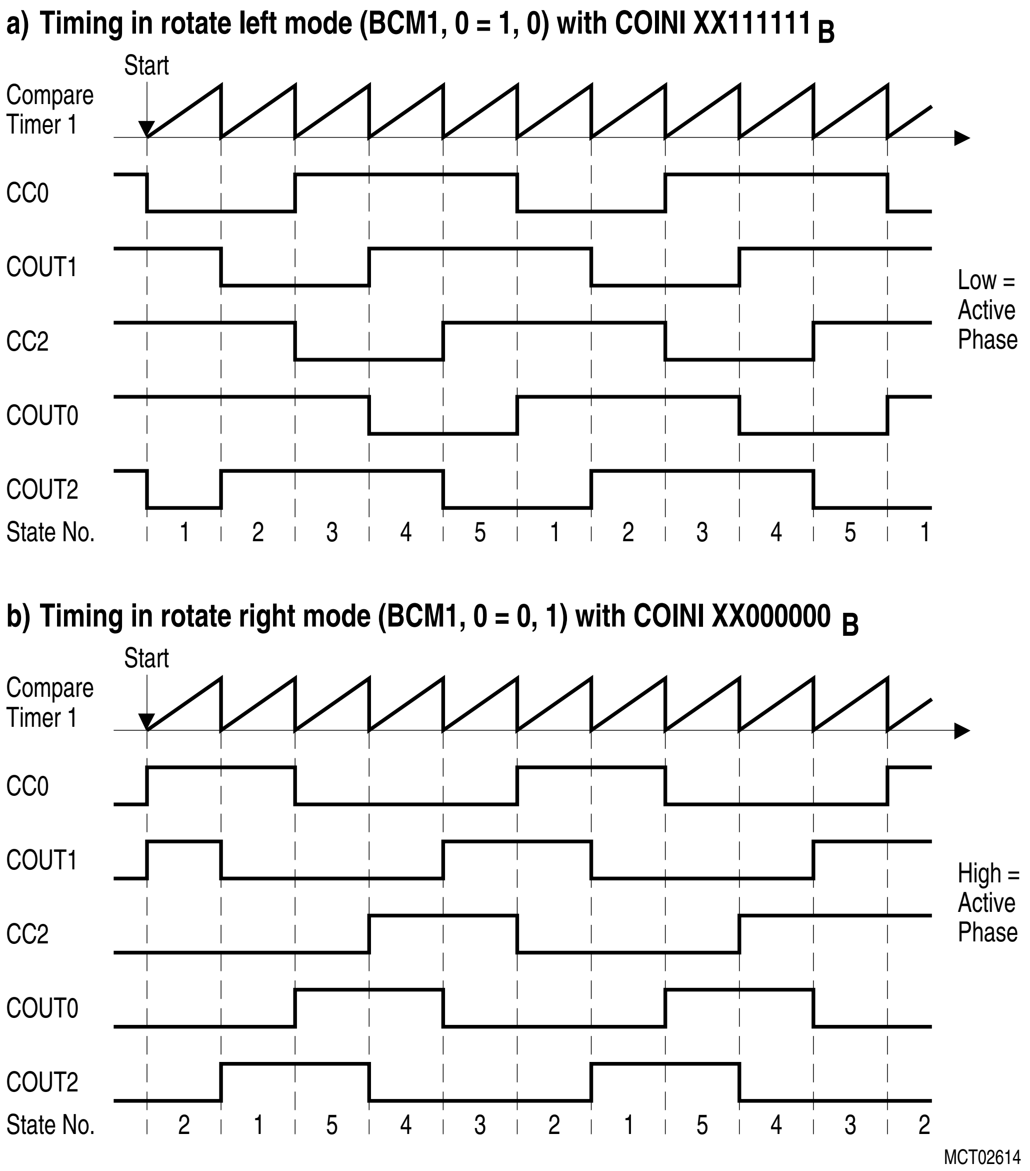


Figure 6-38 Basic Compare Timer 1 Controlled 5-Phase PWM Timing

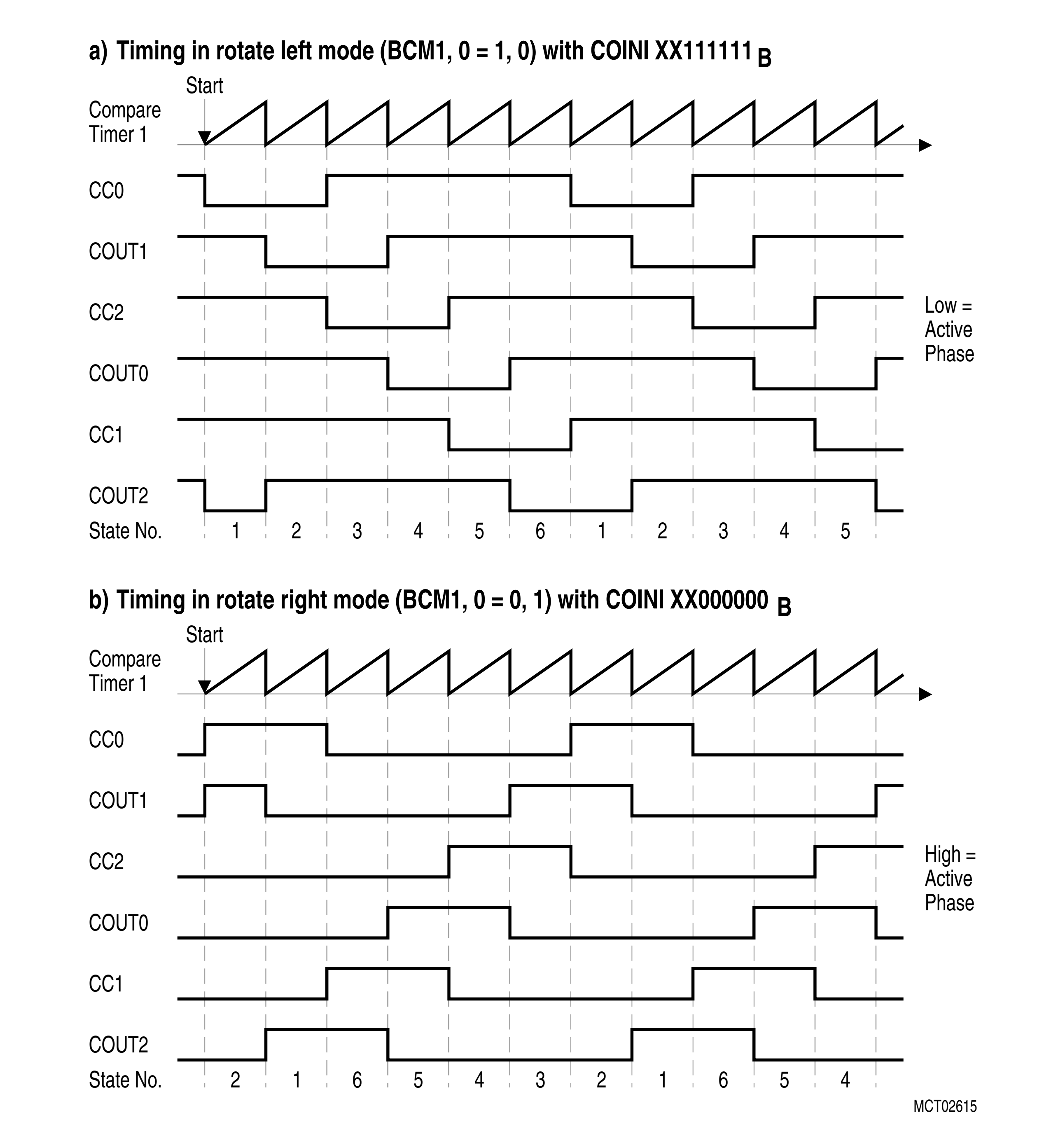


Figure 6-39 Basic Compare Timer 1 Controlled 6-Phase PWM Timing

Table 6-11 to Table 6-13 show the basic signal pattern definitions of the three multi-channel PWM modes. They also include information about slow down mode and idle mode (bits BMC1, 0 = 0, 0 and 1, 1).

Table 6-11 4-Phase PWM Timing State Table

Actual State and PWM Phase					Follower State (No.)			
No.	**Output Signals**				**BCM1, BCM0 =**			
	CC0	**COUT1**	**CC2**	**COUT2**	**0, 1**	**1, 0**	**0, 0**	**1, 1**
0	inactive	inactive	inactive	inactive	2	1	0	5
1	active	inactive	inactive	active	4	2	0	5
2	active	active	inactive	inactive	1	3	0	5
3	inactive	active	active	inactive	2	4	0	5
4	inactive	inactive	active	active	3	1	0	5
5	inactive	active	inactive	active	2	1	0	5

Note: In the inactive phase the PWM outputs drive a logic state as defined by the related bits in register COINI. During the active phase, the PWM outputs can be modulated by CT1 and/or CT2.

Table 6-12 5-Phase PWM Timing State Table

Actual State and PWM Phase						Follower State (No.)			
No.	**Output Signals**					**BCM1, BCM0 =**			
	CC0	**COUT1**	**CC2**	**COUT0**	**COUT2**	**0, 1**	**1, 0**	**0, 0**	**1, 1**
0	inactive	inactive	inactive	inactive	inactive	2	1	0	6
1	active	inactive	inactive	inactive	active	5	2	0	6
2	active	active	inactive	inactive	inactive	1	3	0	6
3	inactive	active	active	inactive	inactive	2	4	0	6
4	inactive	inactive	active	active	inactive	3	5	0	6
5	inactive	inactive	inactive	active	active	4	1	0	6
6	inactive	active	inactive	active	active	2	1	0	6

Note: In the inactive phase, the PWM outputs drive a logic state as defined by the related bits in register COINI. During the active phase, the PWM outputs can be modulated by CT1 and/or CT2.

Table 6-13 6-Phase PWM Timing State Table

Actual State and PWM Phase							Follower State (No.)			
No.	**Output Signals**						**BCM1, BCM0 =**			
	CC0	**COUT1**	**CC2**	**COUT0**	**CC1**	**COUT2**	**0, 1**	**1, 0**	**0, 0**	**1, 1**
0	inactive	inactive	inactive	inactive	inactive	inactive	2	1	0	7
1	active	active	inactive	inactive	inactive	inactive	5	2	0	7
2	inactive	active	active	inactive	inactive	inactive	1	3	0	7
3	inactive	inactive	active	active	inactive	inactive	2	4	0	7
4	inactive	inactive	inactive	active	active	inactive	3	5	0	7
5	inactive	inactive	inactive	inactive	active	active	4	6	0	7
6	active	inactive	inactive	inactive	inactive	active	5	1	0	7
7	inactive	active	inactive	active	inactive	active	2	1	0	7

Note: In the inactive phase, the PWM outputs drive a logic state as defined by the related bits in register COINI. During the active phase, the PWM outputs can be modulated by CT1 and/or CT2.

6.3.4.5 Software Controlled State Switching in Multi-Channel PWM Modes

In the 4-/5-/6-phase multi-channel PWM modes, the Compare Timer 1 overflow controlled switching of the follower state can be switched off. Instead of the Compare Timer 1 overflow, a setting of bit NMCS in SFR CMSEL1 selects the follower state, which is defined in the **Table 6-11** to **Table 6-13**. Bit ESMC in SFR CMSEL1 enables the software controlled state switching.

If this software controlled 4-/5-/6-phase multi-channel PWM mode generation is selected, the Compare Timer 1 can be used for PWM signal generation (compare mode) in order to modulate the outputs. It can be further used, for example, for timer based interrupt generation. The waveforms of a PWM output signal in the multi-channel PWM modes can be selected as shown in **Figure 6-34** (static low or high during active phase) or as shown in **Figure 6-35** (Compare Timer 2 controlled modulation during active phase).

Figure 6-40 shows for the 5-pole PWM timing the possible waveforms of the active phase when the software controlled state switching in the multi-channel PWM modes is selected.

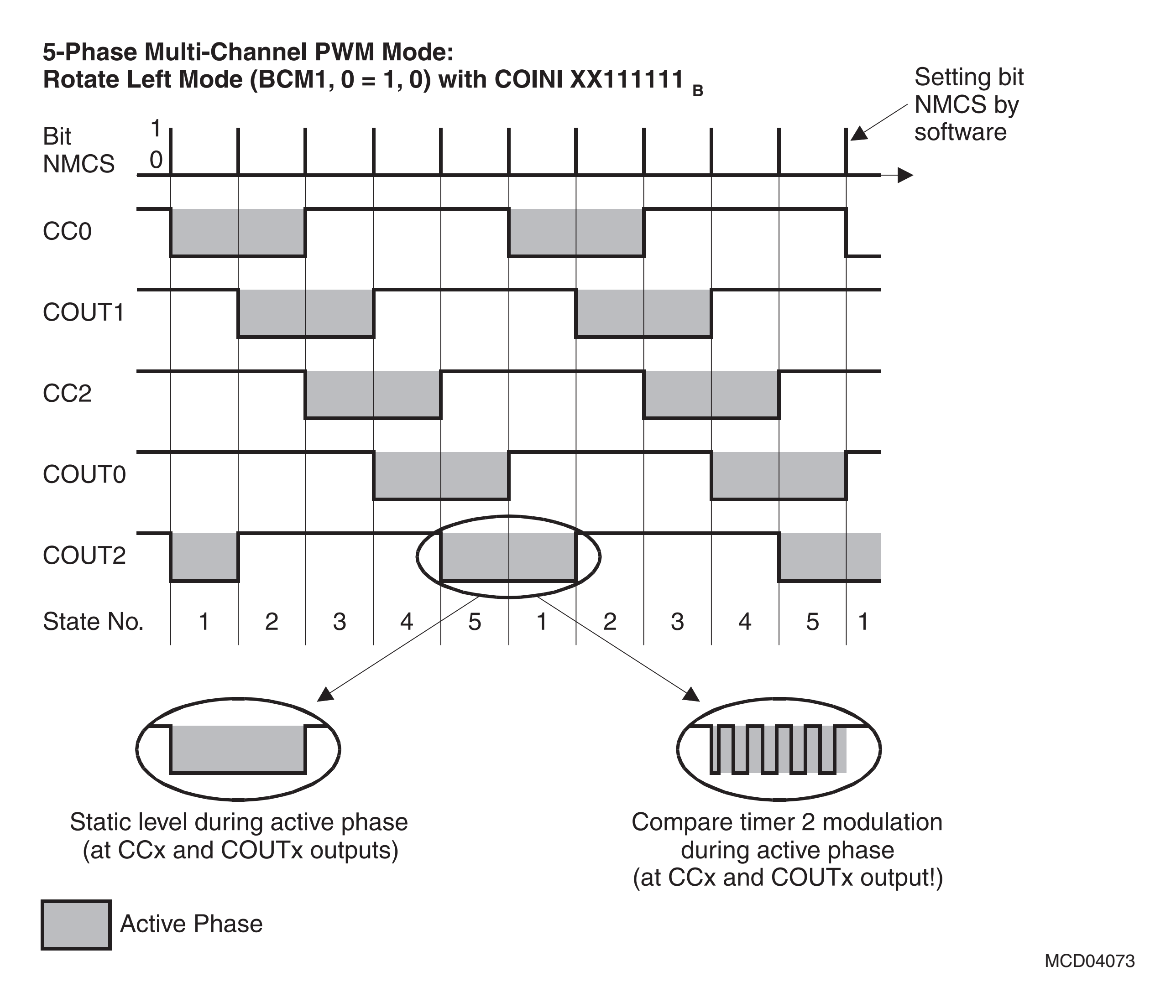

5-Phase Multi-Channel PWM Mode:
Rotate Left Mode (BCM1, 0 = 1, 0) with COINI XX111111 B
Setting bit NMCS by software
Bit NMCS
1
0
CC0
COUT1
CC2
COUT0
COUT2
State No.
1
2
3
4
5
1
2
3
4
5
1
Static level during active phase (at CCx and COUTx outputs)
Compare timer 2 modulation during active phase (at CCx and COUTx output!)
Active Phase
MCD04073

Figure 6-40 Software Controlled State Switching in 5-Phase Multi-Channel PWM Mode

Static Level during Active Phase:

When bit ESMC in SFR CMSEL1 is set, static active or passive output levels during the active phase of a multi-phase PWM timing are generated when the following conditions are met:

– The 16-bit offset register of Compare Timer 1 must be 0000_H (CT1OFH = CT1OFL = 00_H)
– static active: compare values = 0000_H
  static passive: compare values > period value
– The bits CMSELx3 (x = 0-2) in the SFRs CMSEL0/CMSEL1 must be 0.

The logic state of the inactive/active phases at the CCx and COUTx outputs is defined by the bits in SFR COINI.

Compare Timer 2 Controlled Active Phase at COUTx:

When bit ESMC in SFR CMSEL1 is set, Compare Timer 2 controlled output levels at COUTx during the active phase of a multi-pole PWM timing are generated when the following conditions are met:

– The 16-bit offset register of Compare Timer 1 must be 0000_H (CT1OFH = CT1OFL = 00_H)
– The 16-bit capture/compare registers must be 0000_H (CCL0 = CCH0 = CCL1 = CCH1 = CCL2 = CCH2 = 00_H)
– Bits CMSELx3 (x = 0-2) in the SFRs CMSEL0/CMSEL1 must be set
– Compare Timer 2 must be enabled and initialized for compare output signal generation

Both, the CCx and the COUTx outputs can be controlled by Compare Timer 2.

A combination of outputs modulated by Compare Timer 1 and/or Compare Timer 2 is supported.

6.3.4.6 Trap Function in Multi-Channel Block Commutation Mode

The trap function in block commutation mode is similar to the trap function described in **Chapter 6.3.2.7**, **"Trap Function of the CAPCOM Unit in Compare Mode"**. But there is one difference: when $\overline{\text{CTRAP}}$ becomes inactive (high), the CCx and COUTx outputs are again switched back to the PWM pulse generation when Compare Timer 2 reaches the count value 000_H (instead of Compare Timer 1 in all other modes).

All other trap functions of the multi-channel PWM modes are identical as described in **Chapter 6.3.2.7**.

6.4 Serial Interface

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

The serial port can operate in 4 modes (one synchronous mode, three asynchronous modes). The baudrate clock for the serial port is derived from the oscillator frequency (Modes 0 and 2) or generated either by Timer 1 or a dedicated baudrate generator (Modes 1 and 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 with the Least Significant Bit (LSB) first. The baudrate is fixed at 1/3 of the oscillator frequency (see **Chapter 6.4.4** 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 special function register SCON. The baudrate in Mode 1 is variable (see **Chapter 6.4.5** 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’. Alternatively, the parity bit (P, in the PSW) could be moved into TB8 an example. 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 1/8 or 1/16 of the oscillator frequency (see **Chapter 6.4.6** 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). In fact, Mode 3 is the same as Mode 2 in all respects except the baudrate. The baudrate in Mode 3 is variable (see **Chapter 6.4.6** 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 a frame is completed. The corresponding interrupt request flags are TI or RI, respectively. See **Chapter 7** for more details about the interrupt structure. The interrupt request flags, TI and RI, can also be used for polling the serial interface, if the serial interrupt is not to be used (that is, serial interrupt is not enabled).

6.4.1 Multiprocessor Communication

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

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

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

6.4.2 Serial Port Registers

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

SBUF is the receive and transmit buffer of 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 No.	MSB							LSB	
	$9F_H$	$9E_H$	$9D_H$	$9C_H$	$9B_H$	$9A_H$	99_H	98_H	
98_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON
	7	6	5	4	3	2	1	0	
99_H	Serial Interface Buffer Register								SBUF

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>SM0
SM1</td><td><b>Serial Port 0 operating mode selection bits</b>
<table>
<tr><th>SM0</th><th>SM1</th><th>Selected operating mode</th></tr>
<tr><td>0</td><td>0</td><td>Serial mode 0: Shift register, fixed baudrate ($f_{OSC}/3$)</td></tr>
<tr><td>0</td><td>1</td><td>Serial mode 1: 8-bit USART, variable baudrate</td></tr>
<tr><td>1</td><td>0</td><td>Serial mode 2: 9-bit USART, fixed baudrate ($f_{OSC}/8$ or $f_{OSC}/16$)</td></tr>
<tr><td>1</td><td>1</td><td>Serial mode 3: 9-bit USART, variable baudrate</td></tr>
</table></td></tr>
<tr><td>SM2</td><td><b>Enable serial port multiprocessor communication in Modes 2 and 3</b>
In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit is not received. In Mode 0, SM2 should be 0.</td></tr>
<tr><td>REN</td><td><b>Enable receiver of serial port</b>
Set by software to enable serial reception. Cleared by software to disable serial reception.</td></tr>
<tr><td>TB8</td><td><b>Serial port transmitter bit 9</b>
TB8 is the 9th data bit that will be transmitted in Modes 2 and 3. Set or cleared by software as desired.</td></tr>
<tr><td>RB8</td><td><b>Serial port receiver bit 9</b>
In Modes 2 and 3, RB8 is the 9th data bit that is received. In Mode 1, if SM2 = 0, RB8 is the stop bit that is received. In Mode 0, RB8 is not used.</td></tr>
</table>

Bit	Function
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.

6.4.3 Baudrate Generation

There are several possibilities to generate the baudrate clock for the serial port, depending on the mode in which it is operating.

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

The serial interface requires a clock rate which is 16 times the baudrate for internal synchronization. Therefore, the baudrate generators must provide a “baudrate clock” to the serial interface which - there divided by 16 - results in the actual “baudrate”. However, all formulae given in the following section already include the factor and calculate the final baudrate.

