SIEMENS

C515A

8-Bit CMOS Microcontroller

User's Manual 08.97

C515A User's Manual Revision History :		08.97
Previous Releases :		Original Version
Page (previous version)	Page (new version)	Subjects (changes since last revision)

Edition 08.97

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

Attention please!

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

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

Terms of delivery and rights to change design reserved.

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

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

Siemens AG is an approved CECC manufacturer.

Packing

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

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

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

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

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

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

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

1 Introduction

The C515A is a member of the Siemens C500 family of 8-bit microcontrollers. lt is functionally fully compatible with the SAB 80C515/83C515A-5 microcontrollers.

The C515A basically operates with internal and/or external program memory. The C515A-L is identical to the C515A-4R, except that it lacks the on-chip program memory. Therefore, in this documentation the term C515A refers to all versions within this specification unless otherwise noted.

Figure 1-1 shows the different functional units of the C515A and **figure 1-2** shows the simplified logic symbol of the C515A.

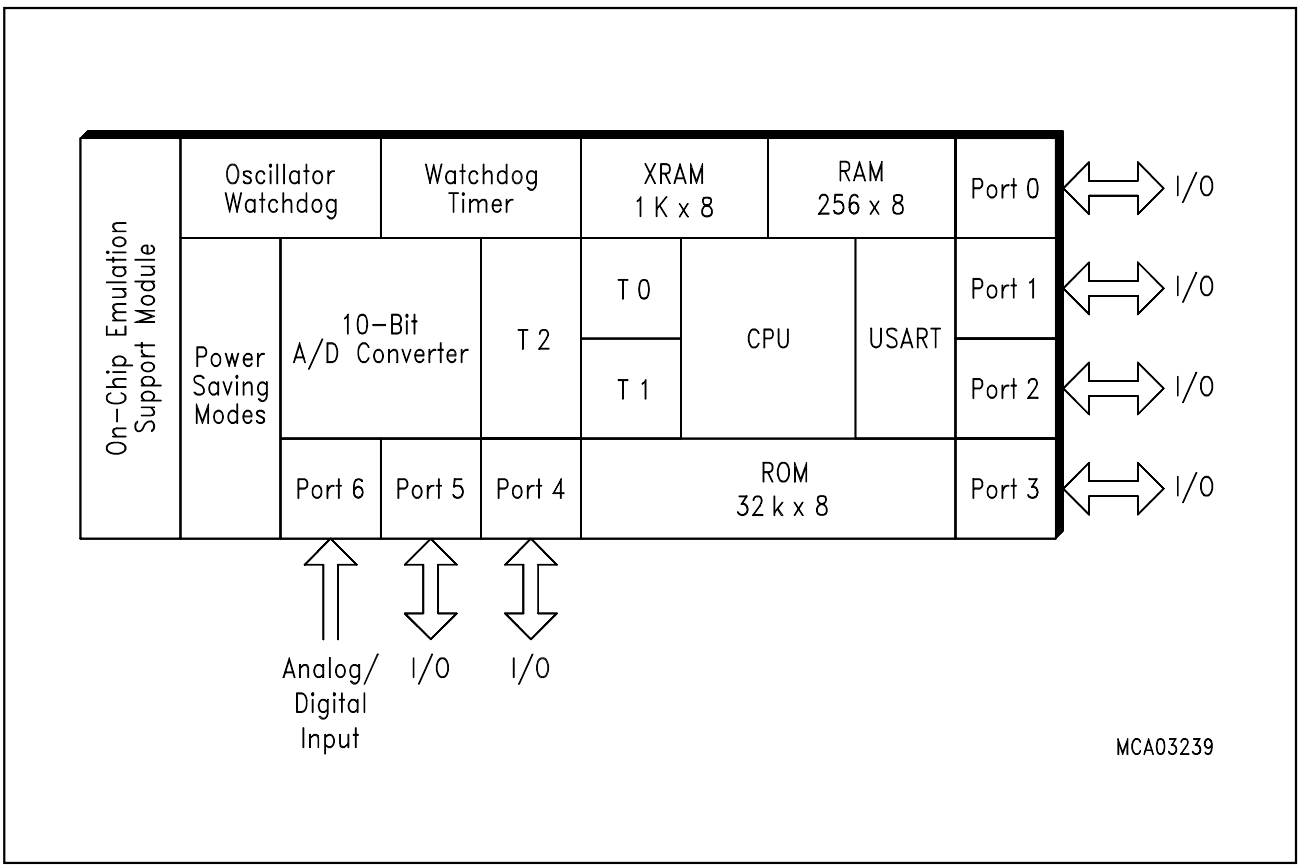


Figure 1-1
C515A Functional Units

Listed below is a summary of the main features of the C515A:

- Full upward compatibility with SAB 80C515A/83C515A-5
- Up to 24 MHz external operating frequency
  – 500 ns instruction cycle at 24 MHz operation
- 32K byte on-chip ROM (with optional ROM protection)
  – alternatively up to 64K byte external program memory
- Up to 64K byte external data memory
- 256 byte on-chip RAM
- 1K byte on-chip RAM (XRAM)
- On-chip emulation support logic (Enhanced Hooks Technology ™)
- Six 8-bit parallel I/O ports
- One input port for analog/digital input
- Full duplex serial interface (USART)
  – 4 operating modes, fixed or variabie baud rates
- Three 16-bit timer/counters
  – Timer 0 / 1 (C501 compatible)
  – Timer 2 for 16-bit reload, compare, or capture functions
- 10-bit A/D converter
  – 8 multiplexed analog inputs
  – Built-in self calibration
- 16-bit watchdog timer
- Power saving modes
  – Slow down mode
  – Idle mode (can be combined with slow down mode)
  – Software power down mode with wake-up capability through $\overline{\text{INT0}}$ pin
  – Hardware power down mode
- 12 interrupt sources (7 external, 5 internal) selectable at 4 priority levels
- ALE switch-off capability
- P-MQFP-80 packages
- Temperature Ranges : SAB-C515A-4R T_A = 0 to 70 °C
  SAF-C515A-4R T_A = – 40 to 85 °C
  SAH-C515A-4R T_A = – 40 to 110 °C
  SAK-C515A-4R T_A = – 40 to 125 °C (max. frequency : 18 MHz)