The baudrate of the serial port is controlled by two bits which are located in the special function registers as 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 DF_H	DE_H	DD_H	DC_H	DB_H	DA_H	$D9_H$	LSB $D8_H$	
$D8_H$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	ADCON0

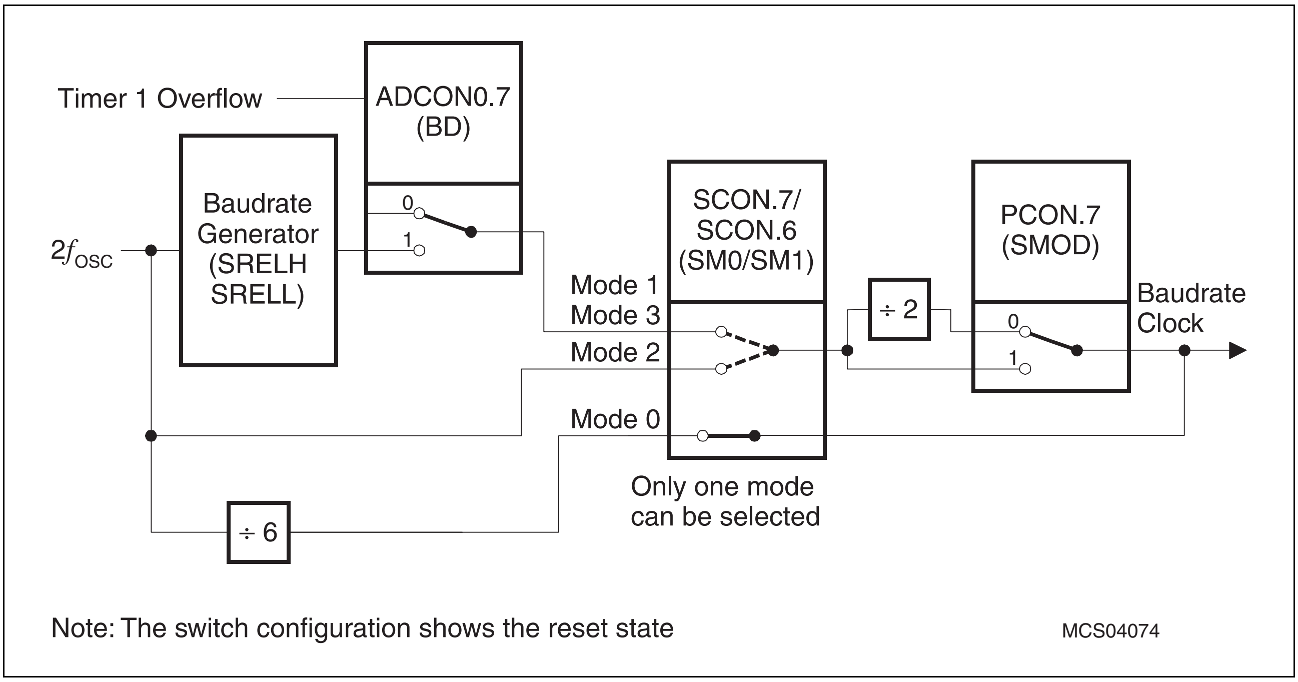
	7	6	5	4	3	2	1	0	
87_H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

▨ The shaded bits are not used for controlling the baudrate.

Bit	Function
BD	**Baudrate generator enable** When set, the baudrate of the 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.

Note: Bit CLK of SFR ADCON0 must be written with a ‘0’.

Figure 6-41 shows the configuration for the baudrate generation of the serial port.



Note: The switch configuration shows the reset state

Figure 6-41 Baudrate Generation for the Serial Port

Depending on the programmed operating mode different paths are selected for the baudrate clock generation. Figure 6-41 shows the dependencies of the serial port baudrate clock generation on the two control bits and from the mode which is selected in the special function register SCON.

6.4.3.1 Baudrate in Mode 0

The baudrate in Mode 0 is fixed to:

$$\text{Mode 0 baudrate} = \frac{\text{oscillator frequency}}{3}$$

6.4.3.2 Baudrate in Mode 2

The baudrate in Mode 2 depends on the value of bit SMOD in special function register PCON. If SMOD = 0 (which is the value after reset), the baudrate is 1/16 of the oscillator frequency. If SMOD = 1, the baudrate is 1/8 of the oscillator frequency.

$$\text{Mode 2 baudrate} = \frac{2^{\text{SMOD}}}{16} \times \text{oscillator frequency}$$

6.4.3.3 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.

6.4.3.3.1 Using the Internal Baudrate Generator

In Modes 1 and 3, the C508 can use an internal baudrate generator for the serial port. To enable this feature, bit BD (bit 7 of special function register 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 $2 \times f_{OSC}$ (output of PLL).

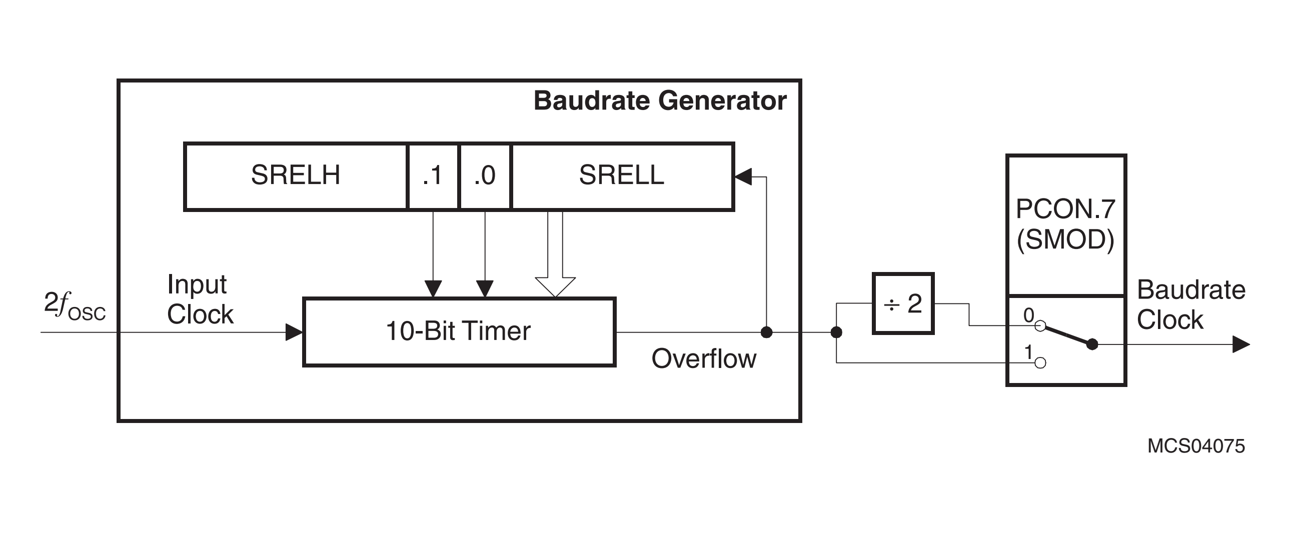


Figure 6-42 Serial Port Input Clock when using the Baudrate Generator

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 No.	MSB 7	6	5	4	3	2	1	LSB 0	
BA_H	–	–	–	–	–	–	MSB .9	.8	SRELH
AA_H	.7	.6	.5	.4	.3	.2	.1	LSB .0	SRELL

▨ The shaded bits are not used for reload operation.

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 10 MHz oscillator frequency, a reload value of $37E_H$ is required to achieve the commonly used baudrates of 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) at a deviation of 0.16%.

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

$$\text{Mode 1, 3 baudrate} = \frac{2^{\text{SMOD}} \times \text{oscillator frequency}}{16 \times (\text{baudrate generator overflow rate})}$$

$$\text{Baudrate generator overflow rate} = 2^{10} - \text{SREL}$$
$$\text{with SREL} = \text{SRELH.1 - 0, SRELL.7 - 0}$$

6.4.3.3.2 Using Timer 1 to Generate Baudrates

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

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

The Timer 1 interrupt should be disabled in this application. Timer 1, itself, can be configured for either “timer” or “counter” operation, and 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:

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

Very low baudrates can be achieved with Timer 1 by leaving the Timer 1 interrupt enabled, and 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.

6.4.4 Details about Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted/received with the LSB first. The baudrate is fixed at $f_{OSC}/3$.

Figure 6-43 shows a simplified functional diagram of the serial port in Mode 0. The associated timing is illustrated in **Figure 6-44**.

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

As data bits shift out to the right, '0's come in from the left. When the Most Significant Bit (MSB) of the data byte is at the output position of the shift register, the '1' 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 '0's. This condition flags the TX control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "Write-to-SBUF".

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

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

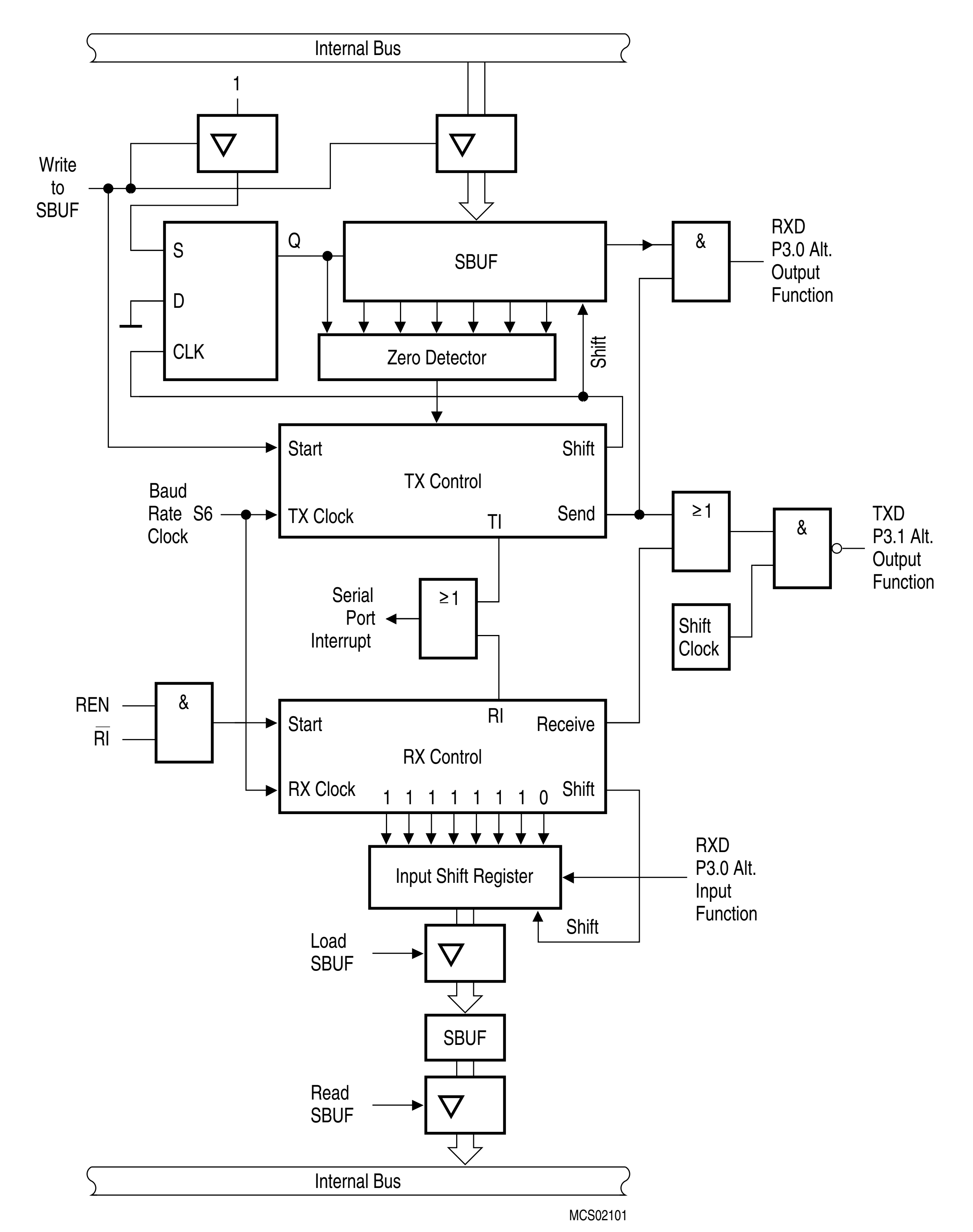


Figure 6-43 Serial Interface, Mode 0, Functional Diagram

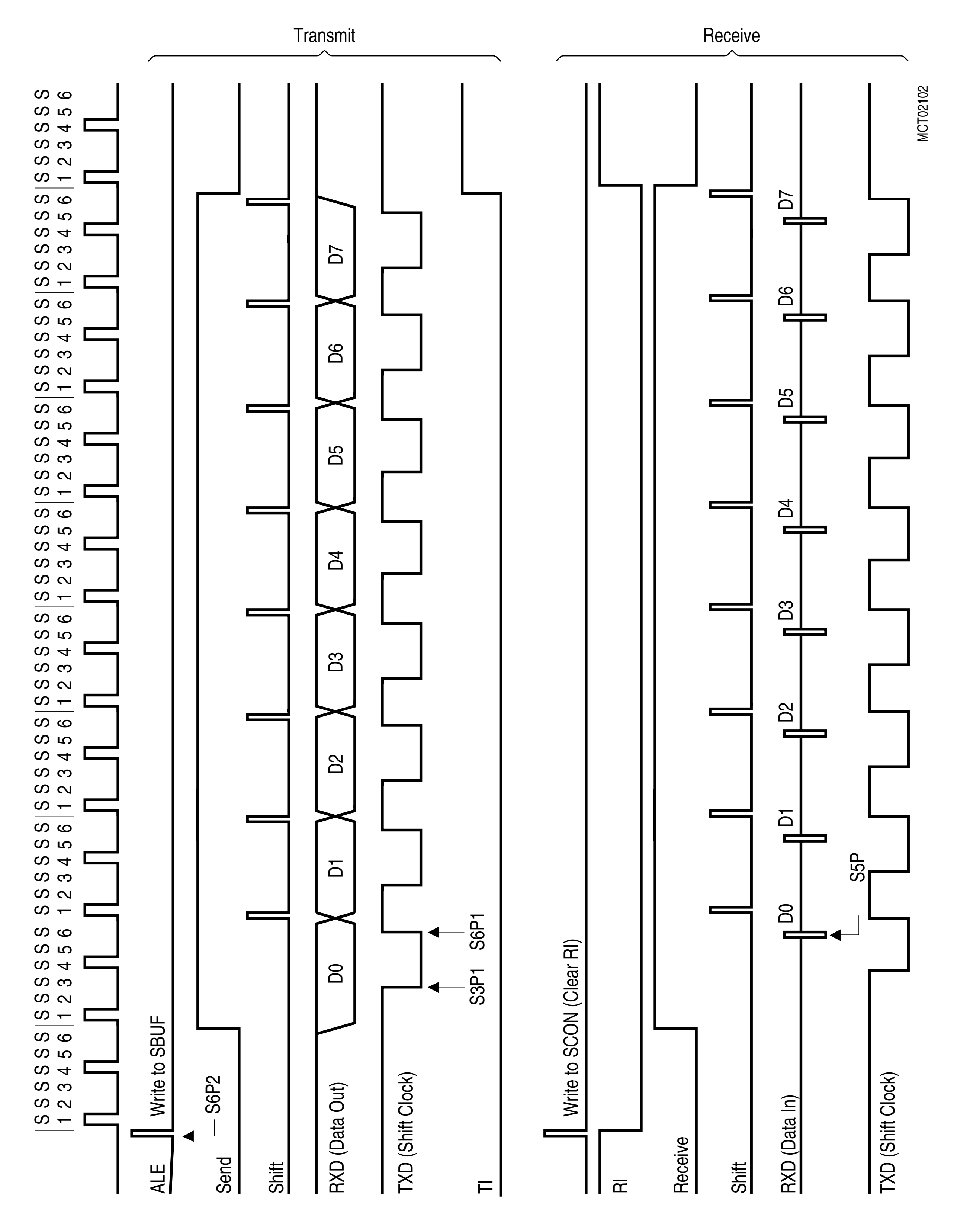


Figure 6-44 Serial Interface, Mode 0, Timing Diagram

6.4.5 Details about Mode 1

Ten bits are transmitted through TxD or received through RxD: a start bit (0), eight data bits (LSB first), and a stop bit (1). On 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 6-45 shows a simplified functional diagram of the serial port in Mode 1. Timing associated with transmit and receive is illustrated in **Figure 6-46**.

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

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

As data bits shift out to the right, '0's are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the '1' which was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain '0's. This condition flags the TX control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 10th divide-by-16 rollover after "Write-to-SBUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of sixteen times the established baudrate. When a transition is detected, the divide-by-16 counter is immediately reset. The input shift register is written with $1FF_H$ and reception of the rest of the frame will proceed.

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

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

1. RI = 0, and
2. either SM2 = 0, or the received stop bit = 1

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

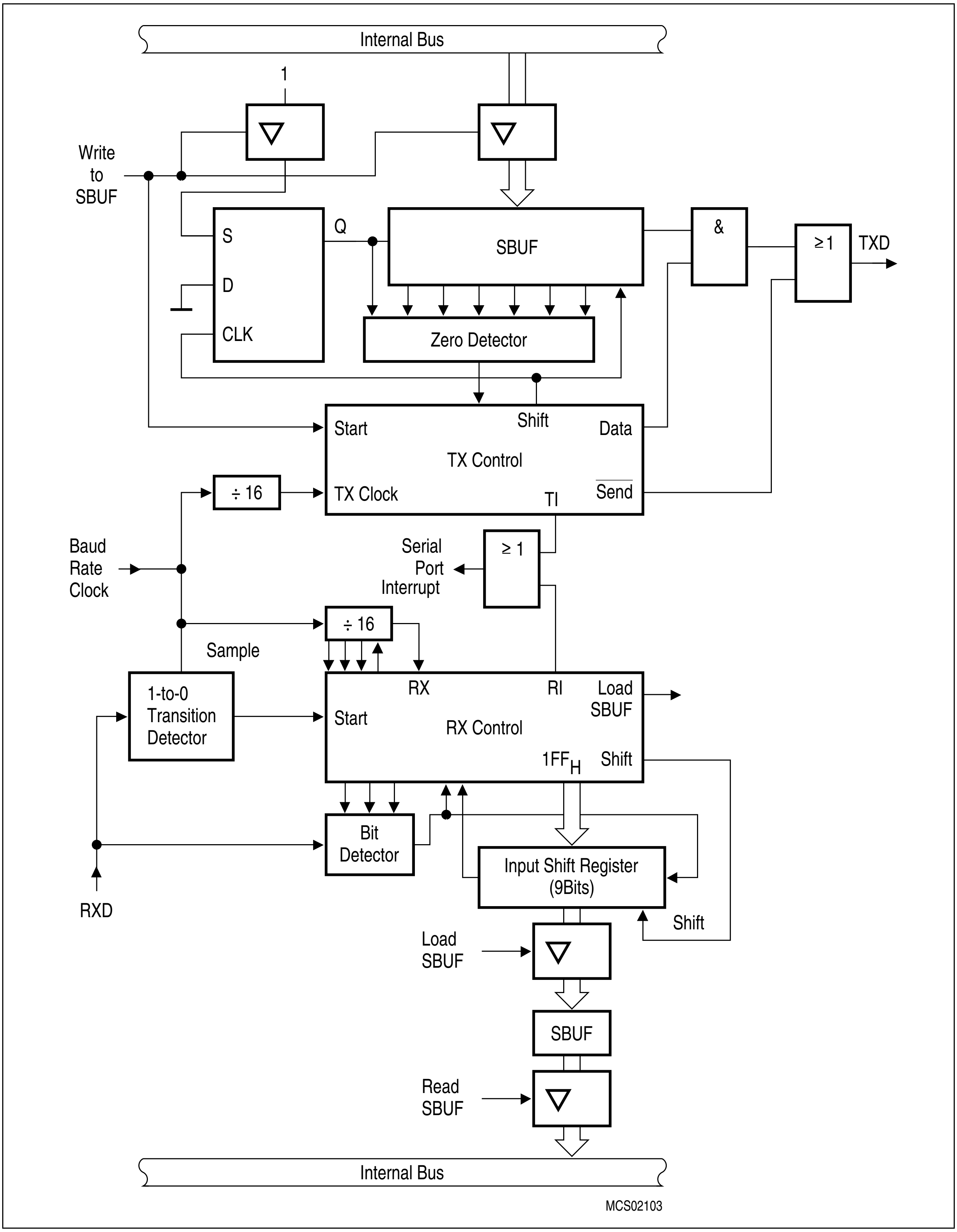


Figure 6-45 Serial Interface, Mode 1, Functional Diagram

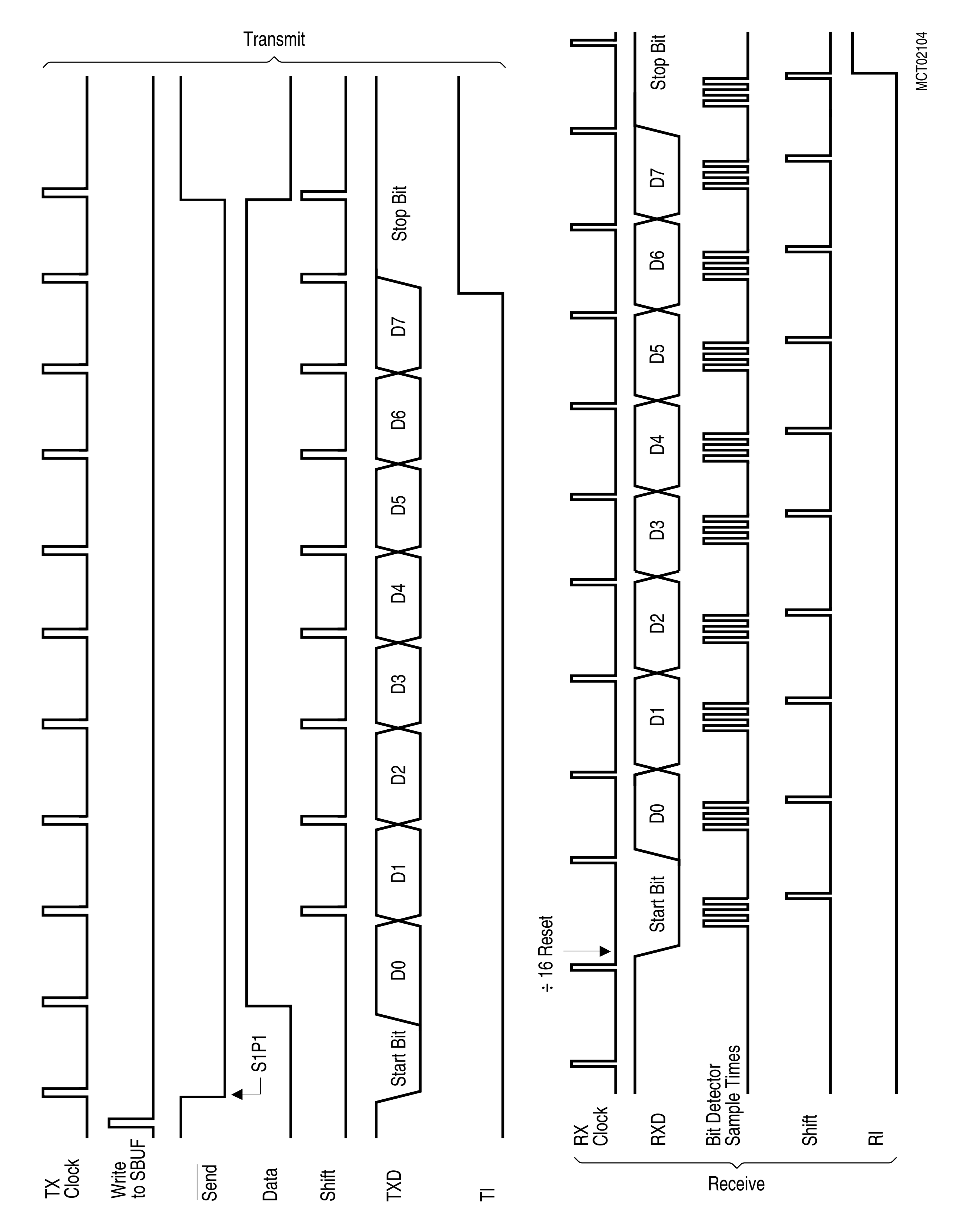


Figure 6-46 Serial Interface, Mode 1, Timing Diagram

6.4.6 Details about Modes 2 and 3

Eleven bits are transmitted through TxD or received through RxD: a start bit (0), eight data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On 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 6-47 shows a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register. The associated timings for transmit/receive are illustrated in **Figure 6-48**.

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

Transmission begins with the activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a '1' (the stop bit) into the 9th bit position of the shift register. Thereafter, only '0's are clocked in. Thus, as data bits shift out to the right, '0's are clocked in from the left. When TB8 is at the output position of the shift register, the stop bit is just to the left of TB8, and all positions to the left of that contain '0's. This condition flags the TX control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "Write-to-SBUF".

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

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

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

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

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

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

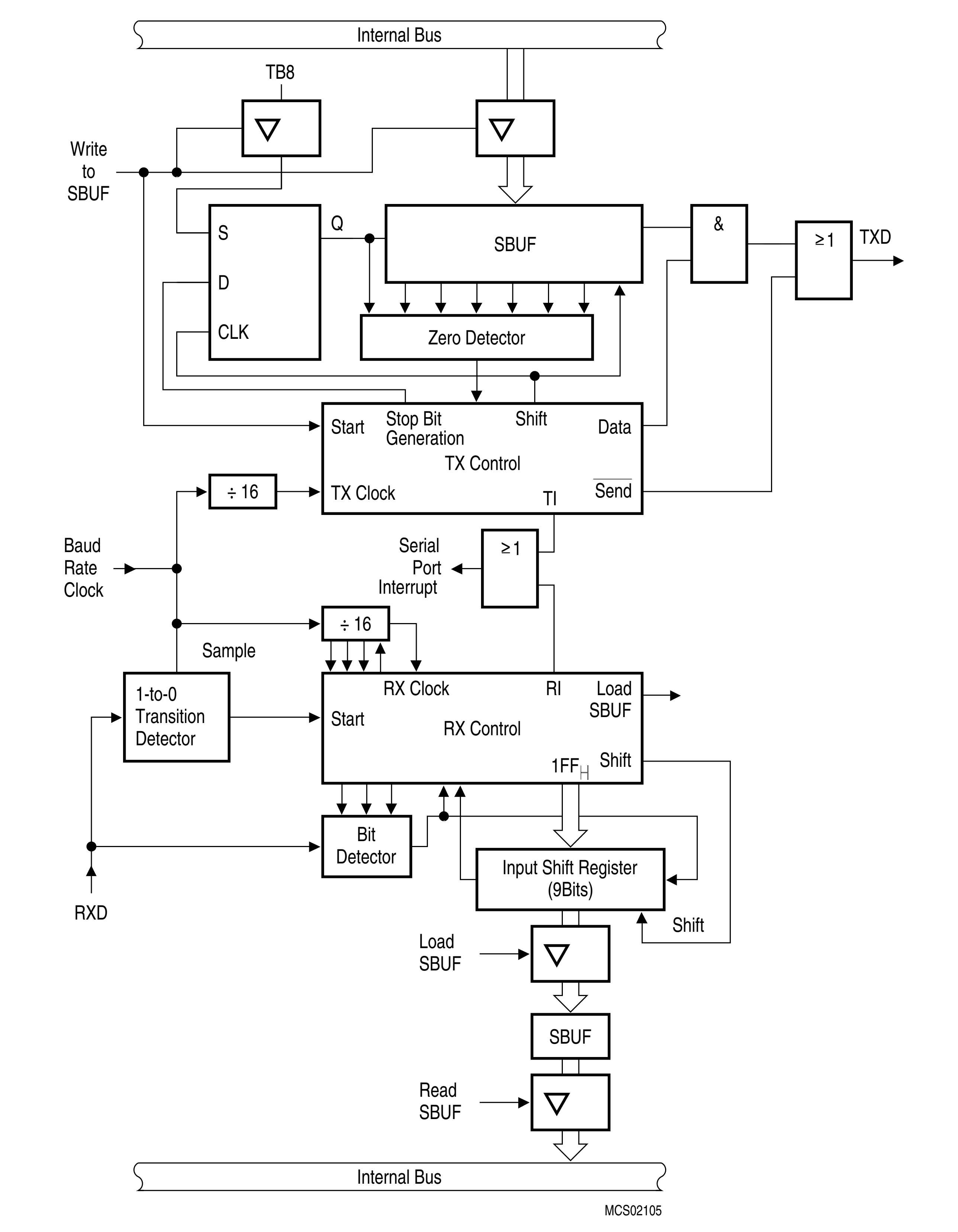


Figure 6-47 Serial Interface, Mode 2 and 3, Functional Diagram

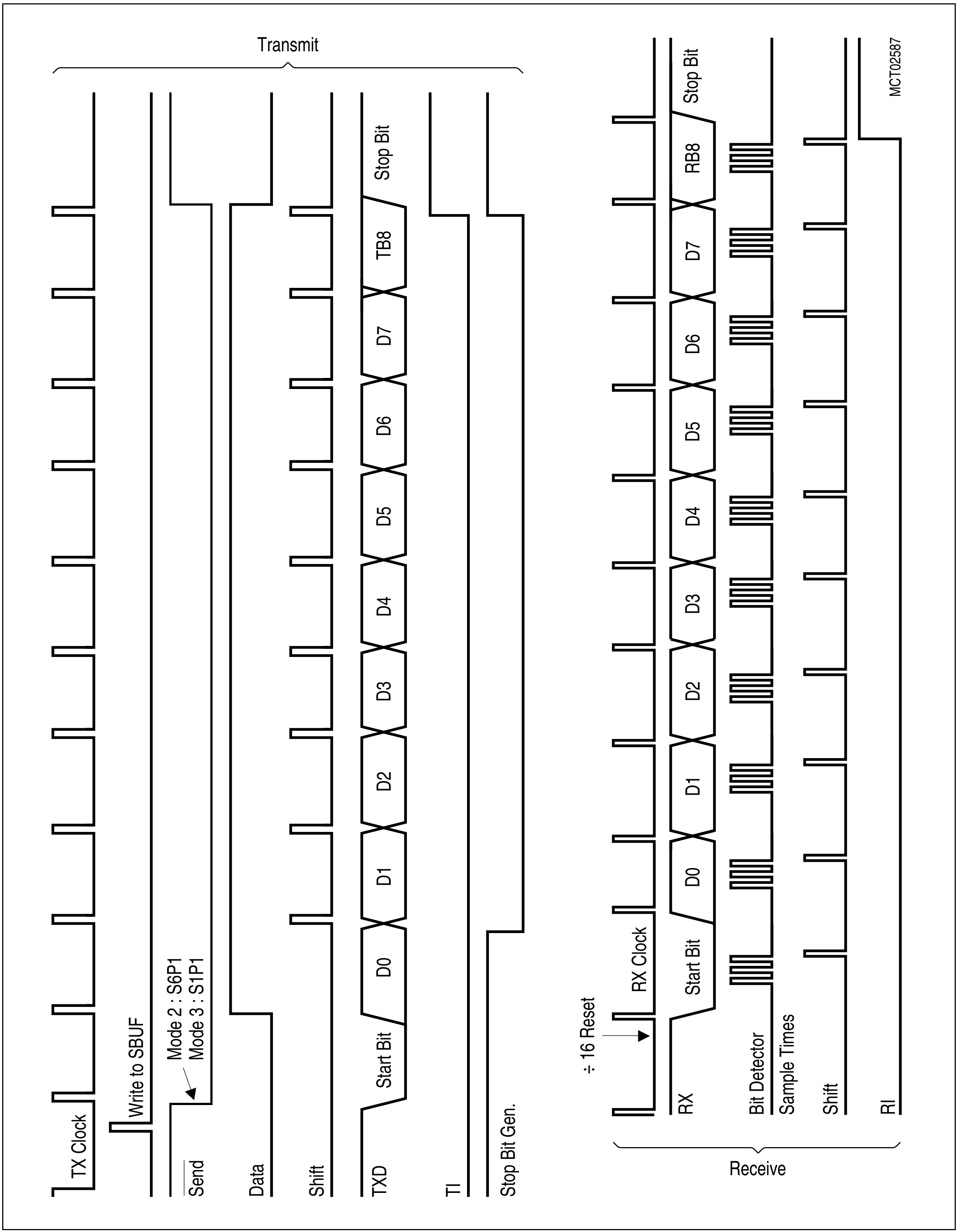


Figure 6-48 Serial Interface, Mode 2 and 3, Timing Diagram

6.5 A/D Converter

The C508 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 A/D Converter provides the following features:

- 8 input channels (Port 4), which can also be used as digital inputs
- 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 must be held at a fixed value within the specifications. The main functional blocks of the A/D Converter are shown in **Figure 6-49**.

6.5.1 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 which 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), a new A/D conversion is triggered automatically upon completion of a previous conversion, until bit ADM is reset.

The busy flag BSY (ADCON0.4) is automatically set when an A/D conversion is in progress. After completion of the conversion, it is reset by hardware. This flag is read only; 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 register 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 4 is an input port. These pins can be used either for digital input functions or as the analog inputs of the A/D Converter. If less than 8 analog inputs are required, the unused inputs are free for digital input functions. Any unused inputs should be connected to V_{SSA}.

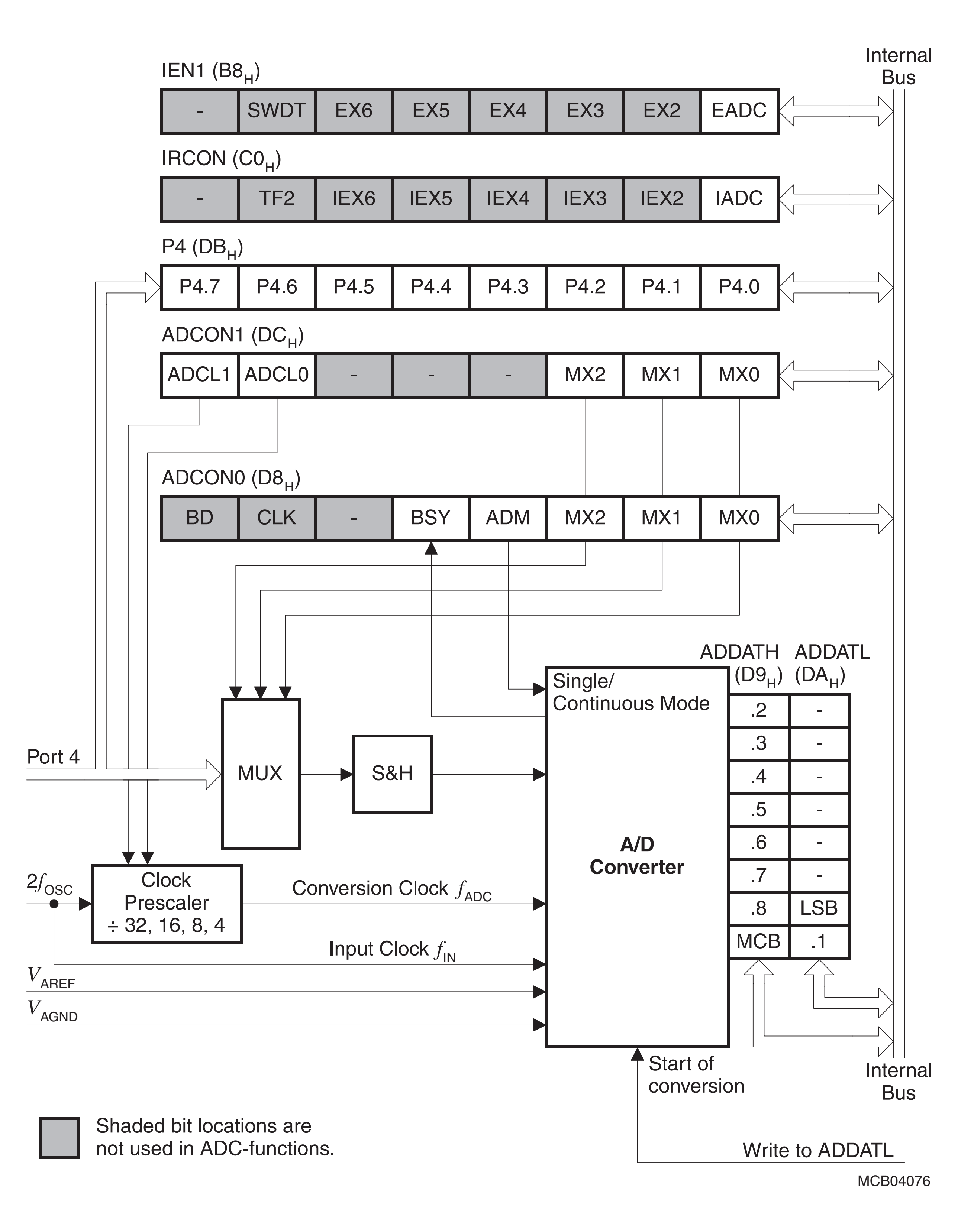


Figure 6-49 Block Diagram of the A/D Converter

6.5.2 A/D Converter Registers

This section describes the bits/functions of all registers which are used by the A/D Converter.

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							LSB	
	7	6	5	4	3	2	1	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 most significant bit of the 10-bit conversion result is bit 7 of ADDATH. The least significant bit 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 A/D Converter of the C508 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: $01XXX000_B$

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
$D8_H$	BD	CLK	–	BSY	ADM	MX2	MX1	MX0	ADCON0
DC_H	ADCL1	ADCL0	–	–	–	MX2	MX1	MX0	ADCON1

The shaded bits are not used for A/D Converter control.

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>–</td><td>Reserved bits for future use</td></tr>
<tr><td>BSY</td><td><b>Busy flag</b>
This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is completed.</td></tr>
<tr><td>ADM</td><td><b>A/D conversion mode</b>
When set, continuous A/D conversion is selected. If cleared during a running A/D conversion, the conversion is stopped at its end.</td></tr>
<tr><td>MX2 - MX0</td><td><b>A/D Converter input channel select bits</b>
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 the following table:
<table>
<tr><th>MX2</th><th>MX1</th><th>MX0</th><th>Selected Analog Input</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>P4.0 / AN0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>P4.1 / AN1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>P4.2 / AN2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>P4.3 / AN3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>P4.4 / AN4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>P4.5 / AN5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>P4.6 / AN6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>P4.7 / AN7</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>ADCL1
ADCL0</td><td><b>A/D Converter clock prescaler selection</b>
ADCL1 and ADCL0 select the prescaler ratio for the A/D conversion clock f_{ADC}. Depending on the clock rate f_{OSC} of the C508, f_{ADC} must be adjusted in a way that the resulting conversion clock f_{ADC} is less than or equal to 2 MHz (see <b>Chapter 6.5.3</b>).
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.

Note: Bit CLK of SFR ADCON0 must be written with a ‘0’.

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 (if no A/D conversion has occurred after the last reset operation).

When bit ADM is reset by software in continuous conversion mode, the current A/D conversion in progress will not be interrupted; it will be completed as the last conversion.

The A/D Converter interrupt is controlled by bits which are located in the SFRs IEN1 and IRCON.

Special Function Register IEN1 (Address $B8_H$) Reset Value: $X0000000_B$
Special Function Register IRCON (Address $C0_H$) Reset Value: $X0000000_B$

Bit No.	MSB BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	LSB $B8_H$	
$B8_H$	–	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	–	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

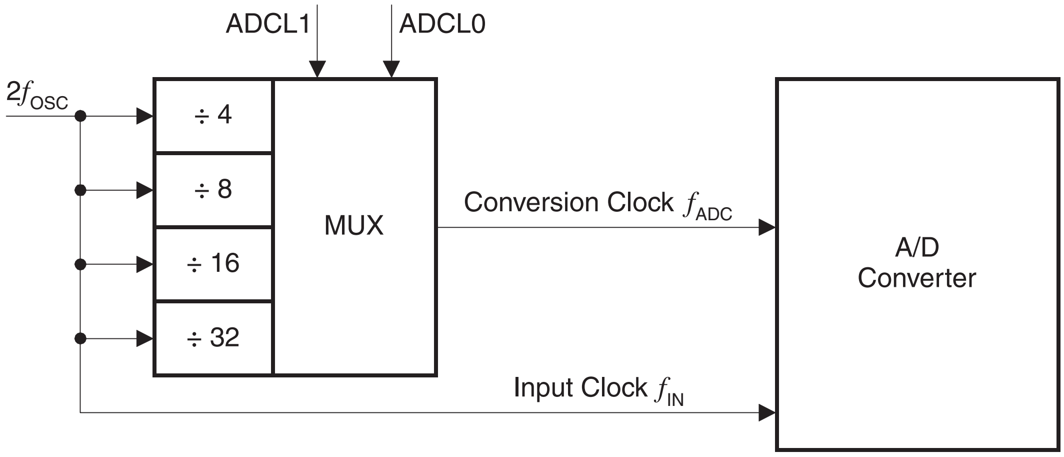
The shaded bits are not used for A/D Converter control.

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

6.5.3 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 C508 system clock, 2 × f_{OSC}, which is twice the crystal frequency applied at the XTAL pins via the ADC clock prescaler as shown in **Figure 6-50**. The input clock f_{IN} is equal to 2 × 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 which ensures 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 6-50** shows the prescaler ratio which must be selected by ADCL1 and ADCL0 for typical system clock rates. Up to 8 MHz external crystal frequency, the selected prescaler ratio must be at least 8. Between 8 MHz and 10 MHz, a prescaler ratio of at least 16 must be selected. A prescaler ratio of 32 can used for any of the above frequency ranges. A prescaler ratio of 4 should be used only when the C508 is operating in slowdown mode.



Condition: $f_{ADC\ max}$ = 2 MHz $f_{IN} = 2f_{OSC}$ = 4 TCL[1)]

Oscillator Clock Rate (f_{OSC})	**f_{IN} [MHz]**	**Prescaller Ratio**	**f_{ADC} [MHz]**	**ADCL1**	**ADCL0**
5 MHz	10	÷ 8	1.25	0	1
8 MHz	16	÷ 8	2	0	1
10 MHz	20	÷ 16	1.25	1	0

MCB04102

[1)] Note: Please refer to the C508 Data Sheet for the definition of TCL.

Figure 6-50 A/D Converter Clock Selection

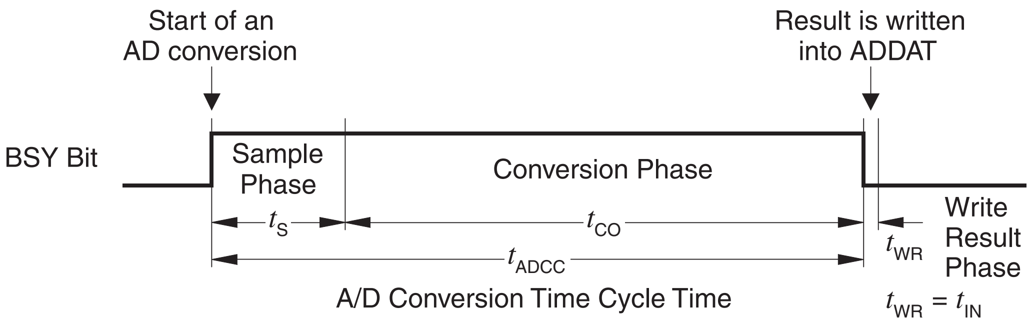
The duration of an A/D conversion is a multiple of the period of the f_{IN} clock signal. The calculation of the A/D conversion time is shown in the next section.

6.5.4 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 a 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 three phases of an A/D conversion is specified by their corresponding timing parameter as shown in **Figure 6-51**.



$t_{ADCC} = t_S + t_{CO}$
PS = Prescaler value

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

MCT04078

Figure 6-51 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 internally fed 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. In this calibration, alternating offset and linearity calibration cycles are executed (see also **Chapter 6.5.5**). At the end of the calibration time, the BSY bit is reset and the IADC bit in SFR IRCON is set indicating an A/D Converter interrupt condition.

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

Figure 6-52 shows how an A/D conversion is embedded into the microcontroller cycle scheme using the relation 6 × 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.

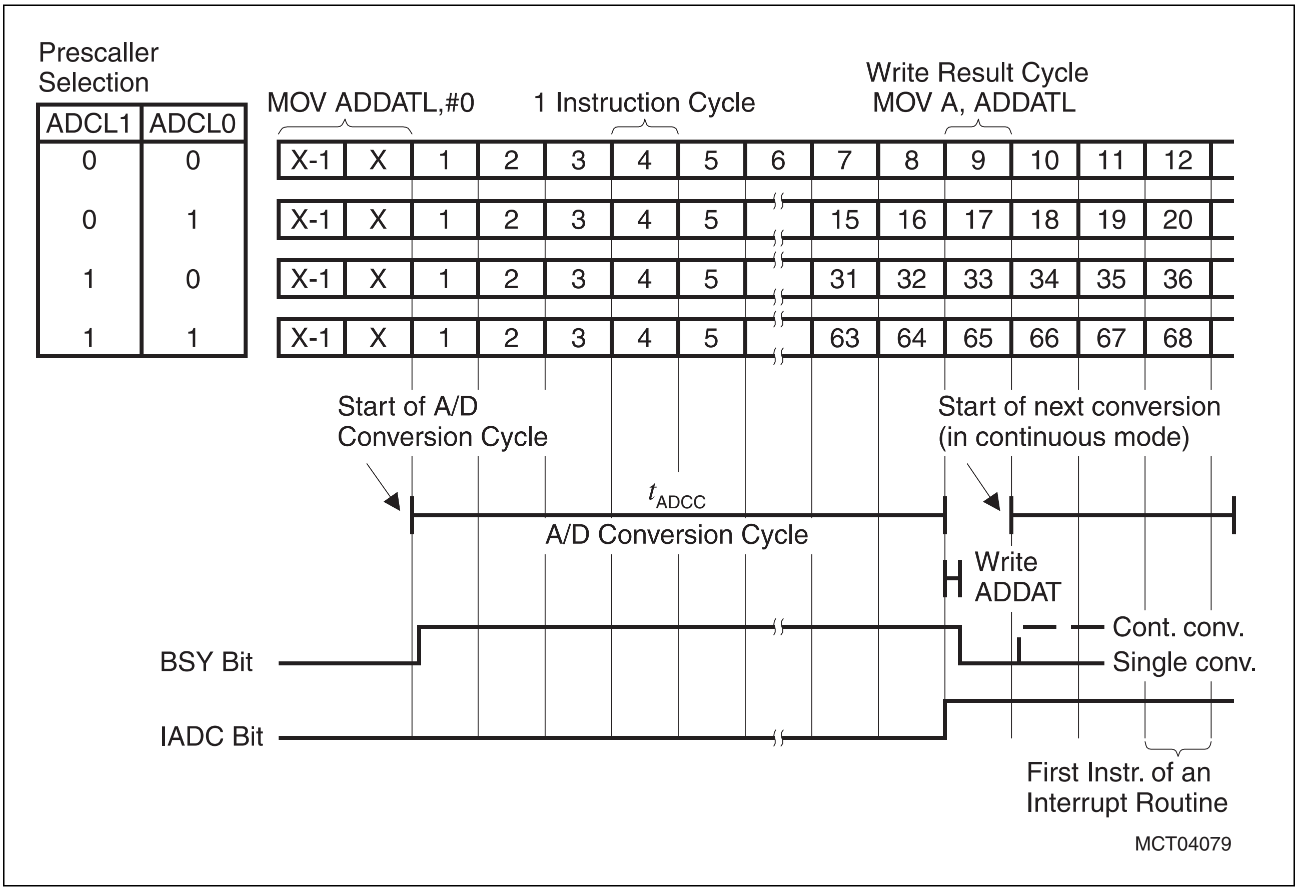

Prescaller Selection
ADCL1
ADCL0
0 0
0 1
1 0
1 1
MOV ADDATL,#0
1 Instruction Cycle
Write Result Cycle
MOV A, ADDATL
X-1 X 1 2 3 4 5 6 7 8 9 10 11 12
X-1 X 1 2 3 4 5 15 16 17 18 19 20
X-1 X 1 2 3 4 5 31 32 33 34 35 36
X-1 X 1 2 3 4 5 63 64 65 66 67 68
Start of A/D Conversion Cycle
Start of next conversion (in continuous mode)
t_{ADCC}
A/D Conversion Cycle
Write ADDAT
BSY Bit
Cont. conv.
Single conv.
IADC Bit
First Instr. of an Interrupt Routine
MCT04079

Figure 6-52 A/D Conversion Timing in Relation to Processor Cycles

Depending on the selected prescaler ratio (see **Figure 6-50**), three 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 6-52**, 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 phase) 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 this result can be read in the same cycle (for example, MOV A, ADDATL). If continuous conversion is selected (bit ADM set), the next conversion is started with the beginning of the machine cycle which 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 set again with the beginning of the machine cycle which follows the write result cycle.