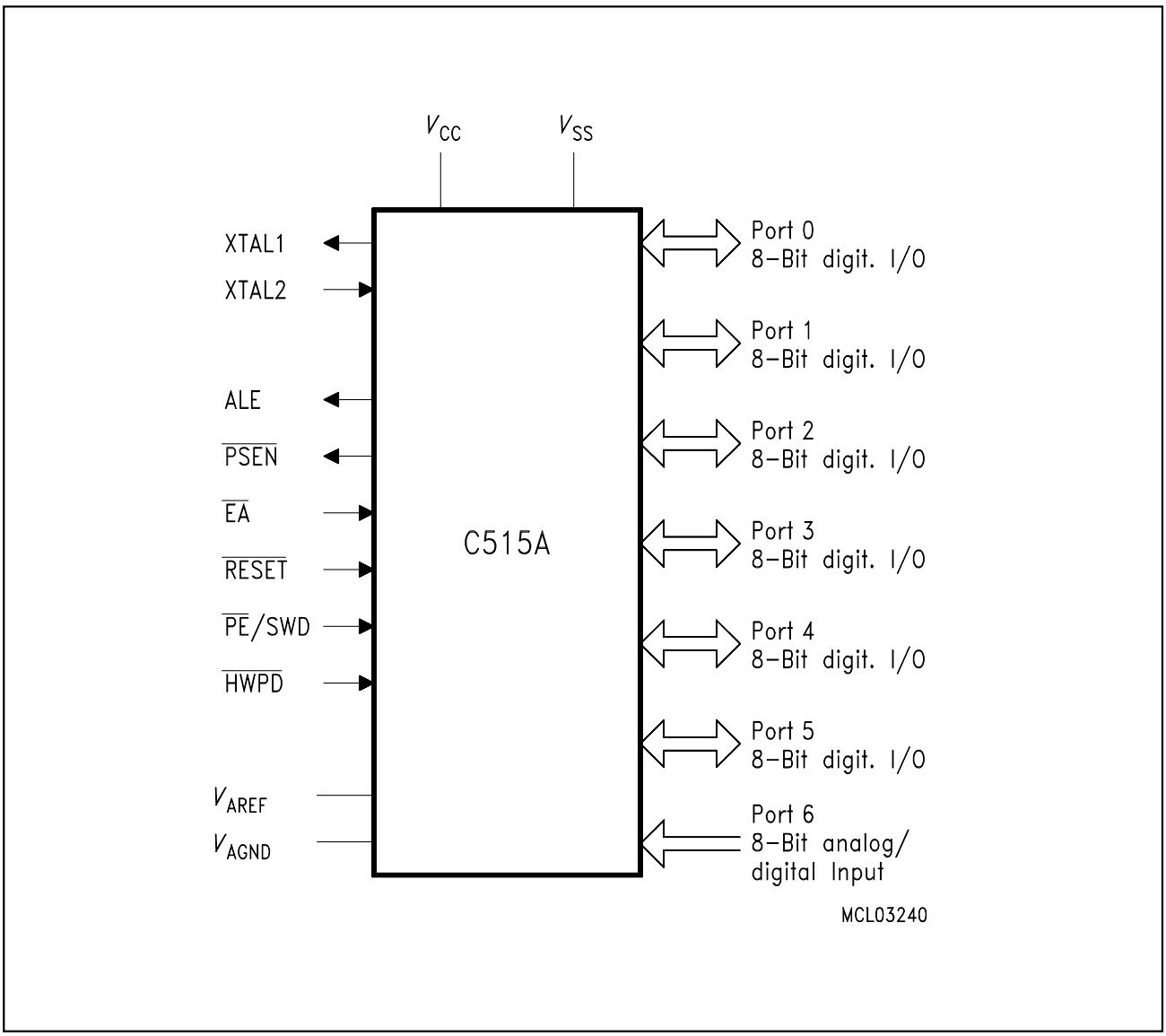


Figure 1-2
Logic Symbol

1.1 Pin Configuration

This section describes the pin configuration of the C515A in the P-MQFP-80 package.

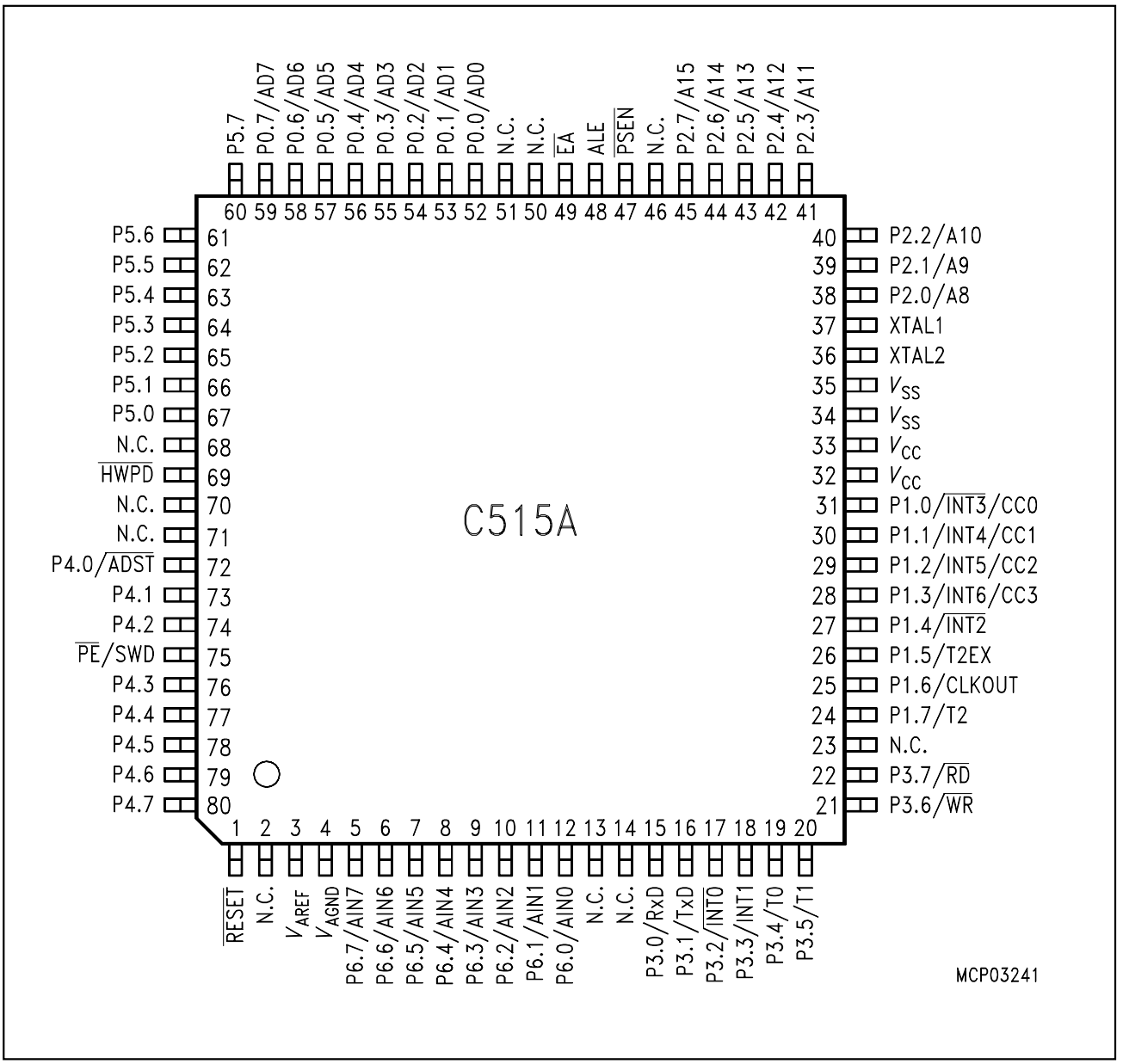


Figure 1-3
Pin Configuration P-MQFP-80 Package (top view)

1.2 Pin Definitions and Functions

This section describes all external signals of the C515A with its function.

Table 1-1
Pin Definitions and Functions

Symbol	Pin Number (P-MQFP-80)	I/O*)	Function
P4.0-P4.7	72-74, 76-80 72	I/O	**Port 4** is an 8-bit quasi-bidirectional I/O port with internal pull-up resistors. Port 4 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 4 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pull-up resistors. P4 also contains the external A/D converter control pin. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary function is assigned to port 4 as follows : P4.0 / $\overline{\text{ADST}}$ external A/D converter start pin
$\overline{\text{PE}}$/SWD	75	I	**$\overline{\text{Power Saving Mode Enable}}$ / Start Watchdog Timer** A low level on this pin allows the software to enter the power down, idle, and slow down mode. In case the low level is also seen during reset, the watchdog timer function is off on default. Use of the software controlled power saving modes is blocked when this pin is held on high level. A high level during reset performs an automatic start of the watchdog timer immediately after reset. When left unconnected this pin is pulled high by a weak internal pull-up resistor. Note : If $\overline{\text{PE}}$/SWD is low and V_{AREF} is low the oscillator watchdog is disabled (testmode) !
$\overline{\text{RESET}}$	1	I	**$\overline{\text{RESET}}$** A low level on this pin for the duration of two machine cycles while the oscillator is running resets the C515A. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
VAREF	3	–	**Reference Voltage** for the A/D converter
VAGND	4	–	**Reference Ground** for the A/D converter

*) I = Input
O = Output

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

Symbol	Pin Number (P-MQFP-80)	I/O*)	Function
P6.0-P6.7	12-5	I	**Port 6** is an 8-bit unidirectional input port to the A/D converter. Port 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.
P3.0-P3.7	15-22	I/O	**Port 3** is an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 3 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows:
	15		P3.0 / RxD — Receiver data input (asynch.) or data input/output (synch.) of serial interface
	16		P3.1 / TxD — Transmitter data output (asynch.) or clock output (synch.) of serial interface
	17		P3.2 / $\overline{INT0}$ — External interrupt 0 input / timer 0 gate control input
	18		P3.3 / $\overline{INT1}$ — External interrupt 1 input / timer 1 gate control input
	19		P3.4 / T0 — Timer 0 counter input
	20		P3.5 / T1 — Timer 1 counter input
	21		P3.6 / $\overline{WR}$ — $\overline{WR}$ control output; latches the data byte from port 0 into the external data memory
	22		P3.7 / $\overline{RD}$ — $\overline{RD}$ control output; enables the external data memory

*) I = Input
O = Output

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

Symbol	Pin Number (P-MQFP-80)	I/O*)	Function
P1.0 - P1.7	31-24	I/O	**Port 1** is an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. The port is used for the low-order address byte during program verification. Port 1 also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows :
	31		P1.0 / $\overline{\text{INT3}}$ / CC0 Interrupt 3 input / compare 0 output / capture 0 input
	30		P1.1 / INT4 / CC1 Interrupt 4 input / compare 1 output / capture 1 input
	29		P1.2 / INT5 / CC2 Interrupt 5 input / compare 2 output / capture 2 input
	28		P1.3 / INT6 / CC3 Interrupt 6 input / compare 3 output / capture 3 input
	27		P1.4 / $\overline{\text{INT2}}$ Interrupt 2 input
	26		P1.5 / T2EX Timer 2 external reload / trigger input
	25		P1.6 / CLKOUT System clock output
	24		P1.7 / T2 Counter 2 input
V_{CC}	32, 33	–	**Supply Voltage** during normal, idle, and power down mode.
V_{SS}	34, 35	–	**Ground** (0V) during normal, idle, and power down operation.