The interrupt flag IADC is set at the end of the A/D conversion. If the A/D Converter interrupt is enabled and the A/D Converter interrupt is 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, typically there are three methods to handle the A/D conversion in the C508.

– 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. Attention: a polling JB instruction which is two machine cycles long, possibly 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 A/D Converter interrupt is enabled. The result of the A/D conversion is read in the interrupt service routine. If other C508 interrupts are enabled, the interrupt latency must be regarded. Therefore, this software method is the slowest method to get the result of an A/D conversion.

Depending on the oscillator frequency of the C508 and the selected divider ratio of the conversion clock prescaler, the total time of an A/D conversion is calculated according to **Figure 6-51** and **Table 6-14**. **Figure 6-53** on the next page shows the minimum A/D conversion time in relation to the oscillator frequency f_{OSC}. The minimum conversion time is 6 µs and can be achieved at f_{OSC} of 8 (or whenever f_{ADC} = 2 MHz).

Table 6-14 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]
5 MHz	÷ 8	1.25	1.6	9.6
6 MHz	÷ 8	1.5	1.33	8
8 MHz	÷ 8	2	1	6
10 MHz	÷ 16	1.25	1.6	9.6

Note: The prescaler ratios in ***Table 6-14*** *are minimum values.*

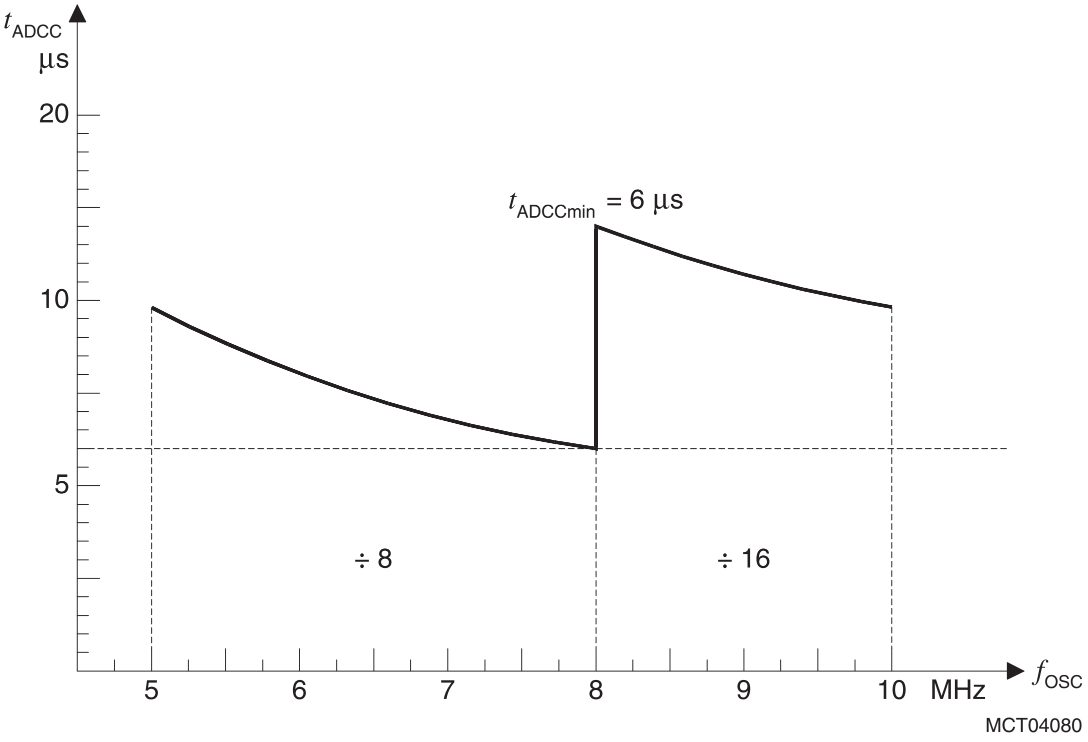


Figure 6-53 Minimum A/D Conversion Time in Relation to Oscillator Clock

6.5.5 A/D Converter Calibration

The C508 A/D Converter includes hidden internal calibration mechanisms which assure a safe functionality of the A/D Converter according to the DC characteristics. The A/D Converter calibration is implemented in a way that a user program which executes A/D conversions is not affected by its operation. Further, the 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 A/D Converter calibration operates in two phases. The first phase is the calibration after a reset operation and the second is the calibration at each A/D conversion. The calibration phases are controlled by a state machine in the A/D Converter. This state machine executes the calibration phases and stores the calibration results dynamically in a small calibration RAM.

After a reset operation, the A/D calibration is automatically started. In this reset calibration phase which takes 3328 f_{ADC} clocks, alternating offset and linearity calibration is executed. Therefore, at 8 MHz oscillator frequency, and with a default prescaler value of 8, a reset calibration time of approximately 1.664 ms is reached. For achieving a proper reset calibration, the f_{ADC} prescaler value must satisfy the condition $f_{\mathrm{ADC\ max}} \leq 2$ 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. It is also recommended to have the proper voltages, as specified in the Data Sheet, applied at the V_{AREF} and V_{AGND} pins before the reset calibration has started.

After the reset calibration phase, the A/D Converter is calibrated according to its DC characteristics. Nevertheless, during the reset calibration phase, single or continuous A/D conversion can be executed. In this case, it must be regarded that 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) needs to be valid for an A/D conversion, it is recommended to start the first A/D conversion after reset when the reset calibration phase has been completed. Depending on the oscillator frequency used, the reset calibration phase can be possibly 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, alternatively, offset and linearity calibration values, which are stored in the calibration RAM, are checked when an A/D conversion is executed. These values are corrected if required.

7 Interrupt System

The C508 provides nineteen interrupt vectors with four priority levels. Nine interrupt requests are generated by the on-chip peripherals (Timer 0, Timer 1, Timer 2, Serial Channel, A/D Converter, and the Capture/Compare Unit with four interrupts); ten interrupts may be triggered externally. Four of the external interrupts ($\overline{\text{INT3}}$, INT4, INT5 and INT6) can also be generated by Timer 2 in the capture/compare mode.

The wake-up from power-down mode interrupt has a special functionality which allows the software power-down mode to be terminated by a short negative pulse at pins P3.2/$\overline{\text{INT0}}$ or P5.7/INT7.

The nineteen interrupt sources are divided into six groups. Each group can be programmed to one of the four interrupt priority levels.

7.1 Structure of the Interrupt System

Figure 7-1 through **Figure 7-5** provide a general overview of the interrupt sources and illustrate the request and control flags described in the following sections.

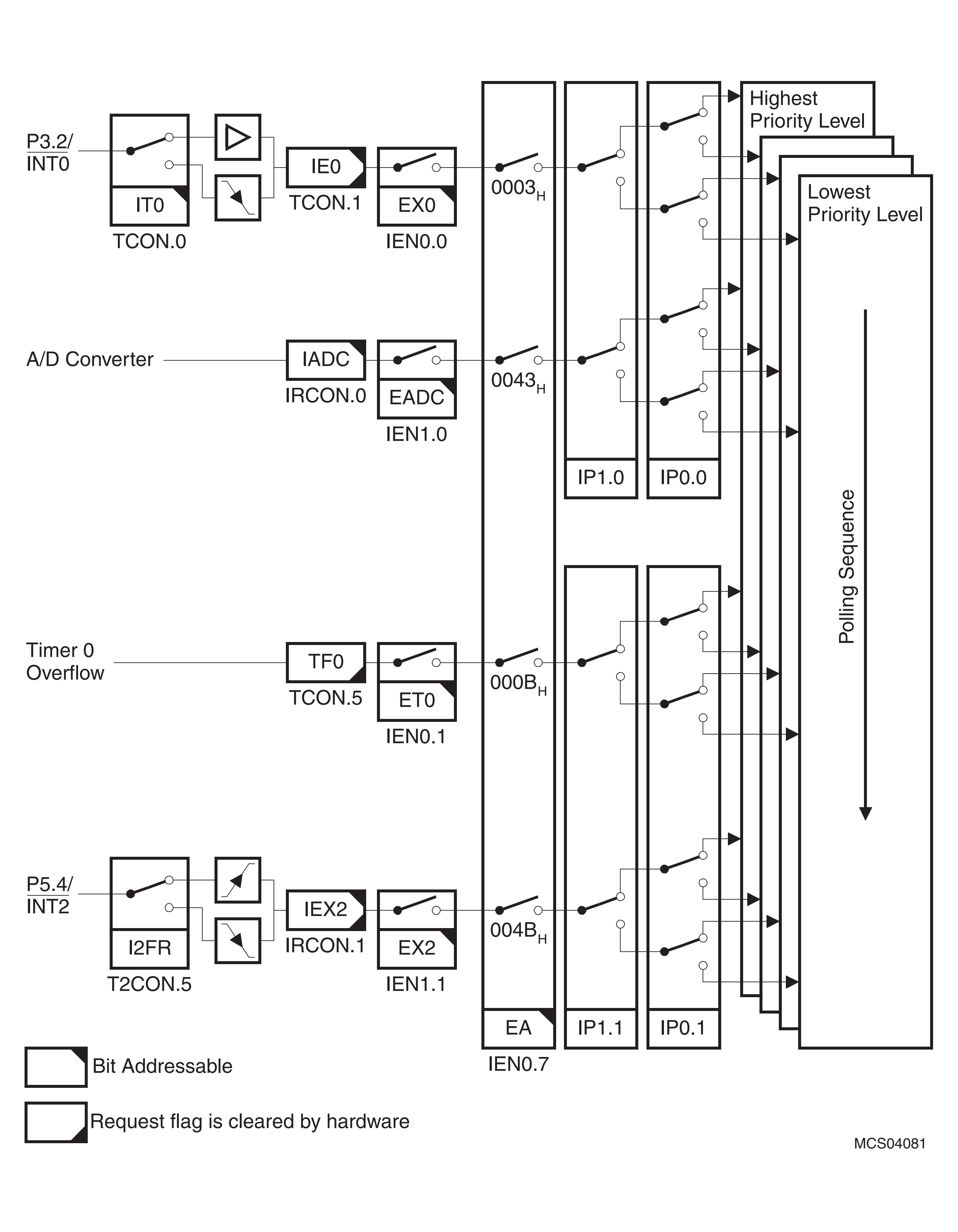


Figure 7-1 Interrupt Structure, Overview Part 1

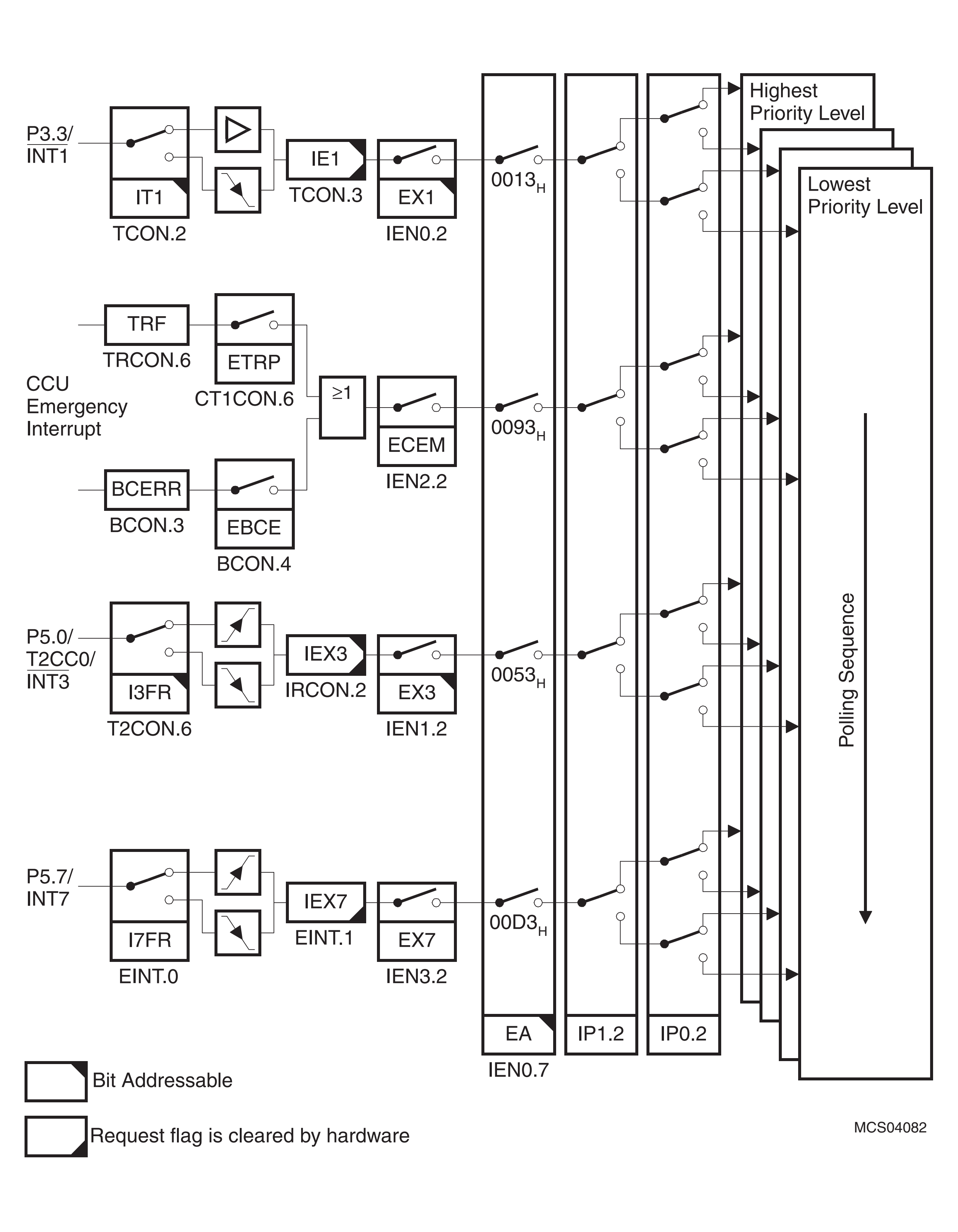


Figure 7-2 Interrupt Structure, Overview Part 2

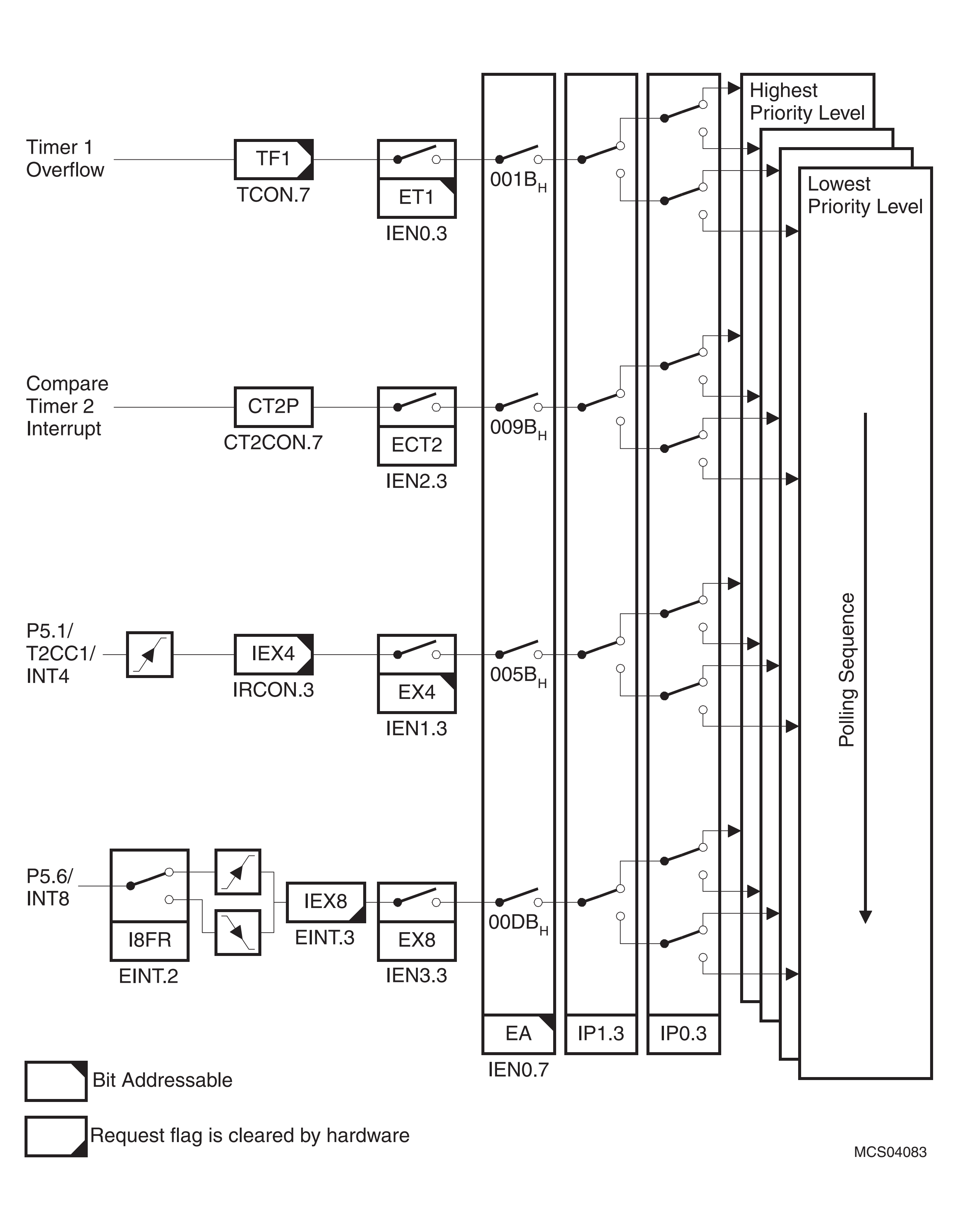


Figure 7-3 Interrupt Structure, Overview Part 3

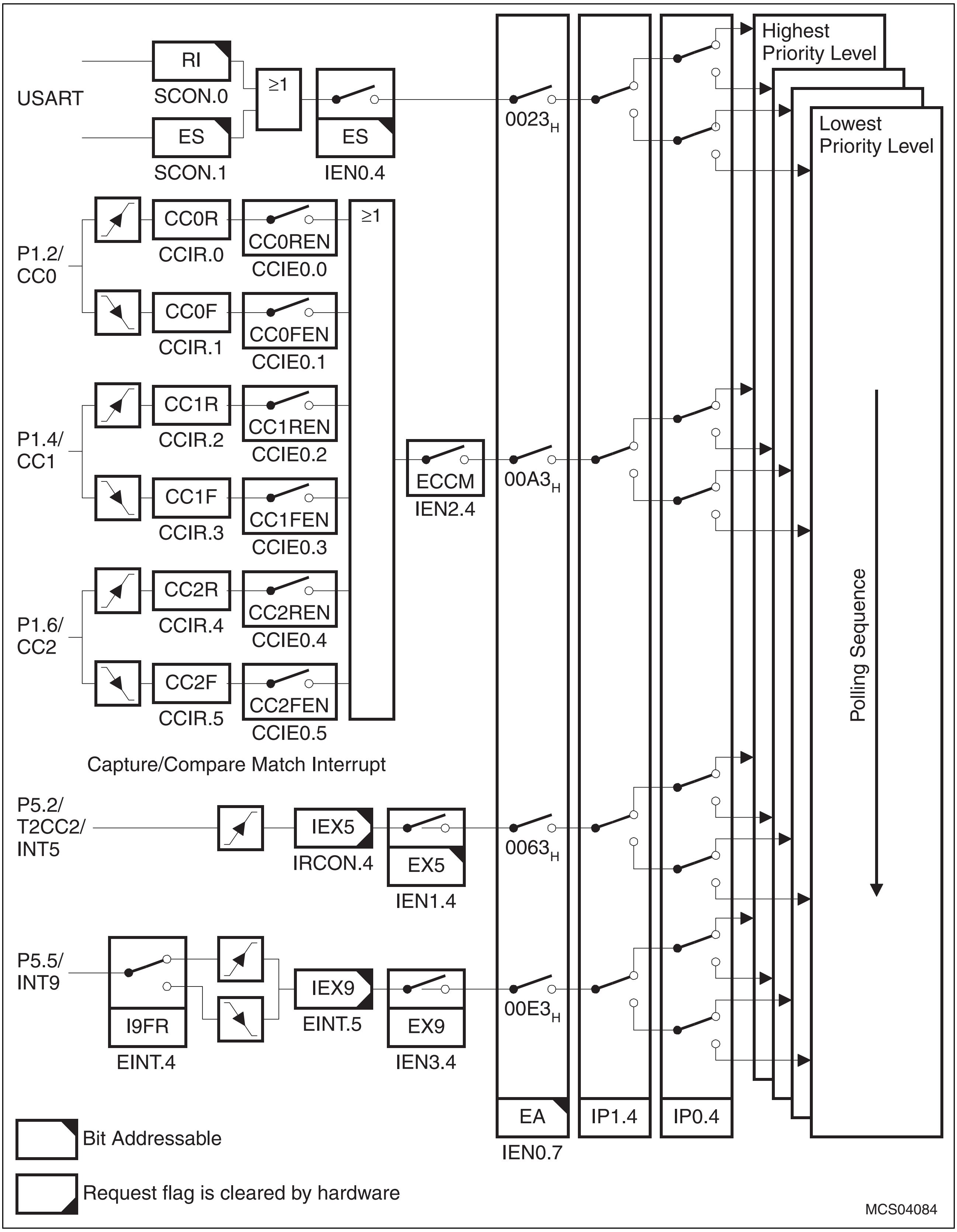


Figure 7-4 Interrupt Structure, Overview Part 4

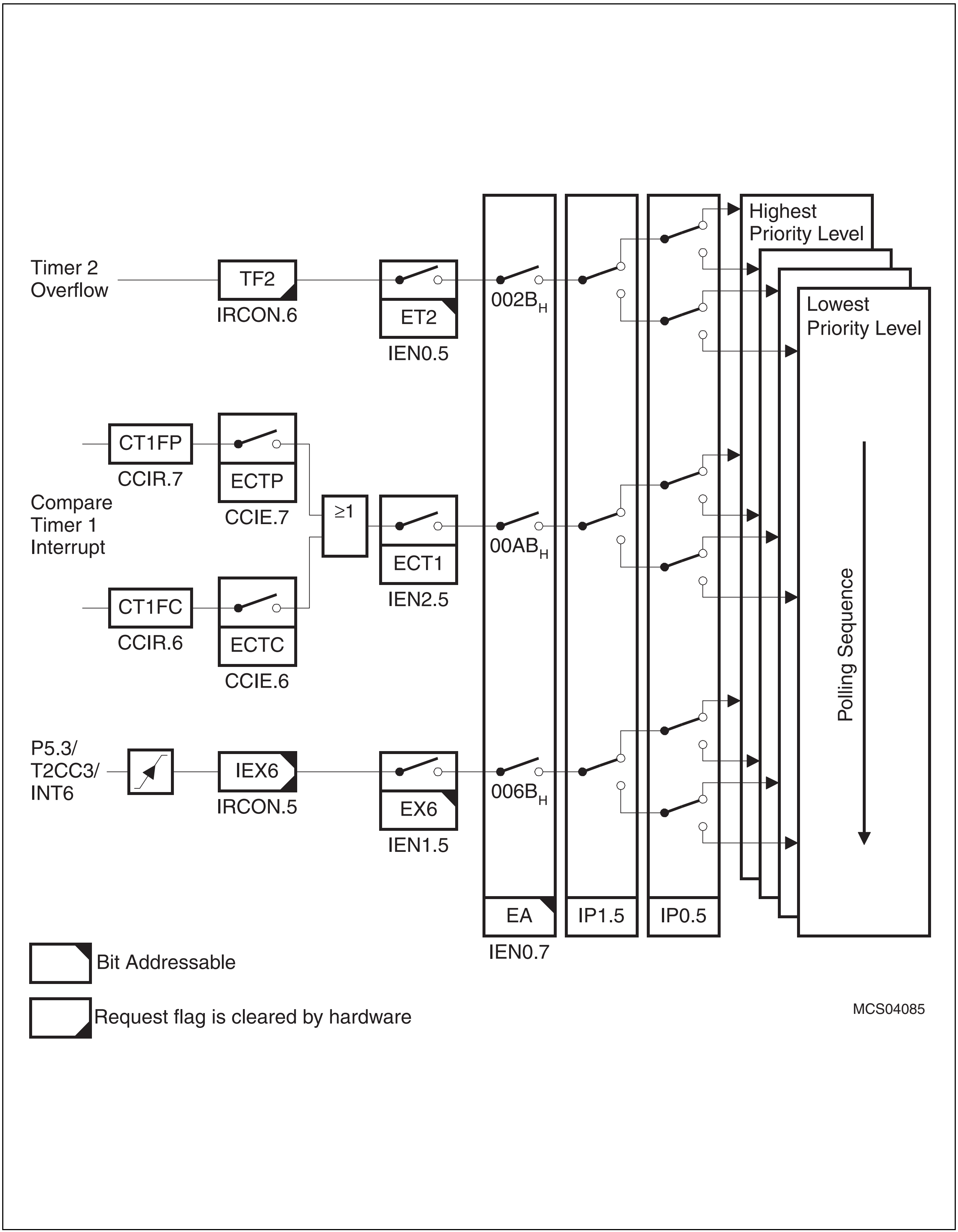


Figure 7-5 Interrupt Structure, Overview Part 5

7.2 Interrupt Registers

7.2.1 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, IEN1, IEN2 and IEN3. Register IEN0 also contains the global disable bit (EA), which can be cleared to disable all interrupts at once. Generally, all interrupt enable bits are cleared to 0 after reset; the corresponding interrupts are disabled.

The SFR IEN0 contains the enable bits for 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.

Bit	Function
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 SFR IEN1 contains the enable bits for the external interrupts 2 to 6, and the A/D Converter interrupt.

Special Function Register IEN1 (Address $B8_H$) Reset Value: $X0000000_B$

	MSB							LSB	
Bit No.	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	–	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

▨ The shaded bits are not used for interrupt control.

Bit	Function
EX6	**External interrupt 6/Timer 2 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/Timer 2 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/Timer 2 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/Timer 2 capture/compare interrupt 0 enable** If EX3 = 0, external interrupt 3 is disabled. If EX3 = 1, external interrupt 3 is enabled.
EX2	**External interrupt 2 enable** If EX2 = 0, external interrupt 2 is disabled. If EX2 = 1, external interrupt 2 is enabled.
EADC	**A/D Converter interrupt enable** If EADC = 0, the A/D Converter interrupt is disabled. If EADC = 1, the A/D Converter interrupt is enabled.

The SFR IEN2 contains the enable bits for the four interrupts from the Capture/Compare Unit (CCU).

Special Function Register IEN2 (Address 9A$_H$) **Reset Value: XX0000XX$_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
9A$_H$	–	–	ECT1	ECCM	ECT2	ECEM	–	–	IEN2

Bit	Function
ECT1	**Compare Timer 1 interrupt enable** If ECT1 = 0, the compare timer 1 interrupt is disabled. If ECT1 = 1, the compare timer 1 interrupt is enabled.
ECCM	**Compare/Capture match interrupt enable** If ECCM = 0, the compare/capture match interrupt is disabled. If ECCM = 1, the compare/capture match interrupt is enabled.
ECT2	**Compare Timer 2 interrupt enable** If ECT2 = 0, the compare timer 2 interrupt is disabled. If ECT2 = 1, the compare timer 2 interrupt is enabled.
ECEM	**CCU emergency interrupt enable** If ECEM = 0, the emergency interrupt of the CCU is disabled. If ECEM = 1, the emergency interrupt of the CCU is enabled.

The SFR IEN3 contains the enable bits for the external interrupts 7 to 9.

Special Function Register IEN3 (Address BE_H) **Reset Value: $XXX000XX_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
BE_H	–	–	–	EX9	EX8	EX7	–	–	IEN3

Bit	Function
EX9	**External interrupt 9 enable** If EX9 = 0, external interrupt 9 is disabled. If EX9 = 1, external interrupt 9 is enabled.
EX8	**External interrupt 8 enable** If EX8 = 0, external interrupt 8 is disabled. If EX8 = 1, external interrupt 8 is enabled.
EX7	**External interrupt 7 enable** If EX7 = 0, external interrupt 7 is disabled. If EX7 = 1, external interrupt 7 is enabled.

7.2.2 Interrupt Request Flags

The request flags for the various interrupt sources are located in several Special Function Registers. This section describes the locations and meanings of these interrupt request flags in detail.

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

	MSB							LSB	
Bit No.	8F$_H$	8E$_H$	8D$_H$	8C$_H$	8B$_H$	8A$_H$	89$_H$	88$_H$	
88$_H$	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

▨ The shaded bits are not used for interrupt control.

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.

Each of the **external interrupts 0 and 1** (P3.2/$\overline{\text{INT0}}$ and P3.3/$\overline{\text{INT1}}$) can be 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 which generated this interrupt is cleared by the hardware when the service routine is vectored to – but, only if the interrupt was transition-activated. If, however, the interrupt was level-activated, the requesting external source controls the request flag directly, 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 the service routine is vectored to.

Special Function Register T2CON (Address $C8_H$) Reset Value: $0000X0X0_B$

	MSB							LSB	
Bit No.	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	T2PS	I3FR	I2FR	T2R	–	T2CM	–	T2I	T2CON

▨ The shaded bits are not used for interrupt control.

Bit	Function
I2FR	**External interrupt 2 rising/falling edge control flag** If I2FR = 0, the external interrupt 2 is activated by a falling edge at P5.4/$\overline{\text{INT2}}$. If I2FR = 1, the external interrupt 2 is activated by a rising edge at P5.4/$\overline{\text{INT2}}$.
I3FR	**External interrupt 3 rising/falling edge control flag** If I3FR = 0, the external interrupt 3 is activated by a falling edge at P5.0/ T2CC0/$\overline{\text{INT3}}$. If I3FR = 1, the external interrupt 3 is activated by a rising edge at P5.0/ T2CC0/$\overline{\text{INT3}}$.

The **external interrupt 2** ($\overline{\text{INT2}}$) can be either positive or negative transition-activated, depending on bit I2FR in register T2CON. The flag that actually generates this interrupt is bit IEX2 in register IRCON. The flag IEX2 is cleared by hardware when the service routine is vectored to.

As with the external interrupt 2, the **external interrupt 3** ($\overline{\text{INT3}}$) can be either positive or negative transition-activated, 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 P5.0/T2CC0/$\overline{\text{INT3}}$, regardless of the compare mode established and the transition at the respective pin. The flag IEX3 is cleared by hardware when the service routine is vectored to.

The external interrupts 4 (INT4), 5(INT5) and 6(INT6) are positive transition-activated. The flags that actually generate these interrupts are bits IEX4, IEX5 and IEX6 in register IRCON. These flags will also be set if a compare event occurs at the corresponding Pins P5.1/T2CC1/INT4, P5.2/T2CC2/INT5 and P5.3/T2CC3/INT6, 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 hardware when the service routine is vectored to.

Special Function Register IRCON (Address C0$_H$) **Reset Value: X0000000$_B$**

	MSB							LSB	
Bit No.	C7$_H$	C6$_H$	C5$_H$	C4$_H$	C3$_H$	C2$_H$	C1$_H$	C0$_H$	
C0$_H$	–	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

Bit	Function
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 P5.3/T2CC3/INT6. Cleared when the interrupt is processed.
IEX5	**External interrupt 5 edge flag** Set by hardware when external interrupt edge was detected or when a compare event occurred at P5.2/T2CC2/INT5. Cleared when the interrupt is processed.
IEX4	**External interrupt 4 edge flag** Set by hardware when external interrupt edge was detected or when a compare event occurred at P5.1/T2CC1/INT4. Cleared when the interrupt is processed.
IEX3	**External interrupt 3 edge flag** Set by hardware when external interrupt edge was detected or when a compare event occurred at P5.0/T2CC0/$\overline{\text{INT3}}$. Cleared when the interrupt is processed.
IEX2	**External interrupt 2 edge flag** Set by hardware when external interrupt edge was detected or when a compare event occurred at P5.4/$\overline{\text{INT2}}$. Cleared when the interrupt is processed.
IADC	**A/D Converter 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 bit TF2 in register IRCON. This flag is not cleared by hardware when the service routine is vectored to. It should be cleared by software.

The **A/D Converter interrupt** is generated by IADC bit in register IRCON. If an interrupt is generated, 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 A/D Converter interrupt is generated, flag IADC will need to be cleared by software.

Special Function Register EINT (Address FB_H) **Reset Value: $XX000000_B$**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
FB_H	–	–	IEX9	I9FR	IEX8	I8FR	IEX7	I7FR	EINT

Bit	Function
IEX9	**External interrupt 9 edge flag** Set by hardware when external interrupt edge was detected. Cleared by hardware when processor vectors to the interrupt routine.
I9FR	**External interrupt 9 rising/falling edge control flag** If I9FR = 0, the external interrupt 9 is activated by a negative edge transition at P5.5/INT9. If I9FR = 1, the external interrupt 9 is activated by a positive edge transition at P5.5/INT9.
IEX8	**External interrupt 8 edge flag** Set by hardware when external interrupt edge was detected. Cleared by hardware when processor vectors to the interrupt routine.
I8FR	**External interrupt 8 rising/falling edge control flag** If I8FR = 0, the external interrupt 8 is activated by a negative edge transition at P5.6/INT8. If I8FR = 1, the external interrupt 8 is activated by a positive edge transition at P5.6/INT8.
IEX7	**External interrupt 7 edge flag** Set by hardware when external interrupt edge was detected. Cleared by hardware when processor vectors to the interrupt routine.
I7FR	**External interrupt 7 rising/falling edge control flag** If I7FR = 0, the external interrupt 7 is activated by a negative edge transition at P5.7/INT7. If I7FR = 1, the external interrupt 7 is activated by a positive edge transition at P5.7/INT7.

The **external interrupts 7(INT7), 8(INT8)** and **9(INT9)** can be either positive or negative transition-activated, depending on bits I7FR, I8FR and I9FR in register EINT. The flags that actually generate these interrupts are bits IEX7, IEX8 and IEX9 in the same register EINT. They are cleared by hardware when the respective service routine is vectored to.

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

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

▨ The shaded bits are not used for interrupt 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 interface interrupt** is generated by a logical OR of flag RI and TI in SFR SCON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally need to determine whether it was the receive interrupt flag or the transmission interrupt flag which generated the interrupt, and the corresponding bit will need to be cleared by software.

The interrupt request flags of the **CAPCOM capture/compare match interrupt** are located in the register CCIR. All CAPCOM capture/compare match interrupt flags are set by hardware and must be cleared by software. A capture/compare match interrupt is generated with the setting of a CCxR bit (x = 0-2) if the corresponding enable bits are set. These enable bits are contained in register CCIE. The **Compare Timer 1 interrupt** request flags – CT1FP or CT1FC – are also located in the register CCIR. Each flag has a corresponding enable bit which is located in the register CCIE. However, the **Compare Timer 2 interrupt** request flag, CT2P, is located in register CT2CON. The **CCU emergency interrupt** can be triggered by either bit TRF located in register TRCON or by bit BCERR located in register BCON. Each flag has an enable bit. For bit TRF, it is located in register CT1CON; whereas, for bit BCERR, it is found in register BCON.

Special Function Register CCIR (Address E5$_H$) **Reset Value: 00$_H$**

Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
E5$_H$	CT1FP	CT1FC	CC2F	CC2R	CC1F	CC1R	CC0F	CC0R	CCIR
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

Bit	Function
CT1FP	**Compare Timer 1 period flag** Compare Timer 1 operating mode 0: CT1FP is set if Compare Timer 1 reaches the period value. Compare Timer 1 operating mode 1: CT1FP is set if Compare Timer 1 reaches the period value and changes the count direction from up- to down counting. Bit CT1FP must be cleared by software. If Compare Timer 1 interrupt is enabled, the setting of CT1FP will generate a Compare Timer 1 interrupt.
CT1FC	**Compare Timer 1 count direction change flag** This flag can only be set if Compare Timer 1 runs in operating mode 1 (CTM = 1). CT1FC is set when Compare Timer 1 reaches count value 0000$_H$ and changes the count direction from down- to up-counting. If Compare Timer 1 interrupt is enabled, the setting of CT1FC will generate a Compare Timer 1 interrupt. Bit CT1FC must be cleared by software.
CCxR x=0-2	**Capture/Compare match on up-count flag** Capture Mode: CCxR is set at a low-to-high transition (rising edge) of the corresponding CCx capture input signal. Compare Mode: CCxR is set if the Compare Timer 1 value matches the compare register CCx value during the up-count phase.
CCxF x=0-2	**Capture/Compare match on down-count flag** Capture Mode: CCxF is set at a high-to-low transition (falling edge) of the corresponding CCx capture input signal. Compare Mode: CCxF is set if the Compare Timer 1 value matches the compare register CCx value during the down-count phase (only in Compare Timer 1 operating mode 1).

Special Function Registers CCIE (Address D6$_H$) **Reset Value: 00$_H$**

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
D6$_H$	ECTP	ECTC	CC2FEN	CC2REN	CC1FEN	CC1REN	CC0FEN	CC0REN	CCIE

Bit	Function
ECTP	**Enable Compare Timer 1 period interrupt** If ECTP = 0, the compare timer 1 period interrupt is disabled. Compare Timer 1 operating mode 0: If ECTP = 1, an interrupt is generated when Compare Timer 1 reaches the period value. Compare Timer 1 operating mode 1: If ECTP = 1, an interrupt is generated when Compare Timer 1 reaches the period value and changes the count direction from up- to down-counting.
ECTC	**Enable Compare Timer 1 count direction change interrupt status** If ECTC = 0, the Compare Timer 1 count change interrupt is disabled. Compare Timer 1 operating mode 0: Bit has no effect on the interrupt generation. Compare Timer 1 operating mode 1: If ECTC = 1, an interrupt is generated when Compare Timer 1 reaches count value 0000$_H$ and changes its count direction from down- to up-counting.
CCxREN (x = 0-2)	**Capture/Compare rising edge interrupt enable** Capture Mode: If CCxREN is set, an interrupt is generated at a low-to-high transition (rising edge) of the corresponding CCx input signal. Compare Mode: If CCxREN is set, an interrupt is generated if the Compare Timer 1 value matches the compare register CCx value during the up-counting phase of the Compare Timer 1. This function is available in both Compare Timer 1 operating modes.
CCxFEN (x = 0-2)	**Capture/Compare falling edge interrupt enable** Capture Mode: If CCxFEN is set, an interrupt is generated at a high-to-low transition (falling edge) of the corresponding CCx input signal. Compare Mode: If CCxFEN is set, an interrupt is generated only in compare timer mode 1 if the Compare Timer 1 value matches the compare register CCx value during the down-counting phase of the Compare Timer 1. This function is available only in Compare Timer 1 operating mode 1.

Special Function Register CT2CON (Address $F1_H$) Reset Value: 00010000_B

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$F1_H$	CT2P	▨ ECT2O	▨ STE2	▨ CT2RES	▨ CT2R	▨ CLK2	▨ CLK1	▨ CLK0	CT2CON

▨ The shaded bits are not used for interrupt control.

Bit	Function
CT2P	**Compare Timer 2 period flag** When the Compare Timer 2 value matches the Compare Timer 2 period register value, bit CT2P is set. If the Compare Timer 2 interrupt is enabled, the setting of CT2P will generate a Compare Timer 2 interrupt. Bit CT2P must be cleared by software.

Special Function Register BCON (Address $D7_H$) Reset Value: 00_H

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
$D7_H$	▨ BCMP BCEM	▨ PWM1	▨ PWM0	EBCE	BCERR	▨ BCEN	▨ BCM1	▨ BCM0	BCON

▨ The shaded bits are not used for interrupt control.

Bit	Function
EBCE	**Enable interrupt of Block Commutation mode Error** If EBCE is set, the emergency interrupt for a block commutation mode error condition of the CCU is enabled. In block commutation mode, an emergency error condition occurs if a false signal state at $\overline{\text{INT2}}$ - $\overline{\text{INT0}}$ or a wrong follower state (if selected by bit BCEM) is detected.
BCERR	**Block Commutation mode Error flag** In block commutation mode, BCERR is set in rotate right or rotate left mode if, after a transition at $\overline{\text{INTx}}$, all $\overline{\text{INTx}}$ inputs are at high or low level. Additionally, in rotate right or rotate left mode, a “wrong follower” condition according to **Table 6-10** can cause BCERR to be set (see description of bit BCEM). If the block commutation interrupt is enabled (EBCE = 1), the setting of BCERR will generate a CCU emergency interrupt. BCERR must be reset by software.

Special Function Register TRCON (Address FF$_H$) Reset Value: 00$_H$

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
FF$_H$	▨ TRPEN	TRF	▨ TREN5	▨ TREN4	▨ TREN3	▨ TREN2	▨ TREN1	▨ TREN0	TRCON
			CAPCOM Channel 2		CAPCOM Channel 1		CAPCOM Channel 0		

▨ The shaded bits are not used for interrupt control.

Bit	Function
TRF	**Trap flag** TRF is set by hardware if the trap function is enabled (TRPEN = 1) and the $\overline{\text{CTRAP}}$ level becomes active (low). If enabled, an interrupt is generated when TRF is set. TRF must be reset by software.

Special Function Register CT1CON (Address E1$_H$) Reset Value: 00010000$_B$

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	0	
E1$_H$	▨ CTM	ETRP	▨ STE1	▨ CT1RES	▨ CT1R	▨ CLK2	▨ CLK1	▨ CLK0	CT1CON

Bit	Function
ETRP	**CCU emergency trap interrupt enable** If ETRP = 1, the emergency interrupt for the CCU trap signal is enabled.

7.2.3 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.

<table>
<tr><th>Bit</th><th>Function</th></tr>
<tr><td>IP1.x
IP0.x</td><td><b>Interrupt group Priority level bits</b> (x = 0-5, see <b>Table 7-1</b>)
<table>
<tr><th>IP1.x</th><th>IP0.x</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Interrupt group x is set to Priority level 0 (lowest)</td></tr>
<tr><td>0</td><td>1</td><td>Interrupt group x is set to Priority level 1</td></tr>
<tr><td>1</td><td>0</td><td>Interrupt group x is set to Priority level 2</td></tr>
<tr><td>1</td><td>1</td><td>Interrupt group x is set to Priority level 3 (highest)</td></tr>
</table></td></tr>
</table>

7.3 Interrupt Priority Level Structure

The nineteen interrupt sources of the C508 are grouped according to the listing in **Table 7-1**.

Table 7-1 Interrupt Source Structure

Interrupt Group	Associated Interrupts			
1	External interrupt 0	–	A/D Converter interrupt	–
2	Timer 0 overflow	–	External interrupt 2	–
3	External interrupt 1	CCU emergency interrupt	External interrupt 3	External interrupt 7
4	Timer 1 overflow	Compare Timer 2 interrupt	External interrupt 4	External interrupt 8
5	Serial channel interrupt	Capture/Compare match interrupt	External interrupt 5	External interrupt 9
6	Timer 2 overflow	Compare Timer 1 interrupt	External interrupt 6	–

Each group of interrupt sources can be programmed individually to one of the four priority levels by setting or clearing one bit in the Special Function Register 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 polling sequence. This is illustrated in **Table 7-2**.