*) I = Input
O = Output

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

Symbol	Pin Number (P-MQFP-80)	I/O*)	Function
XTAL2	36	–	**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. Minimum and maximum high and low times as well as rise/fall times specified in the AC characteristics must be observed.
XTAL1	37	–	**XTAL1** Output of the inverting oscillator amplifier.
P2.0-P2.7	38-45	I/O	**Port 2** is an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullup resistors when issuing 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.
$\overline{\text{PSEN}}$	47	O	The **$\overline{\text{Program Store Enable}}$** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. The signal remains high during internal program execution.
ALE	48	O	The **Address Latch Enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods, except during an external data memory access. ALE can be switched off when the program is executed internally.

*) I = Input
O = Output

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

Symbol	Pin Number (P-MQFP-80)	I/O*)	Function
$\overline{EA}$	49	I	**$\overline{External\ Access\ Enable}$** When held high, the C515A executes instructions from the internal ROM (C515A-4R) as long as the PC is less than 8000_H. When held low, the C515A fetches all instructions from external program memory. For the C515A-L this pin must be tied low.
P0.0-P0.7	52-59	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 and data memory. In this application it uses strong internal pullup resistors when issuing 1's. Port 0 also outputs the code bytes during program verification in the C515A-4R. External pullup resistors are required during program verification.
P5.0-P5.7	67-60	I/O	**Port 5** is an 8-bit quasi-bidirectional I/O port with internal pullup resistors. Port 5 pins that have 1's written to them are pulled high by the internal pullup 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 pullup resistors.
$\overline{HWPD}$	69	I	**$\overline{Hardware\ Power\ Down}$** A low level on this pin for the duration of one machine cycle while the oscillator is running resets the C515A. A low level for a longer period will force the C515A into Hardware Power Down Mode with the pins floating.
N.C.	2, 13, 14, 23, 46, 50, 51, 68, 70, 71	–	**Not connected** These pins of the P-MQFP-80 package need not be connected.

*) I = Input
O = Output

2 Fundamental Structure

The C515A is fully compatible to the architecture of the standard 8051/C501 microcontroller family. While maintaining all architectural and operational characteristics of the C501, the C515A incorporates a 10-bit A/D converter, a timer 2 with capture/compare functions, an on-chip XRAM data memory, as well as some enhancements in the Fail Save Mechanism unit. **Figure 2-1** shows a block diagram of the C515A.

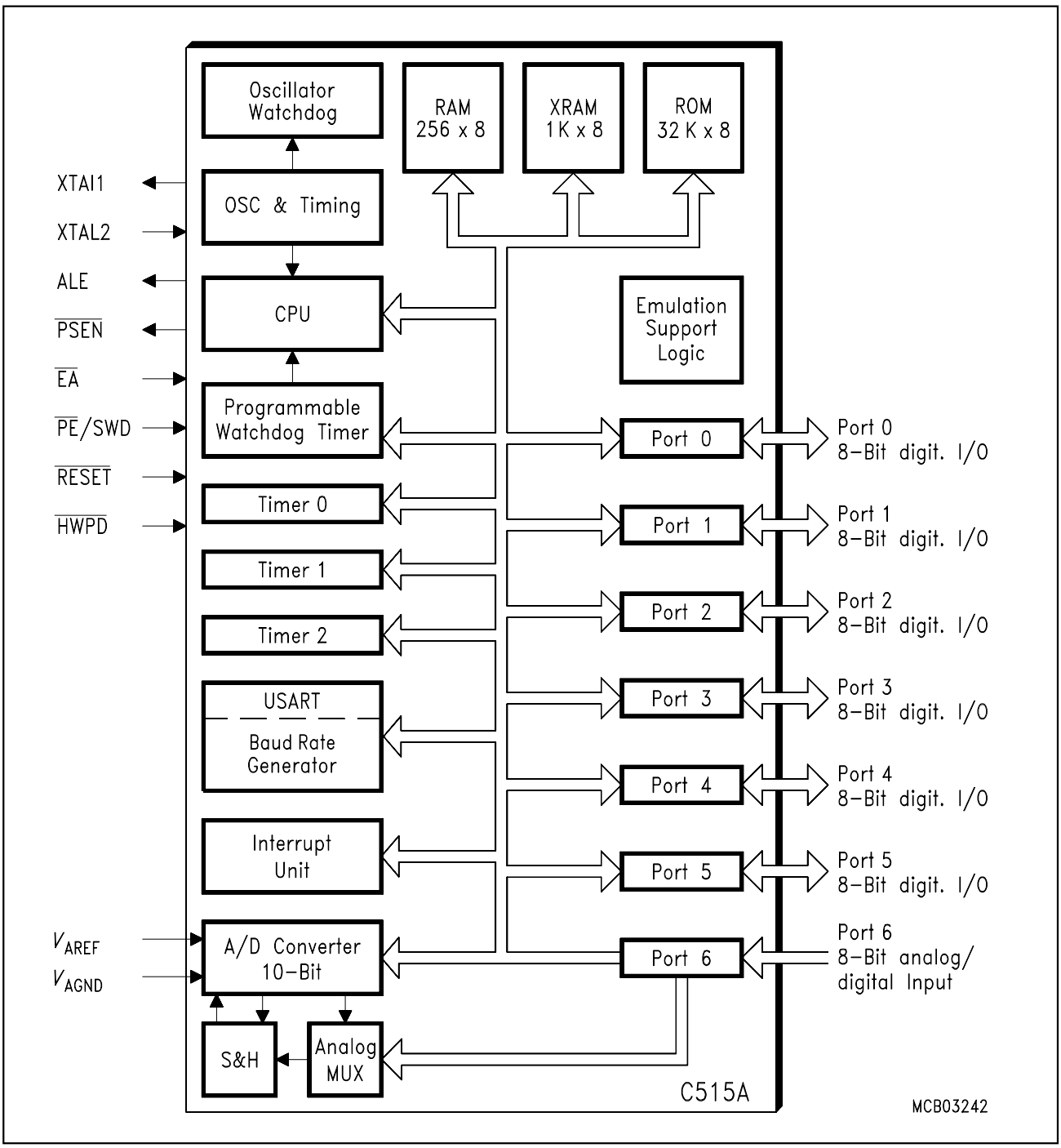


Figure 2-1
Block Diagram of the C515A

2.1 CPU

The C515A is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in its bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz external clock, 58% of the instructions execute in 1.0 µs (24 MHz 500 ns).

The CPU (Central Processing Unit) of the C515A consists of the instruction decoder, the arithmetic section and the program control section. Each program instruction is decoded by the instruction decoder. This unit generates the internal signals controlling the functions of the individual units within the CPU. They have an effect on the source and destination of data transfers and control the ALU processing.

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

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

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

Accumulator

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

Program Status Word

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

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

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

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

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

OV	Overflow Flag Used by arithmetic instruction.
F1	General Purpose Flag
P	Parity Flag Set/cleared by hardware after each instruction to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.

B Register

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

Stack Pointer

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

2.2 CPU Timing

A machine cycle of the C515A consists of 6 states (12 oscillator periods). Each state is devided into a phase 1 half and a phase 2 half. Thus, a machine cycle consists of 12 oscillator periods, numbererd S1P1 (state 1, phase 1) through S6P2 (state 6, phase 2). Each state lasts one oscillator period. Typically, arithmetic and logic operations take place during phase 1 and internal register-to-register transfers take place during phase 2.

The diagrams in **figure 2-2** show the fetch/execute timing related to the internal states and phases. Since these internal clock signals are not user-accessible, the XTAL1 oscillator signals and the ALE (address latch enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

Executing of a one-cycle instruction begins at S1P2, when the op-code is latched into the instruction register. If it is a two-byte instruction, the second reading takes place during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op-code) is ignored (discarded fetch), and the program counter is not incremented. In any case, execution is completed at the end of S6P2. **Figures 2-2 (a)** and **(b)** show the timing of a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most C515A instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete; they take four cycles. Normally two code bytes are fetched from the program memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed. **Figure 2-2 (c)** and **(d)** show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

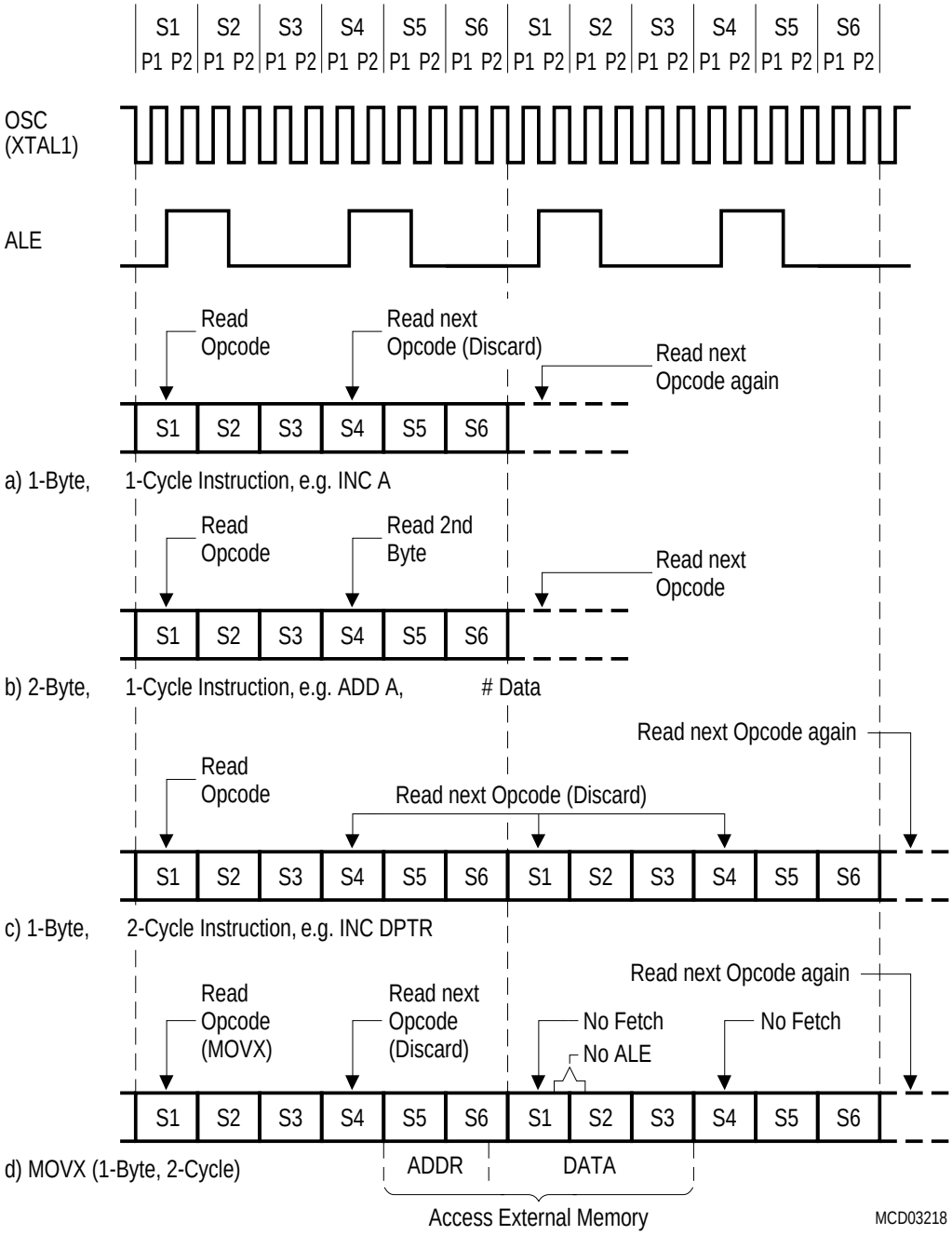


Figure 2-2
Fetch Execute Sequence

3 Memory Organization

The C515A CPU manipulates operands in the following five address spaces:

- up to 64 Kbyte of program memory (32K on-chip program memory for C515A-4R
- up to 64 Kbyte of external data memory
- 256 bytes of internal data memory
- 1K bytes of internal XRAM data memory
- a 128 byte special function register area

Figure 3-1 illustrates the memory address spaces of the C515A.

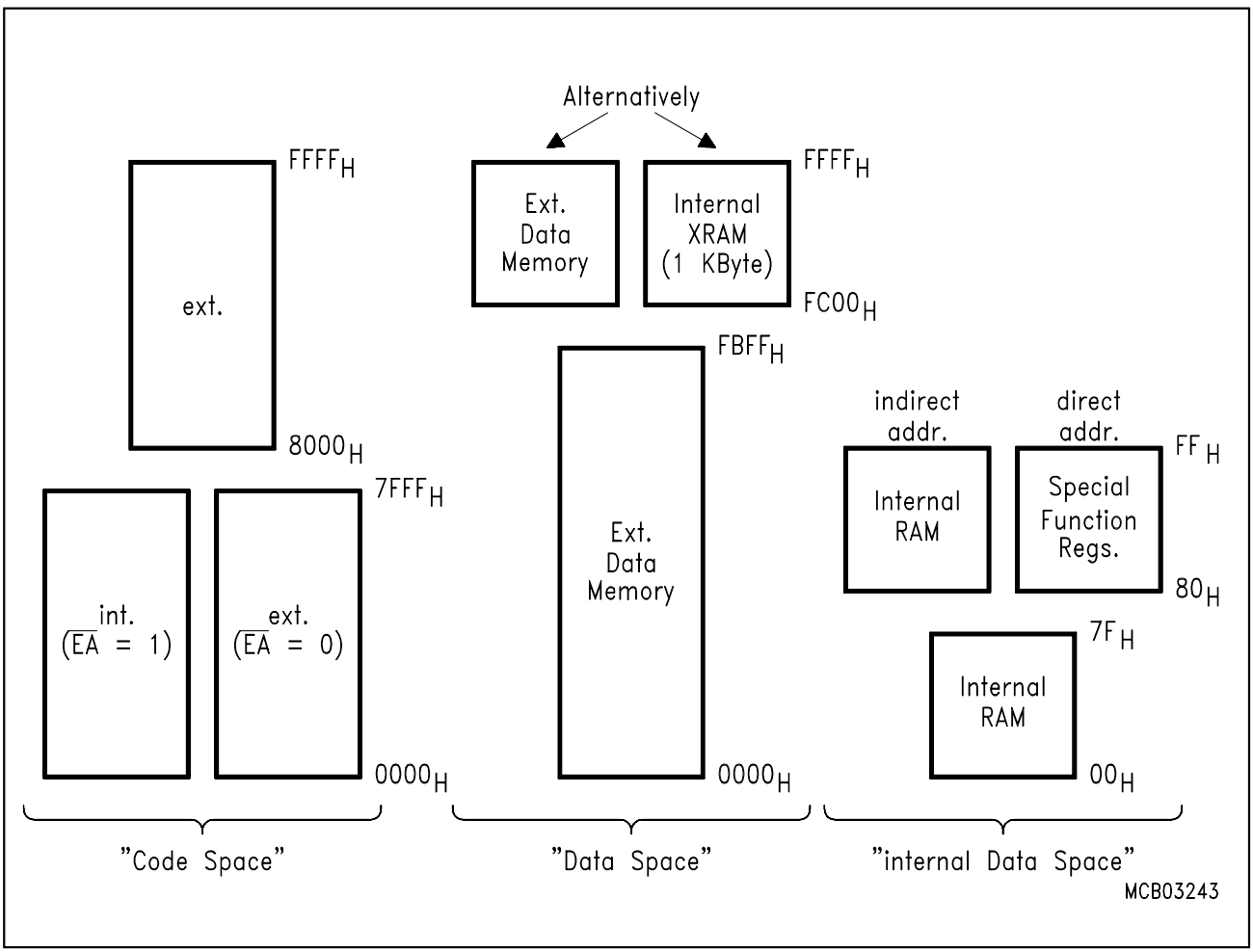


Figure 3-1
C515A Memory Map

3.1 Program Memory, "Code Space"

The C515A-4R has 32 Kbytes of read-only program memory which can be externally expanded up to 64 Kbytes. If the $\overline{EA}$ pin is held high, the C515A-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 $\overline{EA}$ pin is held low, the C515A fetches all instructions from the external 64K byte 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) area. While the upper 128 bytes of data memory and the SFR area share the same address locations, they are accessed through different addressing modes. The lower 128 bytes of data memory can be accessed through direct or register indirect addressing; the upper 128 bytes of RAM can be accessed through register indirect addressing; the special function registers are accessible through direct addressing. Four 8-register banks, each bank consisting of eight 8-bit multi-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 data memory address space, and the stack depth can be expanded up to 256 bytes.

The external data memory can be expanded up to 64 Kbyte 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, alternatively internal XRAM or external data RAM are accessed.

3.3 General Purpose Registers

The lower 32 locations of the internal RAM are assigned to four banks with eight general purpose registers (GPRs) each. Only one of these banks may be enabled at a time. Two bits in the program status word, 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 8 general purpose registers of the selected register bank may be accessed by register addressing. With register addressing the instruction op code indicates which register is to be used. For indirect addressing R0 and R1 are used as pointer or index register to address internal or external memory (e.g. MOV @R0).