Table 7-2 Interrupts - Priority-within-Level

Interrupt Group	Priority Bits of Interrupt Group	Interrupt Source Priority				Priority
		High Priority	→		Low Priority	
1	IP1.0 / IP0.0	IE0	–	IADC	–	High
2	IP1.1 / IP0.1	TF0	–	IEX2	–	↓
3	IP1.2 / IP0.2	IE1	TRF + BCERR	IEX3	IEX7	
4	IP1.3 / IP0.3	TF1	CT2P	IEX4	IEX8	
5	IP1.4 / IP0.4	RI + TI	CCxR + CCxF	IEX5	IEX9	
6	IP1.5 / IP0.5	TF2	CT1FP + CT1FC	IEX6	–	Low

Within a group, the leftmost interrupt is serviced first, then the second and the third and the fourth, when available. The interrupt groups are serviced from top to bottom of the table. A low-priority interrupt can itself be interrupted by a higher-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 which is illustrated in **Table 7-2**.

The “priority-within-level” structure is used only to resolve simultaneous requests of the same priority level.

7.4 Interrupt Handling

The interrupt flags are sampled at S5P2 in each machine cycle. The sampled flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceeding cycle, the polling cycle will find it and the interrupt system will generate an 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 any interrupt is vectored to; this delay guarantees that changes of the interrupt status can be observed by the CPU.

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

The polling cycle/LCALL sequence is illustrated in **Figure 7-6**.

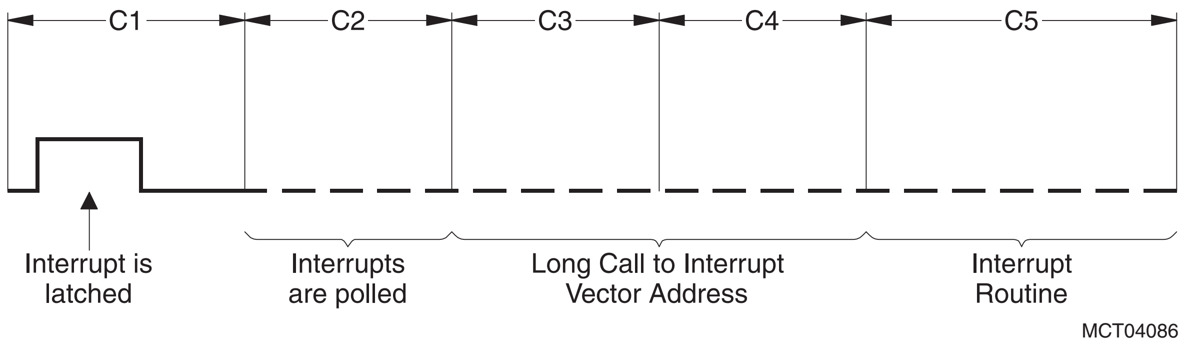


Figure 7-6 Interrupt Response Timing Diagram

Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in **Figure 7-6** then, in accordance with the above rules, it will be vectored to during C5 and C6 without any instruction for the lower priority routine to be executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not; then this must be done by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored to, as shown in the following **Table 7-3**.

Table 7-3 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	$002B_H$	TF2
A/D Converter	0043_H	IADC
External Interrupt 2	$004B_H$	IEX2
External Interrupt 3	0053_H	IEX3
External Interrupt 4	$005B_H$	IEX4
External Interrupt 5	0063_H	IEX5
External Interrupt 6	$006B_H$	IEX6
CAPCOM Emergency Interrupt	0093_H	TRF/BCERR
Compare Timer 2 Interrupt	$009B_H$	CT2P
Capture/Compare Match Interrupt	$00A3_H$	CCxF/CCxF, x = 0 to 2
Compare Timer 1 Interrupt	$00AB_H$	CT1FP/CT1FC
External Interrupt 7	$00D3_H$	IEX7
External Interrupt 8	$00DB_H$	IEX8
External Interrupt 9	$00E3_H$	IEX9
Wake-up from power-down mode	$007B_H$	–

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

7.5 External Interrupts

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

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

The external interrupts 2, 3, 7, 8, and 9 can be programmed to be either negative or positive transition-activated by setting or clearing bits I2FR or I3FR in register T2CON or bits I7FR, I8FR or I9FR in register EINT. If IxFR = 0 (x = 2, 3, 7, 8, or 9) then the external interrupt x is negative transition-activated. If IxFR = 1, the external interrupt is triggered by a positive transition.

The external interrupts 4, 5, and 6 are activated only by a positive transition.

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

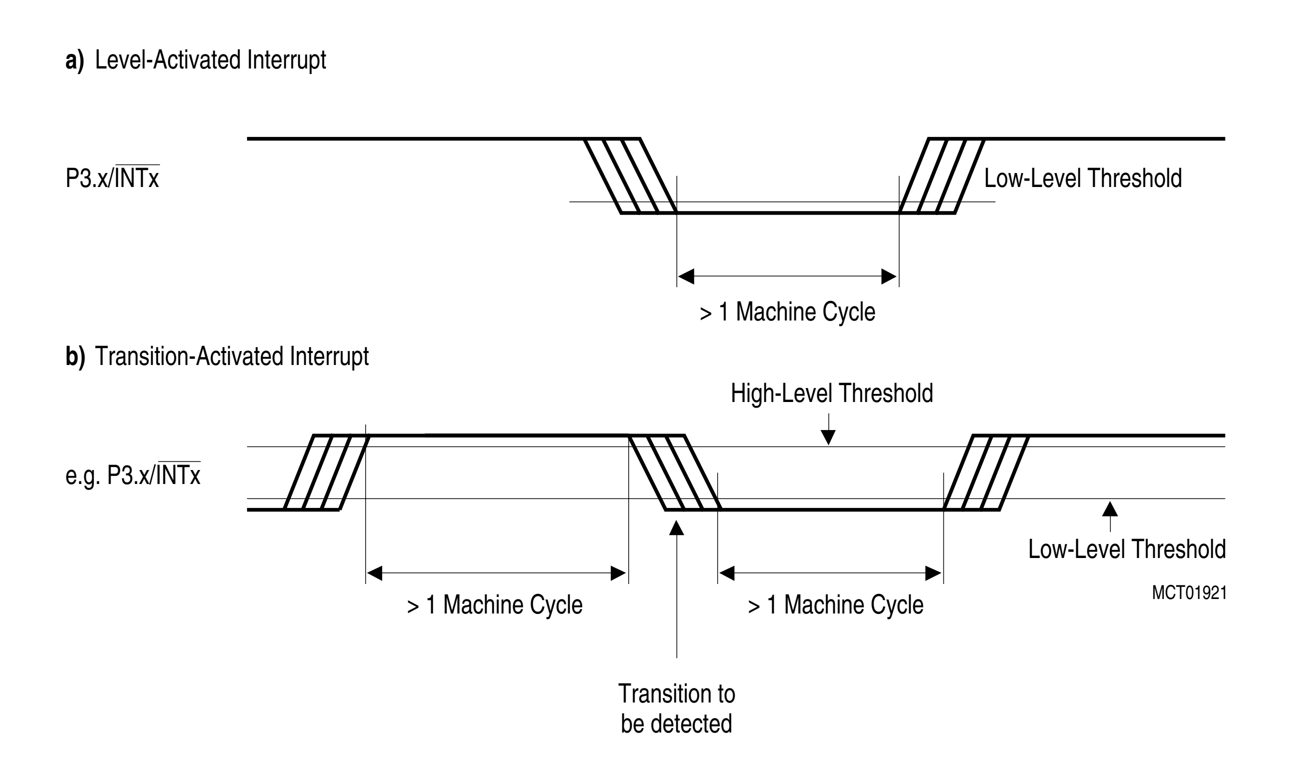


Figure 7-7 External Interrupt Detection

7.6 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 is 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 three cycles since the longest instructions (MUL and DIV) are only four cycles long; and, 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 five cycles (a maximum of one more cycle to complete the instruction in progress, plus four cycles to complete the next instruction, if the instruction is MUL or DIV).

Thus, in a single interrupt system, the response time is always more than three cycles and fewer than nine cycles.

8 Fail Save Mechanisms

The C508 offers enhanced fail save mechanisms which allow automatic recovery from software or hardware failure:

- A programmable Watchdog Timer (WDT) with variable time-out period from 153.6 µs to 314.573 ms at f_{OSC} = 10 MHz.
- An Oscillator Watchdog (OWD) which monitors the on-chip oscillator and forces the microcontroller into reset state if the on-chip oscillator fails. It also provides the clock for a fast internal reset after power-on.

8.1 Programmable Watchdog Timer

To protect the system against software failure, the user's program must clear this Watchdog Timer within a previously programmed time period. If the software fails to refresh the Watchdog Timer periodically, an internal 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 a hardware-related problem.

The Watchdog Timer in the C508 is a 15-bit timer which is incremented by a count rate of f_{OSC}/6 up to f_{OSC}/96. The machine clock of the C508 is divided by two prescalers. One is a divide-by-two prescaler; the other is a divide-by-16 prescaler. To program the Watchdog Timer overflow rate, the upper seven bits of the Watchdog Timer can be written. Figure 8-1 shows the block diagram of the Watchdog Timer unit.

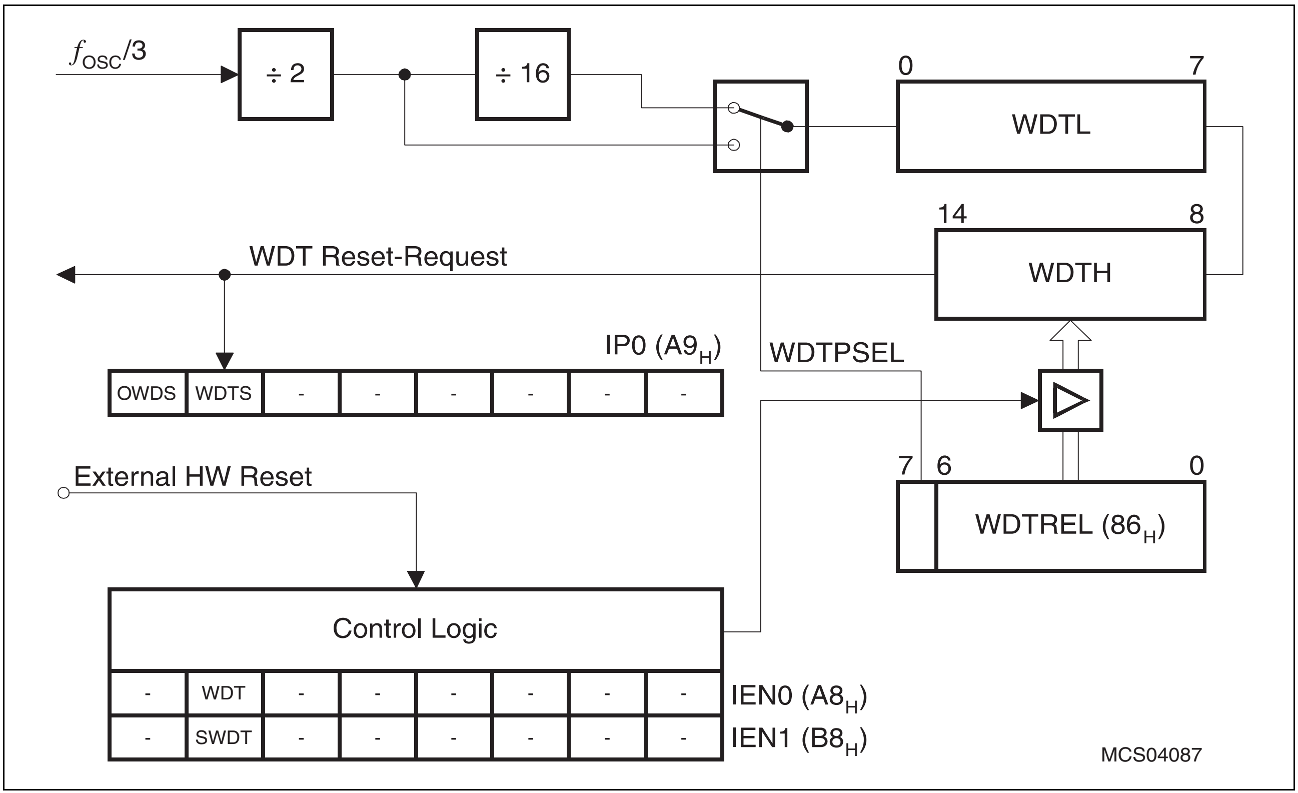


Figure 8-1 Block Diagram of the Programmable Watchdog Timer

SFRs WDTL and WDTH are read-only registers which hold the current watchdog timer value. They are write-protected in order to enhance the integrity of the watchdog timer as a fail safe mechanism. By reading these two registers, the current value of the watchdog timer can be obtained.

Special Function Register WDTH (Address 85_H) **Reset Value: $X0000000_B$**
Special Function Register WDTL (Address 84_H) **Reset Value: 00_H**

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
85_H	–	WDT Value, Upper 7 bits							WDTH
84_H	WDT Value, Low Byte								WDTL

Bit	Function
WDTH.6 - 0	Watchdog Timer value, upper 7 bits. Loaded with WDTREL.6 - 0 after a watchdog timer refresh.
WDTL.7 - 0	Watchdog Timer value, low byte. Reset to zero after a watchdog timer refresh.

8.1.1 Input Clock Selection

The input clock rate of the Watchdog Timer is derived from the system clock of the C508. 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 the resulting timeout periods at f_{OSC} = 5, 8, and 10 MHz.

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

	MSB							LSB	
Bit No.	7	6	5	4	3	2	1	0	
86$_H$	WDT PSEL	Reload Value							WDTREL

Bit	Function
WDTPSEL	**Watchdog Timer Prescaler Select bit** The Watchdog Timer is clocked through an additional divide-by-16 prescaler when this bit is set.
WDTREL.6 - 0	**Watchdog Timer Reload** 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 the WDT and SWDT bits.

Table 8-1 Watchdog Timer Time-Out Periods

WDTREL	Time-Out Period			Comments
	f_{OSC} = 5 MHz	f_{OSC} = 8 MHz	f_{OSC} = 10 MHz	
00$_H$	39.322 ms	24.576 ms	19.668 ms	This is the default value
80$_H$	629.146 ms	393.2 ms	314.573 ms	Maximum time period
7F$_H$	307.2 µs	192 µs	153.6 µs	Minimum time period

8.1.2 Watchdog Timer Control/Status Flags

The Watchdog Timer is controlled by two control flags (located in SFR IEN0 and IEN1) and one status flag (located in SFR IP0).

Special Function Register IEN0 (Address $A8_H$) **Reset Value: 00_H**
Special Function Register IEN1 (Address $B8_H$) **Reset Value: $X0000000_B$**
Special Function Register IP0 (Address $A9_H$) **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
Bit No.	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	–	SWDT	EX6	EX5	EX4	EX3	EX2	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 fail save control.

Bit	Function
WDT	**Watchdog Timer refresh flag** Set to initiate a refresh of the Watchdog Timer. Must be set before SWDT is set to prevent an unintentional refresh of the Watchdog Timer.
SWDT	**Watchdog Timer Start flag** Set to activate the Watchdog Timer. If set after WDT has been set, 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. Register WDTREL is cleared to 00_H after an external HW reset, an Oscillator Watchdog power-on reset, or a Watchdog Timer reset. The lower seven bits of WDTREL can be loaded by software at any time.

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 while the device is in active mode. An internal reset will be initiated if the software fails to clear the Watchdog Timer. The cause of the reset (either an external reset or a 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**). It is not possible to use the idle mode in combination with the Watchdog Timer function. Therefore, even the Watchdog Timer cannot reset the device if one of the power saving modes has been entered accidentally.

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. After the Watchdog has started, it 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 automatically be cleared during the second machine cycle after having been set. For this reason, setting SWDT bit must be a one cycle instruction (for example, SETB SWDT). This double-instruction refresh of the Watchdog Timer is implemented to minimize the chance of an unintentional reset of the Watchdog.

The reload register WDTREL can be written to at any time, as mentioned previously. Therefore, a periodic refresh of WDTREL can be added to the above mentioned starting procedure of the Watchdog Timer. Thus, a wrong reload value, caused by a possible distortion during the write operation to 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 reset 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 C508 and the function of the Watchdog Status flags. The WDTS flag is a flip-flop which is set by a Watchdog Timer reset and cleared by an external HW reset. Bit WDTS allows the software to examine from which source the reset was activated. The Watchdog Timer Status flag can also be cleared by software.

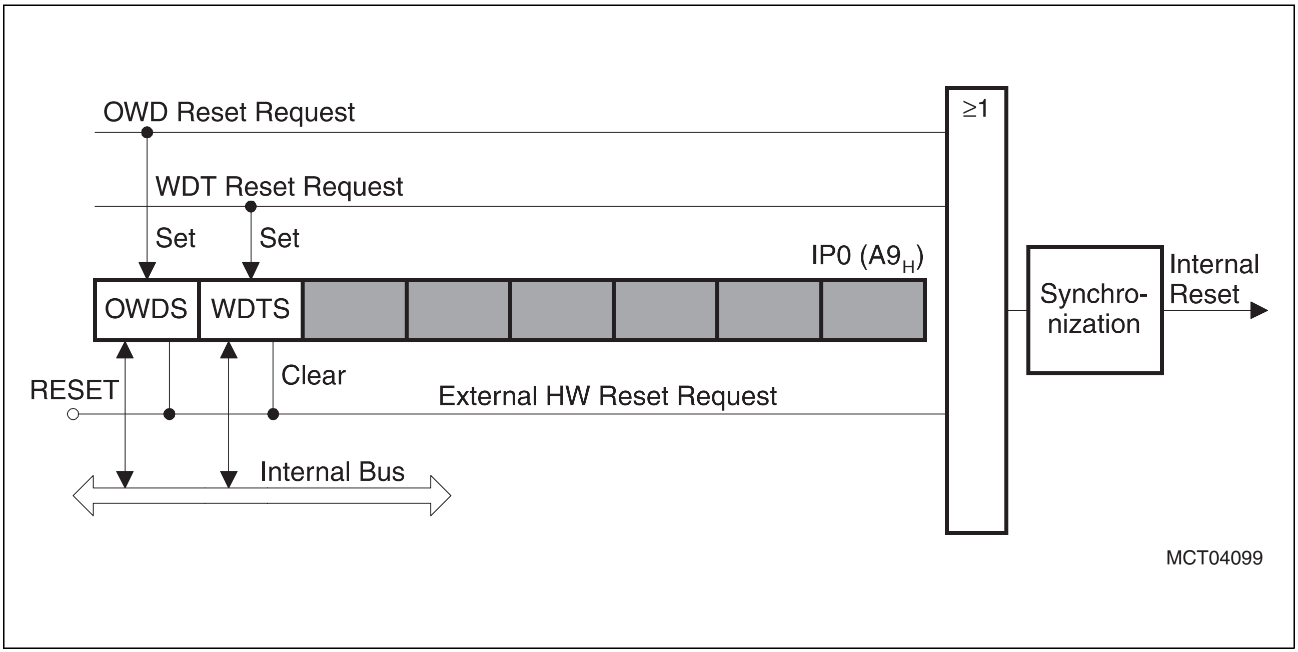


Figure 8-2 Watchdog Timer Status Flags and Reset Requests

8.2 Oscillator Watchdog Unit

The Oscillator Watchdog unit serves for 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 (that is, the on-chip oscillator has a higher frequency than the RC oscillator), the part executes a final reset phase of typically 1 ms to allow the oscillator to stabilize; then, the Oscillator Watchdog reset is released and the part starts program execution again.
- **Fast internal reset after power-on**
  The Oscillator Watchdog unit provides a clock supply for the reset before the on-chip oscillator and the PLL have started. This is described in **Chapter 5.2**.
- **Control of external wake-up from software power-down mode**
  When the software power-down mode is terminated by a low level at pins P3.2/$\overline{\text{INT0}}$ or P5.7/INT7, the Oscillator Watchdog unit ensures that the microcontroller resumes operation (execution of the power-down wake-up interrupt) with the nominal clock rate. The RC oscillator, the on-chip oscillator, and the PLL are stopped in power-down mode. They are started again when power-down mode is terminated. After the on-chip oscillator is stable and the PLL has been locked, the microcontroller starts program execution.

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 save control.

Bit	Function
OWDS	**Oscillator Watchdog Status flag** Set by hardware when an Oscillator Watchdog reset occurrs. Can be set and cleared by software.

8.2.1 Detailed Description of the Oscillator Watchdog Unit

Figure 8-3 shows the block diagram of the Oscillator Watchdog unit. It consists of an internal RC oscillator which provides the reference frequency for comparison with the frequency of the on-chip oscillator. It also shows the additional provisions for integration of wake-up from power-down mode.