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

3.4 XRAM Operation

The XRAM in the C515A 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 in this section.

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

After a reset operation, bit XMAP0 is set. This means that 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 (XRAM access enabled) it cannot be set by software. Only a reset operation will set the XMAP0 bit again. This hardware protection mechanism is done by an asymmetric latch at XMAP0 bit. An unintentional disabling of XRAM could be dangerous since indeterminate values could be read from the external bus. To avoid this the XMAP0 bit is forced to '1' only by a reset operation. Additionally, during reset an internal capacitor is charged. So the reset state is a disabled XRAM. Because of the charge time of the capacitor, XMAP0 bit once written to '0' (that is, discharging the capacitor) cannot be set to '1' again by software. On the other hand any distortion (software hang up, noise,...) is not able to charge this capacitor, too. That is, the stable status is XRAM enabled.

The clear instruction for the XMAP0 bit should be integrated in 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 provides also instructions for accesses to 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 C515A to provide the possibility of accessing the XRAM also with the MOVX @Ri instructions, i.e. XPAGE serves the same function for the XRAM as Port 2 for external data memory.

Special Function Register XPAGE (Address 91_H) **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 controller high address XPAGE.7-0 is the address part A15-A8 when 8-bit MOVX instructions are used to access internal XRAM.

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

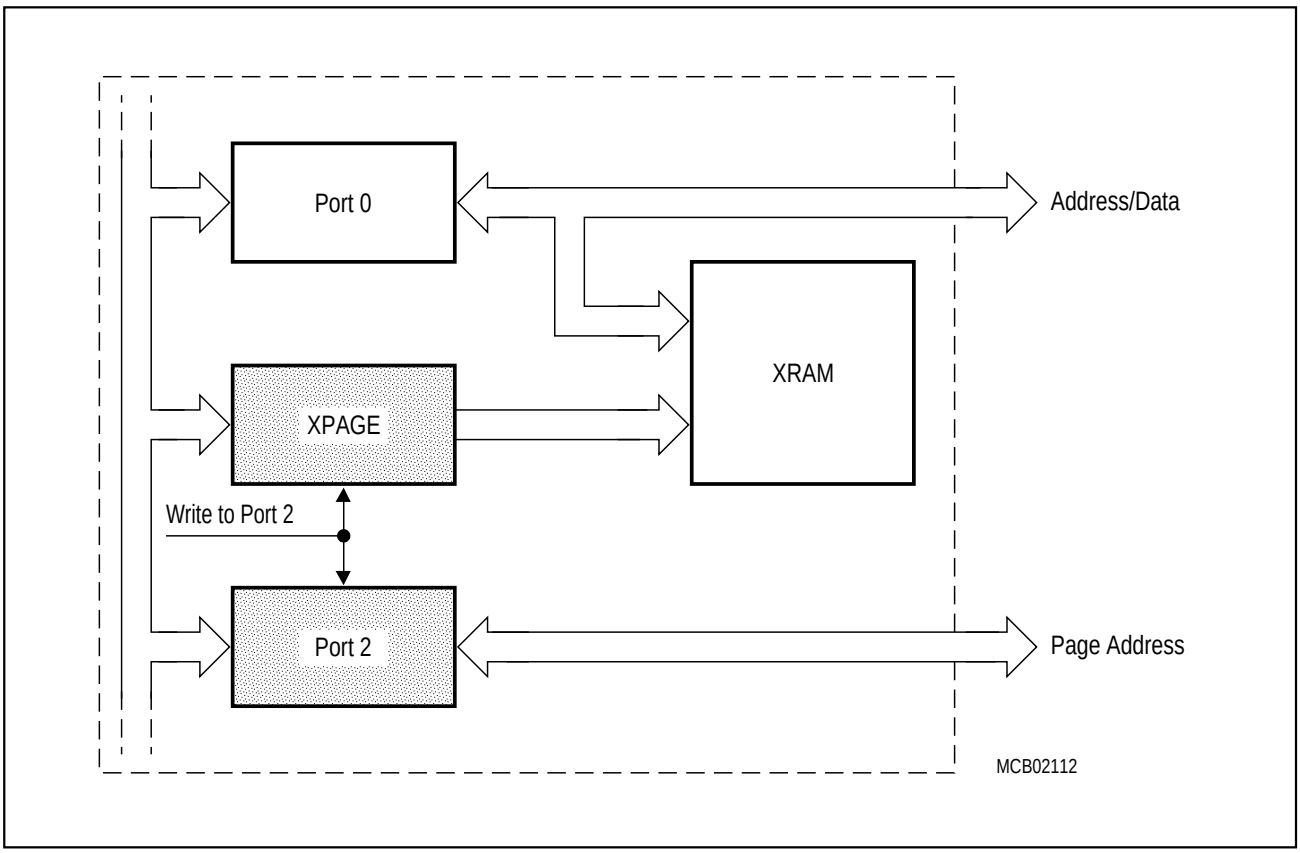


Figure 3-2
Write Page Address to Port 2

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

When external RAM is to be accessed in the XRAM address range, the XRAM has to be disabled. When additional external RAM is to be addressed in an address range < $FC00_H$, the XRAM may remain enabled and there is no need to overwrite XPAGE by a second move.

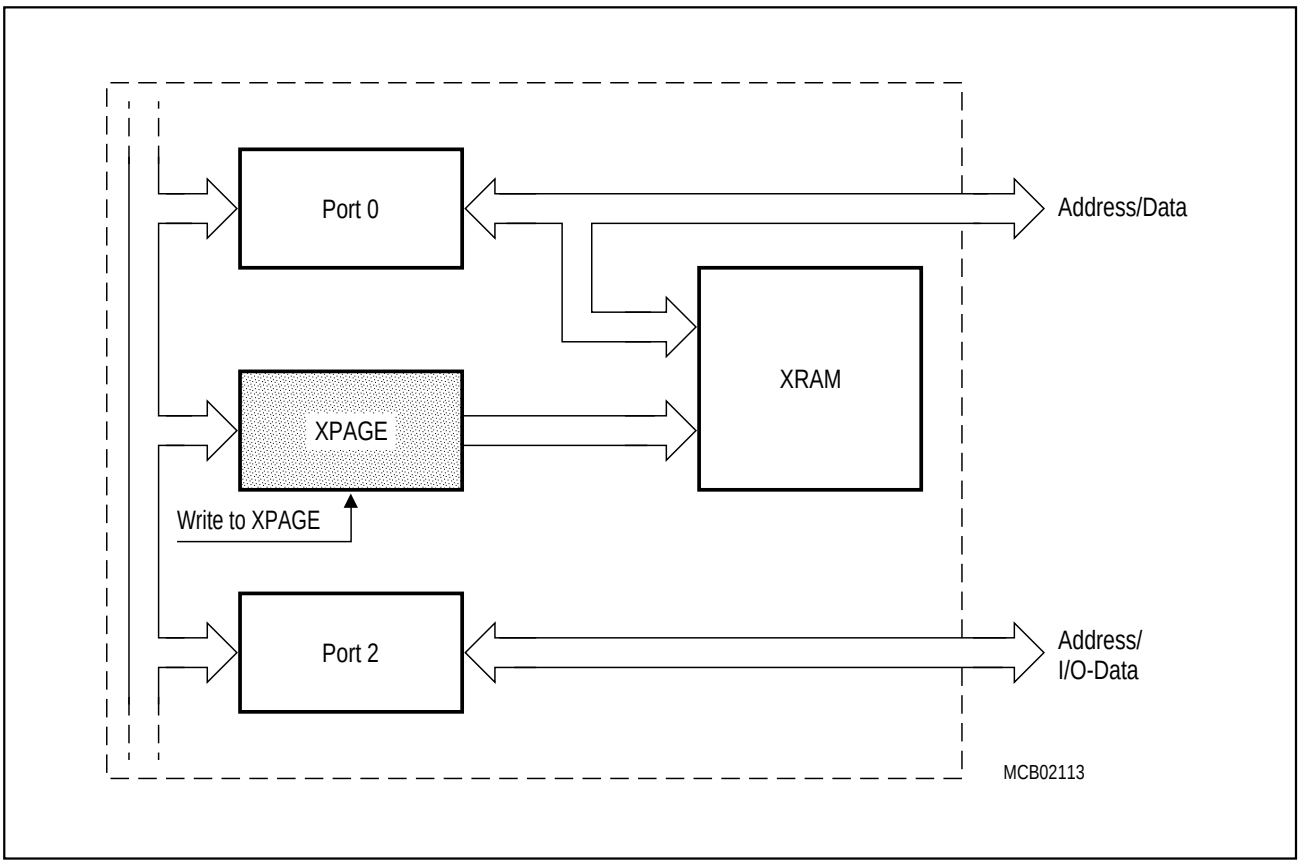


Figure 3-3
Write Page Address to XPAGE

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

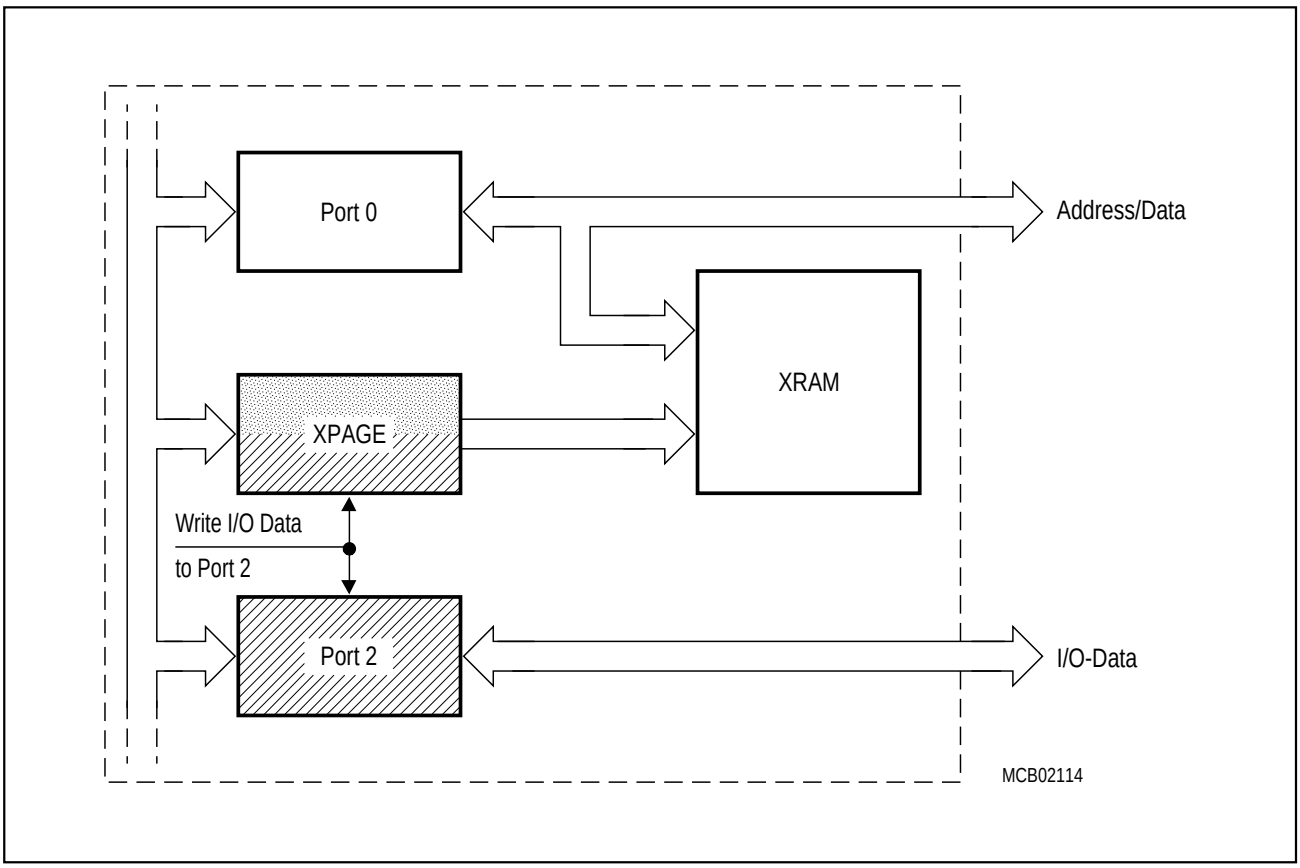


Figure 3-4
Use of Port 2 as I/O Port

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

Example :

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

```
MOV     R0, #30H          ;
MOV     P2, #0AAH         ; P2 shows AAH and XPAGE contains AAH
MOV     XPAGE, #0FFH      ; P2 still shows AAH but XRAM is addressed
MOVX    A, @R0            ; the contents of XRAM at FF30H is moved to accumulator
```

The register XPAGE provides the upper address byte for accesses to XRAM with MOVX @Ri instructions. If the address formed by XPAGE and Ri points outside the XRAM address range, an external access is performed. For the C515A the content of XPAGE must be FC_H - FF_H in order to use the XRAM.

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

a) Access to XRAM : The upper address byte must be written to XPAGE or P2; 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 content of a XRAM memory location depends on the cycle in which the active reset signal is detected (MOVX is a 2-cycle instruction):

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

3.4.5 Behaviour of Port 0 and Port 2

The behaviour of port 0 and port 2 during a MOVX access depends on the control bits in register SYSCON and on the state of pin $\overline{EA}$. The **table 3-1** lists the various operating conditions. It shows the following characteristics:

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

The shaded areas describe the standard operation as each C500 microcontroller device without on-chip XRAM behaves.

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

(shaded) modes compatible to 8051/C501 family

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

3.5 Special Function Registers

The registers, except the program counter and the four general purpose register banks, 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 C515A (PCON1) is located in the mapped special function register area. For accessing this mapped special function register, bit RMAP in 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 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 in this section.

Bit	Function
RMAP	Special function register map bit RMAP = 0 : The access to the non-mapped (standard) special function register area is enabled. RMAP = 1 : The access to the mapped special function register area (SFR PCON1) 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 (SFR PCON1) can be accessed. This bit is not cleared by hardware automatically. Thus, when non-mapped/mapped registers are to be accessed, the bit RMAP must be cleared/set respectively by software.

The 49 special function registers (SFRs) in the standard and mapped SFR area include pointers and registers that provide an interface between the CPU and the other on-chip peripherals. All SFRs with addresses where address bits 0-2 are 0 (e.g. 80_H, 88_H, 90_H, 98_H, ..., $F8_H$, FF_H) are bitaddressable. The SFRs of the C515A are listed in **table 3-1** and **table 3-2**. **Table 3-2** illustrates the contents of the SFRs in numeric order of their addresses.

Table 3-1
Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	**$E0_H$** [1]	00_H
	B	B-Register	**$F0_H$** [1]	00_H
	DPH	Data Pointer, High Byte	83_H	00_H
	DPL	Data Pointer, Low Byte	82_H	00_H
	PSW	Program Status Word Register	**$D0_H$** [1]	00_H
	SP	Stack Pointer	81_H	07_H
	SYSCON [2]	System/XRAM Control Register	$B1_H$	XX10 $XX01_B$ [3]
A/D-Converter	ADCON0 [2]	A/D Converter Control Register 0	**$D8_H$** [1]	00_H
	ADCON1	A/D Converter Control Register 1	DC_H	0XXX $X000_B$ [3]
	ADDATH	A/D Converter Data Register, High Byte	$D9_H$	00_H
	ADDATL	A/D Converter Data Register, low Byte	DA_H [4]	00XX $XXXX_B$ [3]
Interrupt System	IEN0 [2]	Interrupt Enable Register 0	**$A8_H$** [1]	00_H
	IEN1 [2]	Interrupt Enable Register 1	**$B8_H$** [1]	00_H
	IP0 [2]	Interrupt Priority Register 0	$A9_H$	00_H
	IP1 [2]	Interrupt Priority Register 1	$B9_H$	XX00 0000_B [3]
	IRCON	Interrupt Request Control Register	**$C0_H$** [1]	00_H
	TCON [2]	Timer Control Register	**88_H** [1]	00_H
	T2CON [2]	Timer 2 Control Register	**$C8_H$** [1]	00_H
	SCON [2]	Serial Channel Control Register	**98_H** [1]	00_H
Timer 0/ Timer 1	TCON [2]	Timer 0/1 Control Register	**88_H** [1]	00_H
	TH0	Timer 0, High Byte	$8C_H$	00_H
	TH1	Timer 1, High Byte	$8D_H$	00_H
	TL0	Timer 0, Low Byte	$8A_H$	00_H
	TL1	Timer 1, Low Byte	$8B_H$	00_H
	TMOD	Timer Mode Register	89_H	00_H
Compare/ Capture Unit / Timer 2	CCEN	Comp./Capture Enable Reg.	$C1_H$	00_H
	CCH1	Comp./Capture Reg. 1, High Byte	$C3_H$	00_H
	CCH2	Comp./Capture Reg. 2, High Byte	$C5_H$	00_H
	CCH3	Comp./Capture Reg. 3, High Byte	$C7_H$	00_H
	CCL1	Comp./Capture Reg. 1, Low Byte	$C2_H$	00_H
	CCL2	Comp./Capture Reg. 2, Low Byte	$C4_H$	00_H
	CCL3	Comp./Capture Reg. 3, Low Byte	$C6_H$	00_H
	CRCH	Com./Rel./Capt. Reg. High Byte	CB_H	00_H
	CRCL	Com./Rel./Capt. Reg. 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 [2]	Timer 2 Control Register	**$C8_H$** [1]	00_H