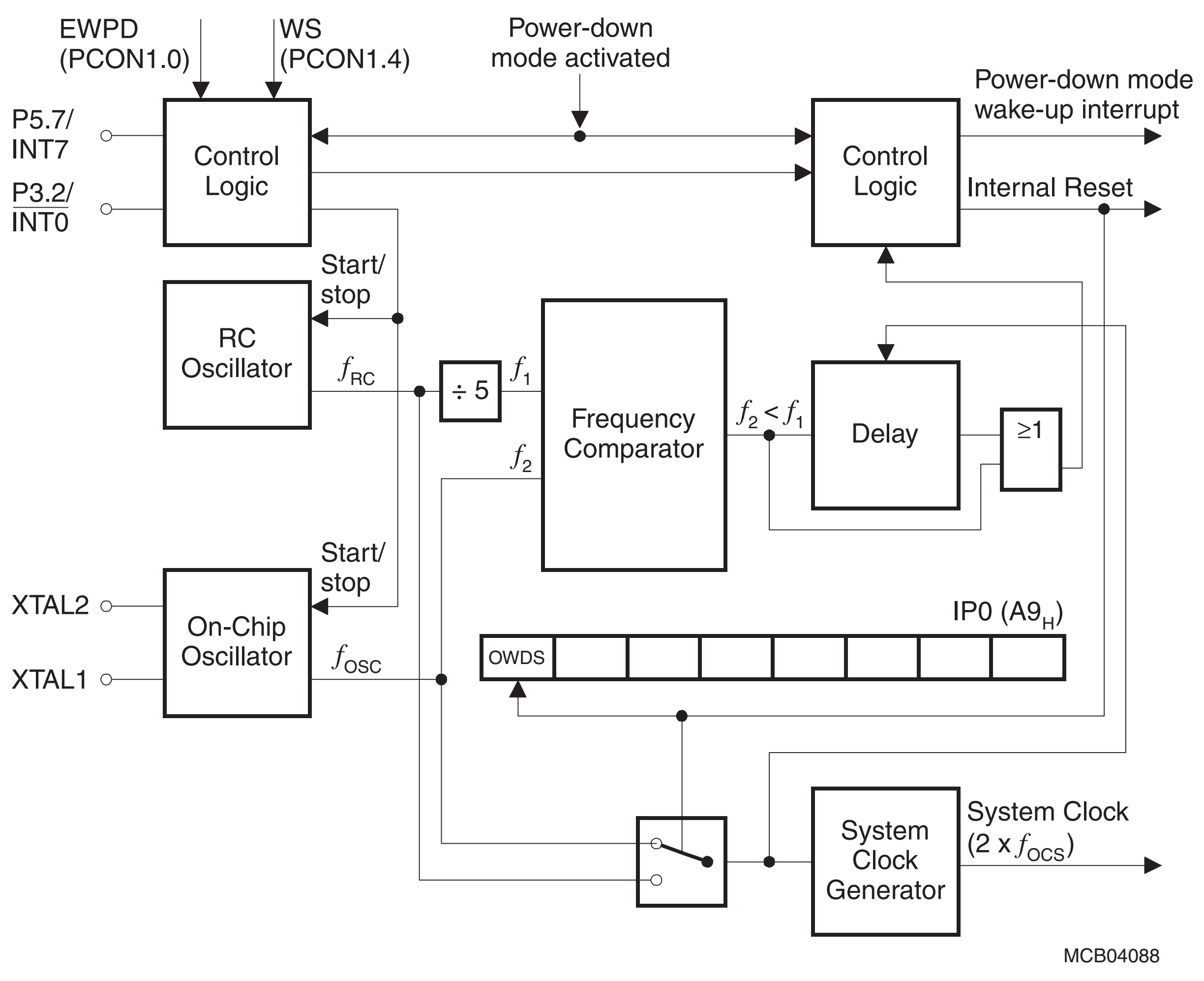


Figure 8-3 Functional Block Diagram of the Oscillator Watchdog

The frequency from the RC oscillator is divided by 5 and compared to the on-chip oscillator's frequency. If the frequency from the on-chip oscillator is found to be lower than the frequency derived from the RC oscillator, the Watchdog detects a failure condition. In this case, the RC oscillator provides the clock source for system clock generation. This means that the part is being clocked even if the on-chip oscillator has stopped or has not yet started. At the same time, the Watchdog activates the internal reset to bring the part into its defined reset state. The reset is performed because a clock is available from the RC oscillator. This internal Oscillator Watchdog reset has the same effect as an externally applied reset signal with the following exceptions: The Watchdog Timer Status flag WDTS is not reset (the Watchdog Timer is, however, stopped); and bit OWDS is set. This allows the software to examine error conditions detected by the Oscillator Watchdog unit even if an oscillator failure occurred in the meantime.

If the frequency derived from the on-chip oscillator is again higher than the reference, the Oscillator Watchdog starts a final reset sequence which takes typically 1 ms. Within that time, the system clock is still supplied by the RC oscillator and the part is held in reset. This allows a reliable stabilization of the on-chip oscillator. When this happens, the PLL will be locked and its clock output will be switched over as the system clock. After that, the Oscillator Watchdog releases its internal reset request. If no other reset is applied at this time, the part 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).

The RC oscillator, the on-chip oscillator, and the PLL are stopped if software power-down mode is activated. Both oscillators and the PLL are again started in power-down mode when a low level is detected at either P3.2/$\overline{\text{INT0}}$ or P5.7/INT7 and when bit EWPD in SFR PCON1 is set (wake-up from power-down mode enabled). Bit WS in SFR PCON1 selects the wake-up source. In this case, the Oscillator Watchdog does not execute an internal reset during startup of the on-chip oscillator. After the startup 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 **Chapter 9.4.2**.

8.2.2 Fast Internal Reset after Power-On

The C508 can use the Oscillator Watchdog unit for a fast internal reset procedure after power-on.

Normally, the members of the 8051 family (for example: 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 to bring the device into the correct reset state. The startup time of the oscillator can be relatively long (typ. 10 ms), especially if a crystal is used. During this time period, the pins have an undefined state which could have severe effect on such things as actuators connected to port pins.

The Oscillator Watchdog unit avoids this situation in the C508. After power-on, the oscillator Watchdog's RC oscillator starts working within a very short startup time (typ. less than 2 µs). The Watchdog circuitry then 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 gated PLL clock output, as described in the previous section). As long as this condition is valid, the Watchdog uses the RC oscillator output as the clock source for the chip. This allows the chip to be correctly reset and brings all ports to the defined state. The exception is Port 1, which will be at its default state when external reset is active (see also **Chapter 5** of this manual).

9 Power Saving Modes

The C508 provides two basic power saving modes: the idle mode and the power-down mode. Additionally, a slow down mode is available. This power saving mode reduces the internal clock rate in normal operating mode and it can also be used for further power reduction in idle mode.

9.1 Power Saving Mode Control Registers

The functions of the power saving modes are controlled by bits located in the Special Function Registers 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 ($B1_H$).

Bits PDE and PDS selects the power-down mode; while bits IDLE and IDLS selects the idle mode. These bits are all located in SFR PCON. If the power-down mode and the idle mode are set at the same time, power down mode takes precedence. Furthermore, register PCON contains two general purpose flags. For example, the flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during idle mode. For this, 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 1**

Symbol	Function
GF0	**General purpose flag 0**
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

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

Bit No.	MSB							LSB	
	7	6	5	4	3	2	1	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 the external wake-up from power down mode capability (more details see **Chapter 9.4.2**).

WS	**Wake-up from power-down source Select**
	WS = 0: wake-up via pin P3.2/$\overline{\text{INT0}}$.
	WS = 1: wake-up via pin P5.7/INT7.
	Pin P3.2/$\overline{\text{INT0}}$ is selected as the default wake-up source after reset.

–	Reserved bits for future use. Read by CPU returns undefined values.

9.2 Idle Mode

In the idle mode, the oscillator of the C508 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D Converter, the Capture/Compare Unit, and all timers (with the exception of the Watchdog Timer) are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption, which can be achieved by this feature, depends on the number of peripherals running. If all timers are stopped, and the A/D Converter and the serial interfaces are not running, the maximum power reduction can be achieved. This state is also the test condition for the idle mode I_{DD}.

Thus, the user must be cautious in determining which peripheral should continue to run and which must be stopped during idle mode. Also the state of all port pins – either the pins controlled by their latches or controlled by their secondary functions – depends on the status of the controller when entering idle mode.

Normally, the port pins hold the logical state they had at the time that the idle mode was activated. If some pins are programmed to serve as alternative functions, they still continue to output during idle mode if the assigned function is on. This especially applies to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\mathrm{PSEN}}$ 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 which makes it possible to “freeze” the processor’s status – either for a predefined time or until an external event reverts the controller to normal operation, as discussed below. The Watchdog Timer is the only peripheral which is automatically stopped during idle mode.

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

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

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

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

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enabled interrupt. 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 which 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 only for 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 (for example, if the controller is waiting for an input signal). In CMOS devices, there is an almost linear dependence of the operating frequency and the power supply current, so, a reduction of the operating frequency results in reduced power consumption.

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 system clock rate. The controller actually enters the slow down mode after a short synchronization period (maximum of 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. CPU operation is resumed, and the interrupt will be serviced. 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 stays enabled and if required termination must be done 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 power saving slow-down mode.
- The other possibility of terminating the combined idle and slow down 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.

9.4 Software Power Down Mode

In the software power-down mode, the RC oscillator, the on-chip oscillator which operates with the XTAL pins, and the PLL are all stopped. Therefore, all functions of the microcontroller are stopped and only the contents of the on-chip RAM, XRAM, and the SFRs are maintained. The port pins, which are controlled by their port latches, output the values that are held by their SFRs. The port pins which serve the alternative 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**).

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

The software power-down mode can be left either by an active reset signal or by a low signal at one of the wake-up source pins. Using reset to leave power-down mode puts the microcontroller with its SFRs into the reset state. Using either the P3.2/$\overline{\text{INT0}}$ pin or the P5.7/INT7 pin to exit power-down mode starts the RC oscillator, the on-chip oscillator, and the PLL; and maintains the state of the SFRs, which have been frozen when power-down mode was entered. Leaving power-down mode should not be done before V_{DD} is restored to its nominal operating level.

9.4.1 Invoking Software Power Down Mode

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

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

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

The instruction that sets bit PDS is the last instruction executed before going into power-down mode. When the double instruction sequence shown above is used, the power-down mode can only be left by a reset operation. If the external wake-up from power-down capability must also be used, its function must be enabled using the following instruction sequence prior to executing the double instruction sequence shown above.

```
ORL   SYSCON,#00010000B ;set RMAP
ORL   PCON1,#80H        ;enable wake-up from power down via P3.2/INT0
ANL   SYSCON,#11101111B ;reset RMAP (for future SFR accesses)
```

Setting EWPD automatically disables all interrupts still maintaining the actual values of the interrupt enable bits. In the above sequence, the value of register PCON1 should be modified for choosing a wake-up via the P5.7/INT7 (bit PCON1.4 should be set).

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

9.4.2 Exit from Software Power Down Mode

If power-down mode is left via a hardware reset, the microcontroller with its SFRs is put into the hardware reset state and the contents of RAM and XRAM are not changed. The reset signal that terminates the power-down mode also restarts the RC oscillator, the on-chip oscillator, and the PLL. 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).

Figure 9-1 shows the procedure which must be executed when power-down mode is left via the P3.2/$\overline{\text{INT0}}$ or the P5.7/INT7 wake-up capability.

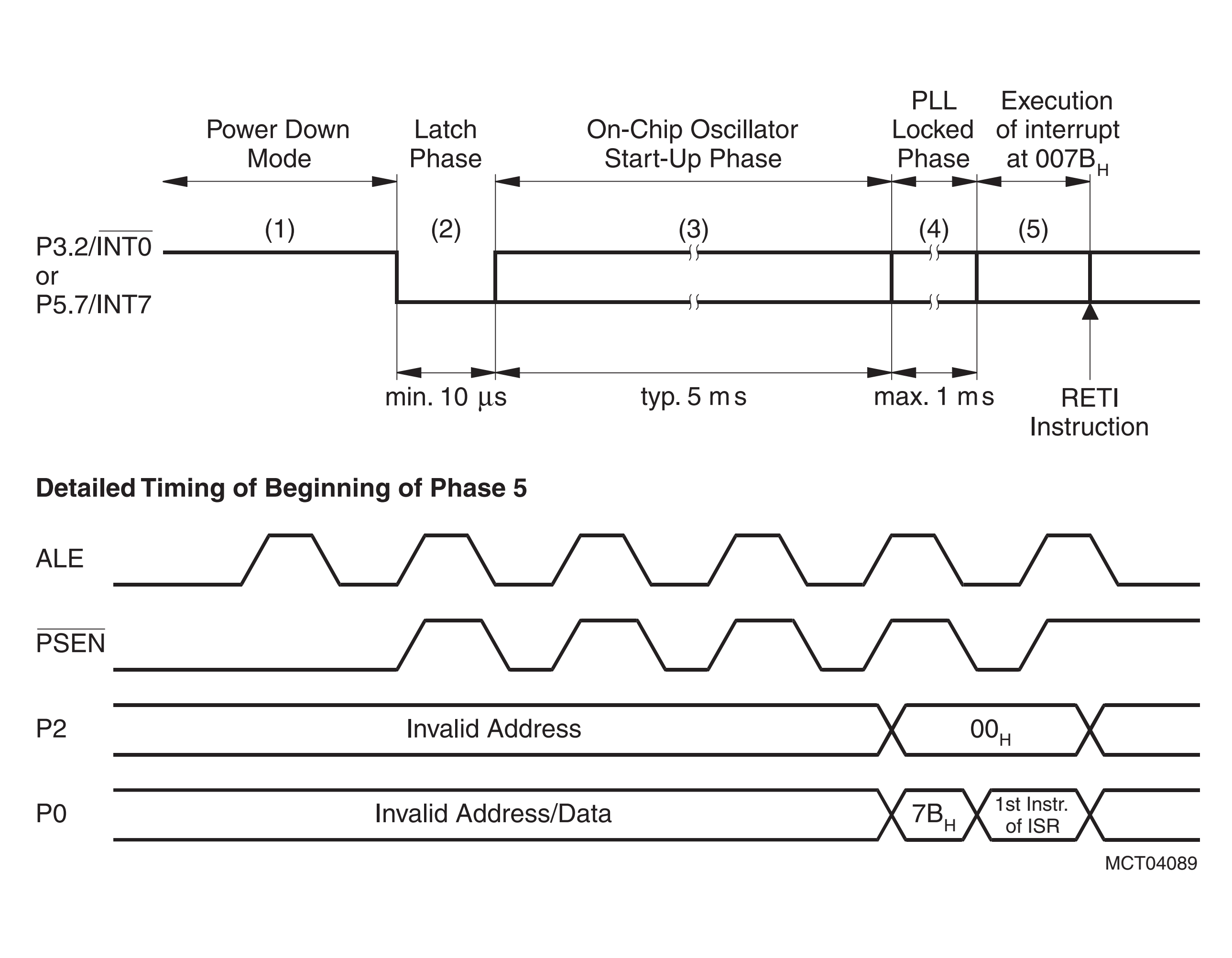


Figure 9-1 Wake-up from Power Down Mode Procedure

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 left via P3.2/INT0 while executing the following procedure:

1. In power-down mode, pin P3.2/INT0 must be held at high level.
2. Power-down mode is left when P3.2/INT0 goes low for at least 10 µs (latch phase). The internal RC oscillator, the on-chip oscillator, and the PLL are started; the state of pin P3.2/INT0 is internally latched; and P3.2/INT0 can be set again to high level if required, after this delay. Thereafter, the Oscillator Watchdog unit controls the wake-up procedure in its start-up phase.
3. The Oscillator Watchdog unit starts operation. Typically, the on-chip oscillator takes about 5 ms to stabilize.
4. The PLL will be locked within 1ms after the on-chip oscillator clock is detected for stable nominal frequency. Subsequently, the microcontroller starts again to initiate the power down wake-up interrupt. The interrupt address of the first instruction to be executed after wake-up is 007B$_H$. ALE and PSEN are in their power-down state up to this time. At the end of Phase 4, the CPU processes the interrupt call and, during these

two machine cycles, ALE and $\overline{\text{PSEN}}$ behave as shown in **Figure 9-1** (that is, at the beginning of Phase 5). Instruction fetches during the interrupt call are, however, discarded.

5. After the RETI instruction of the power-down wake-up interrupt routine has been executed, the instruction which follows the double instruction sequence to initiate the power-down mode will be executed. The functionality of the peripheral units timer 0/1/2, Capture/Compare Unit, and WDT are frozen until end of Phase 5.

All interrupts of the C508 are disabled from Phase 2 until the end of Phase 5. Other Interrupts can be first handled after the RETI instruction of the wake-up interrupt routine.

The procedure to exit the software power-down mode via the P5.7/INT7 pin is identical to the above procedure except that in this case pin P5.7/INT7 replaces pin P3.2/$\overline{\text{INT0}}$, and bit WS in SFR PCON1 should be set prior to entering software power-down mode.

9.5 State of Pins in Software Initiated Power Saving Modes

In the idle mode and in the power-down mode, the status of port pins of the C508 is well defined status. They are listed in **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

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, 5	Data / alternate outputs	Data / last output	Data / alternate outputs	Data / last output

10 OTP Memory Operation

The C508-4E is the OTP version in the C508 microcontroller with a 32 Kbyte One-Time Programmable (OTP) program memory. Fast programming cycles are achieved (1 byte in 100 µs) with the C508-4E. Several levels of OTP memory protection can be selected as well.

10.1 Programming Configuration

During normal program execution, the C508-4E behaves like the C508-4R, which has 32 Kbyte of on-chip ROM. To program the device, the C508-4E must be put into the programming mode. Typically, this is not done in-system but using special programming hardware. In the programming mode, the C508-4E operates as a slave device similar to an EPROM standalone 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 bi-directional data lines and Port 2 is used for the multiplexed address inputs. The upper address information at Port 2 is latched with the signal PALE. For basic programming mode selection, the inputs RESET, $\overline{\text{PSEN}}$, $\overline{\text{EA}}/V_{PP}$, PALE and PMSEL1/0, and $\overline{\text{PSEL}}$ are used. Further, the inputs PMSEL1, 0 are required to select the access types (for example, program/verify data, write lock bits, etc.) in the programming mode. In programming mode, V_{DD}/V_{SS} and a clock signal at the XTAL pins must be applied to the C508-4E. 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 C508-4E required to control the OTP programming mode.

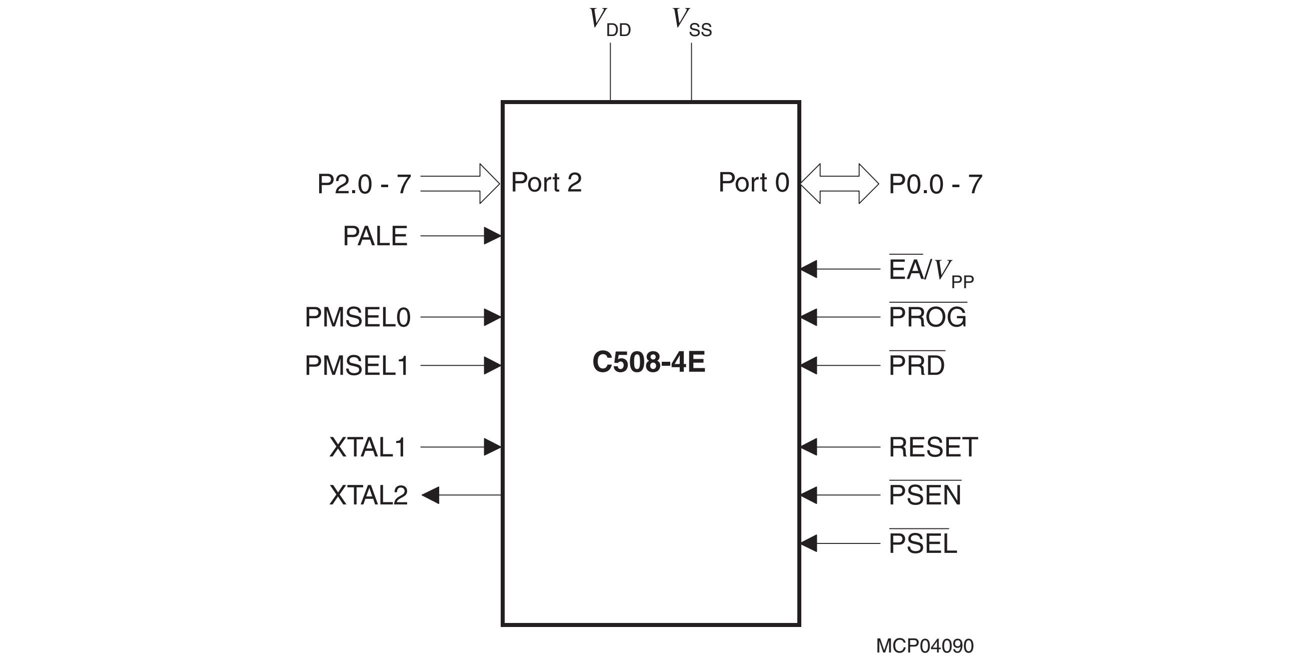


Figure 10-1 Programming Mode Configuration

10.2 Pin Configuration

Figure 10-2 shows the detailed pin configuration of the C508-4E device in programming mode for the P-MQFP-64-1 package.

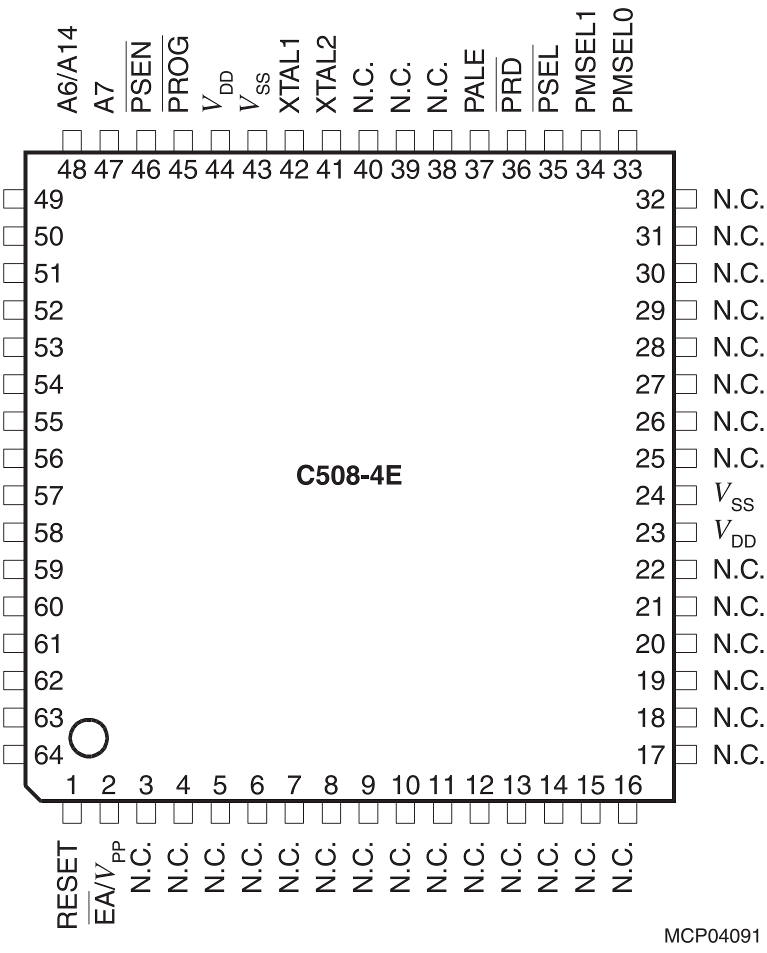


Figure 10-2 OTP Programming Mode Pin Configuration for P-MQFP-64-1 Package (top view)

Figure 10-3 shows the detailed pin configuration of the C508-4E device in programming mode for P-SDIP-64-2 package.