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

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

Block	Symbol	Name	Address	Contents after Reset
Ports	P0	Port 0	**80_H** [1)]	FF_H
	P1	Port 1	**90_H** [1)]	FF_H
	P2	Port 2	**$A0_H$** [1)]	FF_H
	P3	Port 3	**$B0_H$** [1]	FF_H
	P4	Port 4	**$E8_H$** [1)]	FF_H
	P5	Port 5	**$F8_H$** [1)]	FF_H
	P6	Port 6, Analog/Digital Input	DB_H	–
XRAM	XPAGE	Page Address Register for Extended On-Chip RAM	91_H	00_H
	SYSCON [2)]	System/XRAM Control Register	$B1_H$	$XX10\ XX01_B$ [3)]
Serial Channel	ADCON0 [2)]	A/D Converter Control Register	**$D8_H$** [1]	00_H
	PCON [2)]	Power Control Register	87_H	00_H
	SBUF	Serial Channel Buffer Register	99_H	XX_H [3)]
	SCON [2)]	Serial Channel Control Register	**98_H** [1)]	00_H
	SRELL	Serial Channel Reload Register, Low Byte	AA_H	$D9_H$
	SRELH	Serial Channel Reload Register, High Byte	BA_H	$XXXX\ XX11_B$ [3)]
Watchdog	IEN0 [2)]	Interrupt Enable Register 0	**$A8_H$** [1)]	00_H
	IEN1 [2)]	Interrupt Enable Register 1	**$B8_H$** [1)]	00_H
	IP0 [2))]	Interrupt Priority Register 0	$A9_H$	00_H
	IP1 [2)]	Interrupt Priority Register 1	$B9_H$	$XX00\ 0000_B$ [3)]
	WDTREL	Watchdog Timer Reload Register	86_H	00_H
Power Saving Modes	PCON [2)]	Power Control Register	87_H	00_H
	PCON1 [4)]	Power Control Register 1	88_H	$0XXX\ XXXX_B$ [3)]

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

Table 3-2
Contents of the SFRs, SFRs in numeric order of their addresses

Addr	Register	Content after Reset[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80_H [2]	P0	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
81_H	SP	07_H	.7	.6	.5	.4	.3	.2	.1	.0
82_H	DPL	00_H	.7	.6	.5	.4	.3	.2	.1	.0
83_H	DPH	00_H	.7	.6	.5	.4	.3	.2	.1	.0
86_H	WDTREL	00_H	WDT PSEL	.6	.5	.4	.3	.2	.1	.0
87_H	PCON	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	$0XXX\text{-}XXXX_B$	EWPD	–	–	–	–	–	–	–
89_H	TMOD	00_H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
$8A_H$	TL0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8B_H$	TL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8C_H$	TH0	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$8D_H$	TH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
90_H [2]	P1	FF_H	T2	CLK-OUT	T2EX	$\overline{INT2}$	INT6	INT5	INT4	$\overline{INT3}$
91_H	XPAGE	00_H	.7	.6	.5	.4	.3	.2	.1	.0
98_H [2]	SCON	00_H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99_H	SBUF	XX_H	T2	.6	.5	.4	.3	.2	.1	.0
$A0_H$ [2]	P2	FF_H	.7	.6	.5	.4	.3	.2	.1	.0
$A8_H$ [2]	IEN0	00_H	EAL	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
$B0_H$ [2]	P3	FF_H	RD	WR	T1	T0	INT1	INT0	TxD	RxD
$B1_H$	SYSCON	$XX10\text{-}XX01_B$	–	–	EALE	RMAP	–	–	XMAP1	XMAP0
$B8_H$ [2]	IEN1	00_H	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC
$B9_H$	IP1	$XX00\text{-}0000_B$	–	–	.5	.4	.3	.2	.1	.0

[1] X means that the value is undefined and the location is reserved
[2] Bit-addressable special function registers
[3] SFR is located in the mapped SFR area. For accessing this SFR, bit RMAP in SFR SYSCON must be set.

Table 3-2
Contents of the SFRs, SFRs in numeric order of their addresses (cont'd)

Addr	Register	Content after Reset[1)]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BA_H	SRELH	XXXX-XX11$_B$	–	–	–	–	–	–	.1	.0
$C0_H$[2)]	IRCON	00_H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC
$C1_H$	CCEN	00_H	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0
$C2_H$	CCL1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C3_H$	CCH1	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C4_H$	CCL2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C5_H$	CCH2	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C6_H$	CCL3	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C7_H$	CCH3	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$C8_H$[2)]	T2CON	00_H	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
$D8_H$[2)]	ADCON0	00_H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0
$D9_H$	ADDATH	00_H	.9	.8	.7	.6	.5	.4	.3	.2
DA_H	ADDATL	00XX-XXXX$_B$	.1	.0	–	–	–	–	–	–
DB_H	P6	–	.7	.6	.5	.4	.3	.2	.1	.0
DC_H	ADCON1	0XXX-X000$_B$	ADCL	–	–	–	–	MX2	MX1	MX0
$E0_H$[2)]	ACC	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$E8_H$[2)]	P4	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F0_H$[2)]	B	00_H	.7	.6	.5	.4	.3	.2	.1	.0
$F8_H$[2)]	P5	FF_H	.7	.6	.5	.4	.3	.2	.1	.0

[1)] X means that the value is undefined and the location is reserved
[2)] Bit-addressable special function registers

4 External Bus Interface

The C515A allows for external memory expansion. The functionality and implementation of the external bus interface is identical to the common interface for the 8051 architecture with one exception : if the C515A is used in systems with no external memory the generation of the ALE signal can be suppressed. Resetting bit EALE in SFR SYSCON register, the ALE signal will be gated off. This feature reduces RFI emissions of the system.

4.1 Accessing External Memory

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

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

4.1.1 Role of P0 and P2 as Data/Address Bus

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

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

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

Thus the port 2 latch does not have to contain 1s, and the contents of the port 2 SFR are not modified.