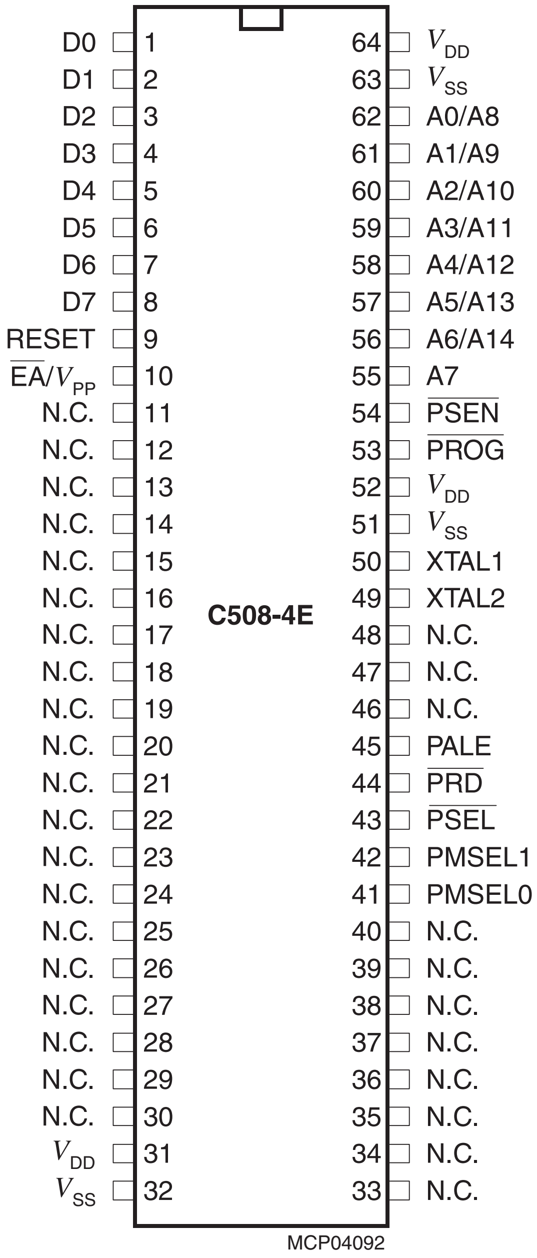


Figure 10-3 OTP Programming Mode Pin Configuration for P-SDIP-64-2 Package (top view)

10.3 Pin Definitions

Table 10-1 contains the functional descriptions of all C508-4E pins which are required for OTP memory programming.

Table 10-1 Pin Definitions and Functions of the C508-4E in Programming Mode

<table>
<tr><th rowspan="2">Symbol</th><th colspan="2">Pin Number</th><th rowspan="2">I/O[1]</th><th rowspan="2">Function</th></tr>
<tr><th>P-MQFP-64-1</th><th>P-SDIP-64-2</th></tr>
<tr><td>RESET</td><td>1</td><td>9</td><td>I</td><td><b>Reset</b>
This input must be at static ‘1’ (active) level throughout the entire programming mode.</td></tr>
<tr><td>PMSEL0
PMSEL1</td><td>33
34</td><td>41
42</td><td>I
I</td><td><b>Programming mode selection pins</b>
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>35</td><td>43</td><td>I</td><td><b>Basic programming mode select</b>
This input is used for the basic programming mode selection and must be switched according to Figure 10-4.</td></tr>
<tr><td>$\overline{\text{PRD}}$</td><td>36</td><td>44</td><td>I</td><td><b>Programming mode read strobe</b>
This input is used for read access control for OTP memory read, version byte read, and lock bit read operations.</td></tr>
</table>

Table 10-1 Pin Definitions and Functions of the C508-4E in Programming Mode (cont'd)

Symbol	Pin Number		I/O[1]	Function
	P-MQFP-64-1	P-SDIP-64-2		
PALE	37	45	I	**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 low level when the logic level of PMSEL1, 0 is changed.
XTAL2	47	49	O	**XTAL2** Output of the inverting oscillator amplifier.
XTAL1	48	50	I	**XTAL1** Input to the oscillator amplifier.
V_{SS}	24, 43, 55	32, 51, 63	–	**Ground** (0 V) must be applied in programming mode.
V_{DD}	23, 44, 56	31, 52, 64	–	**Power Supply** (+5 V) must be applied in programming mode.
P2.0-P2.7	54 - 47	62 - 55	I	**Address lines** P2.0-P2.7 are used as multiplexed address input lines A0-A7 and A8-A14. A8-A14 must be latched with PALE.
$\overline{\text{PSEN}}$	46	54	I	**Program store enable** This input must be at static 0 level during the whole programming mode.
$\overline{\text{PROG}}$	45	53	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 $\overline{\text{PROG}}$.

Table 10-1 Pin Definitions and Functions of the C508-4E in Programming Mode (cont'd)

Symbol	Pin Number		I/O[1]	Function
	P-MQFP-64-1	**P-SDIP-64-2**		
$\overline{\text{EA}}/V_{PP}$	2	10	–	**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_{IH2} high level. This pin is also used for basic programming mode selection. For basic programming mode selection, a low level must be applied to $\overline{\text{EA}}/V_{PP}$.
P0.0-P0.7	57 - 64	1 - 8	I/O	**Data lines** In programming mode, data bytes are transferred via the bi-directional D0-D7 lines which are located at Port 0.
N.C.	3 - 12, 15 - 22, 25 - 32, 38 - 40	11 - 30, 33 - 40, 46 - 48	–	**Not Connected** These pins should not be connected in programming mode.

[1] I = Input
O = Output

10.4 Programming Mode Selection

The selection for 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. Further, after selection of the basic programming mode, OTP memory accesses are executed by using one of the access modes. These access modes are 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 10-4**.

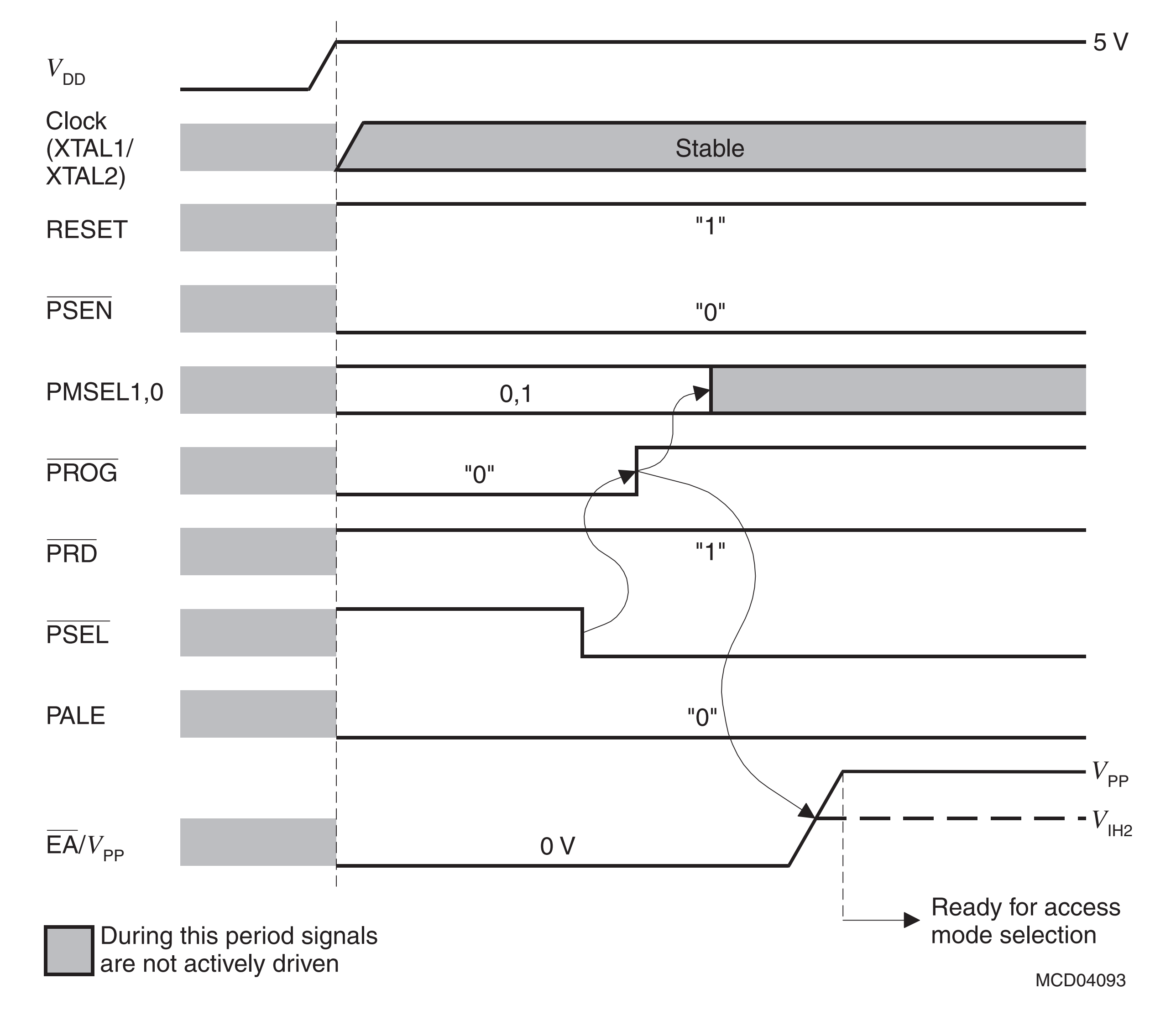


Figure 10-4 Basic Programming Mode Selection

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’.
- $\overline{\text{PROG}}$, 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’.
- $\overline{\text{PSEL}}$ is switched from ‘1’ to ‘0’ level and thereafter $\overline{\text{PROG}}$ is switched to ‘1’.
- PMSEL1, 0 can now be changed; after $\overline{\text{EA}}/V_{PP}$ has been set to V_{IH2} 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 throughout the entire programming mode. With a falling edge of $\overline{\text{PSEL}}$, the logic state of $\overline{\text{PROG}}$ and $\overline{\text{EA}}/V_{PP}$ are internally latched. These two signals are now used as programming write pulse signal ($\overline{\text{PROG}}$) and as programming voltage 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 **Chapter 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 C508-4E 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 10-2**.

Table 10-2 Access Modes Selection

Access Mode	$\overline{\text{EA}}/V_{PP}$	PROG	PRD	PMSEL		Address (Port 2)	Data (Port 0)
				1	0		
Program OTP memory byte	V_{PP}	⊔	H	H	H	A0-A7 A8-A14	D0-D7
Read OTP memory byte	V_{IH}	H	⊔				
Program OTP lock bits	V_{PP}	⊔	H	H	L	–	D1, D0 see **Table 10-3**
Read OTP lock bits	V_{IH}	H	⊔				
Read OTP version byte	V_{IH}	H	⊔	L	H	Byte addr. of version byte	D0-D7

The access modes from **Table 10-2** are basically 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 details of the different access modes.

10.5 Program / Read OTP Memory Bytes

The program/read OTP memory byte access mode is defined by PMSEL1, 0 = 1, 1. It is initiated when the PMSEL1, 0 = 1, 1 is valid at the rising edge of PALE. With the falling edge of PALE, the upper addresses A8-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 only be latched once (page address mechanism).

Figure 10-5 shows a typical basic OTP memory programming cycle with a following OTP memory read operation. In this example, A0-A14 of the read operation are identical to A8-A14 of the proceeding programming operation.

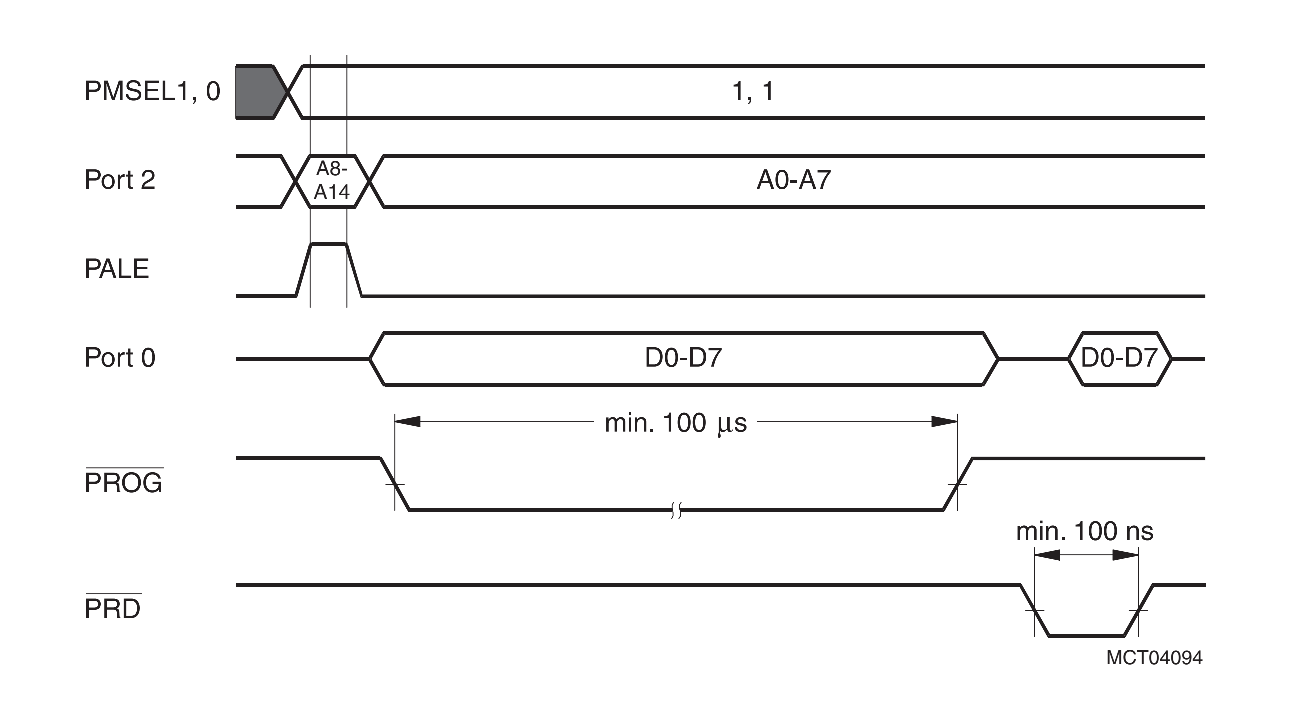


Figure 10-5 Programming / Verify OTP Memory Access Waveform

If the address lines A8-A14 must be updated, PALE must be activated for the latching of 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 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 10-6 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.

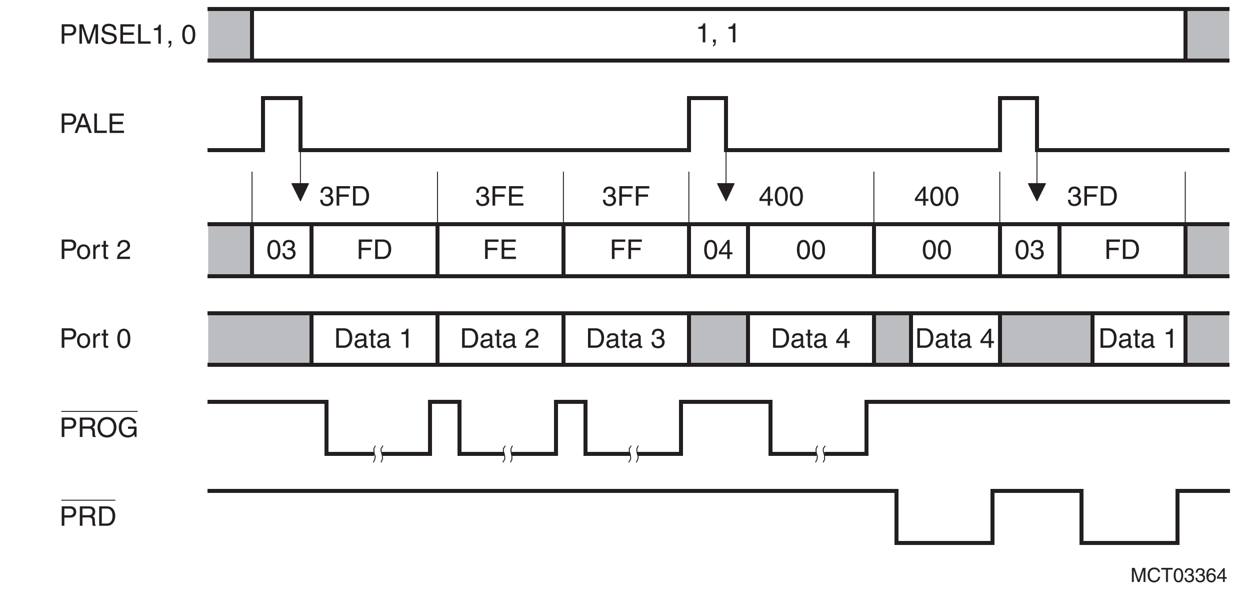


Figure 10-6 Typical OTP Memory Programming / Verify Access Waveform

10.6 Lock Bits Programming / Read

The C508-4E has two programmable lock bits which, when programmed according to Table 10-3, provide four levels of protection for the on-chip OTP code memory.

Table 10-3 Lock Bit Protection Types

Lock Bits at D1, D0		**Protection Level**	**Protection Type**
D1	**D0**		
1	1	Level 0	The OTP lock feature is disabled. During normal operation of the C508-4E, the state of the $\overline{\text{EA}}$ pin is not latched on reset.
1	0	Level 1	During normal operation of the C508-4E, MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{\text{EA}}$ is sampled and latched on reset. An OTP memory read operation is only possible according to OTP verification mode 2. Further programming of the OTP memory is disabled (reprogramming security).
0	1	Level 2	Same as Level 1, but also OTP memory read operation using OTP verification mode is disabled.
0	0	Level 3	Same as Level 2, but additionally external code execution by setting $\overline{\text{EA}}$ = low during normal operation of the C508-4E is no longer possible. External code execution, initiated by an internal program (for example, by an internal jump instruction above the OTP memory boundary), is still possible.

Note: '1' means that the lock bit is unprogrammed; '0' means that lock bit is programmed.

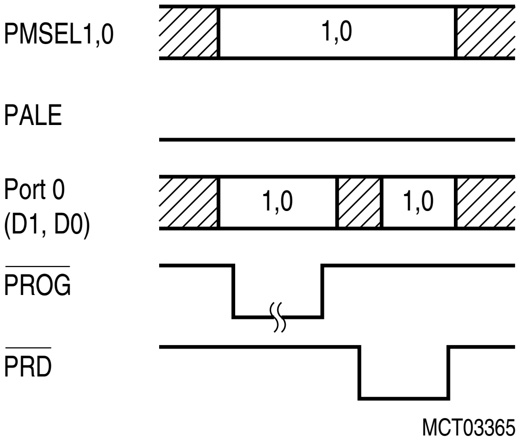
For a, OTP verify operation at protection Level 1, the C508-4E must be put into the OTP verification mode.

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

When a protection level has been activated by programming of the lock bits, the basic programming mode must be left for activation of the protection mechanisms. This means that after the activation of a protection level, further OTP program/verify operations are still possible if 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 left, it is not possible to re-enter the programming mode. In this case, the lock bits can no longer be read.

Figure 10-7 shows the waveform of a lock bit write/read access. For simplicity, the $\overline{\text{PROG}}$ pulse is shortened. In reality, a 100 µs $\overline{\text{PROG}}$ low pulse must be applied for lock bit programming.



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

Figure 10-7 Write/Read Lock Bit Waveform

10.7 Access of Version Bytes

The C508-4E and C508-4R provide three version bytes at address locations FC_H, FD_H, and FE_H. The information stored in the version bytes, is defined by the mask of each microcontroller step. Therefore, the version bytes can be read but cannot be written. The three Version Registers hold such information as manufacturer code, device type, and stepping code.

To read the version bytes, the control lines must be used in accordance with **Table 10-2** and **Figure 10-8**. The address of the version byte must be applied at the Port 2 address lines. PALE must not be activated.

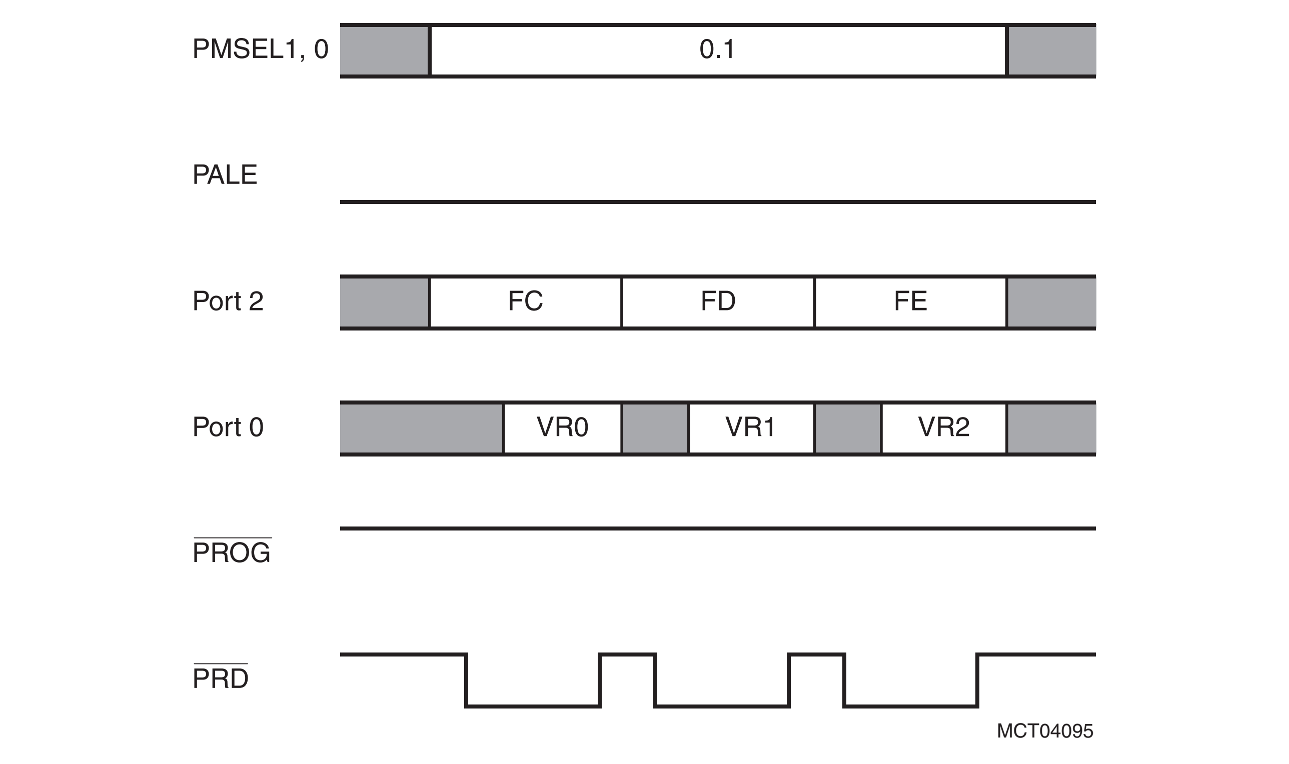


Figure 10-8 Read Version Register(s) Waveform

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

Note: The three version bytes are implemented in such a way that they can be also be read during normal program execution mode as a mapped register with bit RMAP in SFR SYSCON set. The addresses of the version bytes in normal mode and programming mode are identical and, therefore, they are located in the SFR address range.

11 Index

11.1 Keyword Index

This section lists a number of keywords which refer to specific details of the C508 in terms of its architecture, its functional units, or functions. Bold page number entries identify the main definition material for a topic.

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