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

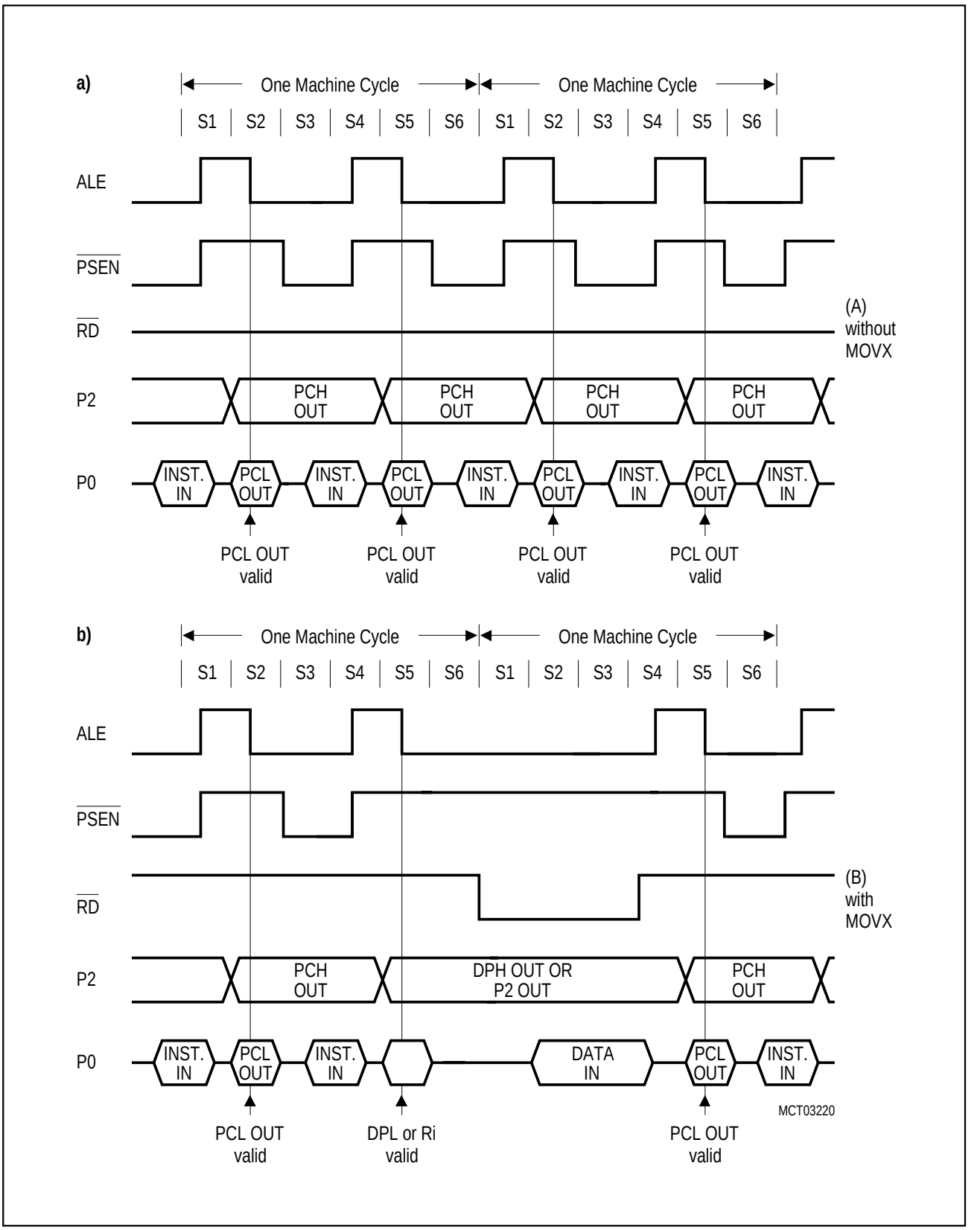


Figure 4-1
External Program Memory Execution

4.1.2 Timing

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

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

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

4.1.3 External Program Memory Access

The external program memory is accessed under two conditions:

- whenever signal $\overline{EA}$ is active (low); or
- whenever the program counter (PC) content is greater than $7FFF_H$

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

Since the C515A-L has no internal program memory, accesses to program memory are always external, and port 2 is at all times dedicated to output the high-order address byte. This means that port 0 and port 2 of the C515A-L can never be used as general-purpose I/O. This also applies to the C515A-1R when it operates with only an external program memory.

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

The read strobe for external program memory fetches is $\overline{PSEN}$. It is not activated for internal program memory fetches. When the CPU is accessing external program memory, $\overline{PSEN}$ is activated twice every instruction cycle (except during a MOVX instruction) no matter whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as for $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 6 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$, takes 3 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 C515A the external program and data memory spaces can be combined by the logical-AND of $\overline{PSEN}$ and $\overline{RD}$. A positive result from this AND operation produces a low active read strobe that can be used for the combined physical memory. Since the $\overline{PSEN}$ cycle is faster than the $\overline{RD}$ cycle, the external memory needs to be fast enough to adapt to the $\overline{PSEN}$ cycle.

4.4 ALE, Address Latch Enable

The C515A allows to switch off the ALE output signal. If the internal ROM is used ($\overline{EA}$=1 and PC $\leq$ $7FFF_H$) and ALE is switched off by EALE=0, then, ALE will only go active during external data memory accesses (MOVX instructions). If $\overline{EA}$=0, the ALE generation is always enabled and the bit EALE has no effect.

After a hardware reset the ALE generation is enabled.

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

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

The function of 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 ($\overline{EA}$=1). With $\overline{EA}$=1, ALE is automatically generated at MOVX instructions. EALE = 1 : ALE generation is enabled If $\overline{EA}$=0, the ALE generation is always enabled and the bit EALE has no effect on 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 on the internal operation of the controllers. Emulation of on-chip ROM based programs is possible, too.
Each C500 production chip has built-in logic for the support of the Enhanced Hooks Emulation Concept. Therefore, no costly bond-out chips are necessary for emulation. This also ensure that emulation and production chips are identical.

The Enhanced Hooks Technology™ [1], which requires embedded logic in the C500 allows the C500 together with an EH-IC to function 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 different versions of the C500 microcontrollers. This includes emulation of ROM, ROM with code rollover and ROMless modes of operation. It is also able to operate in single step mode and to read the SFRs after a break.

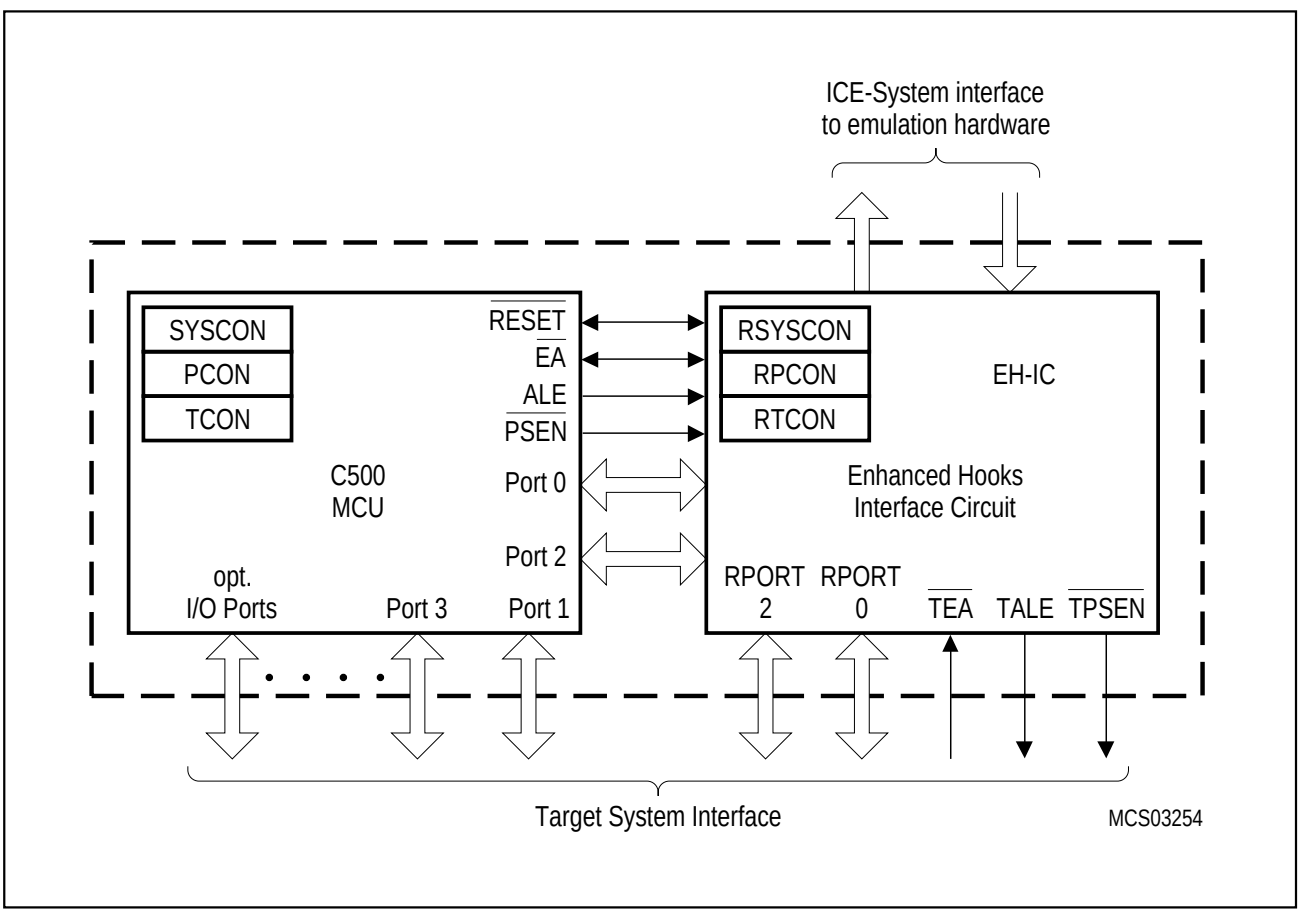


Figure 4-2
Basic C500 MCU Enhanced Hooks Concept Configuration

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

1) "Enhanced Hooks Technology" is a trademark and patent of Metalink Corporation licensed to Siemens.

4.6 ROM Protection for the C515A

The C515A-4R allows to protect the contents of the internal ROM against unauthorized read out. The type of ROM protection (protected or unprotected) is fixed with the ROM mask. Therefore, the customer of a C515A-4R version has to define whether ROM protection has to be selected or not.

The C515A-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 C515A-4R and the ROM data byte is output at port 0. ROM verification mode 2, which is used to verify ROM protected devices, operates different : ROM addresses are generated internally and the expected data bytes must be applied externally to the device (by the manufacturer or by the customer) and are compared internally with the data bytes from the ROM. After 16 byte verify operations the state of the P3.5 pin shows whether the last 16 bytes have been verified correctly.

This mechanism provides a very high security of ROM protection. Only the owner of the ROM code and the manufacturer who know the contents of the ROM can read out and verify it with less effort.

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

4.6.1 Unprotected ROM Mode

If the ROM is unprotected, the ROM verification mode 1 as shown in **figure 4-3** is used to read out the contents of the ROM. The AC timing characteristics of the ROM verification mode is shown in the AC specifications (chapter 10).

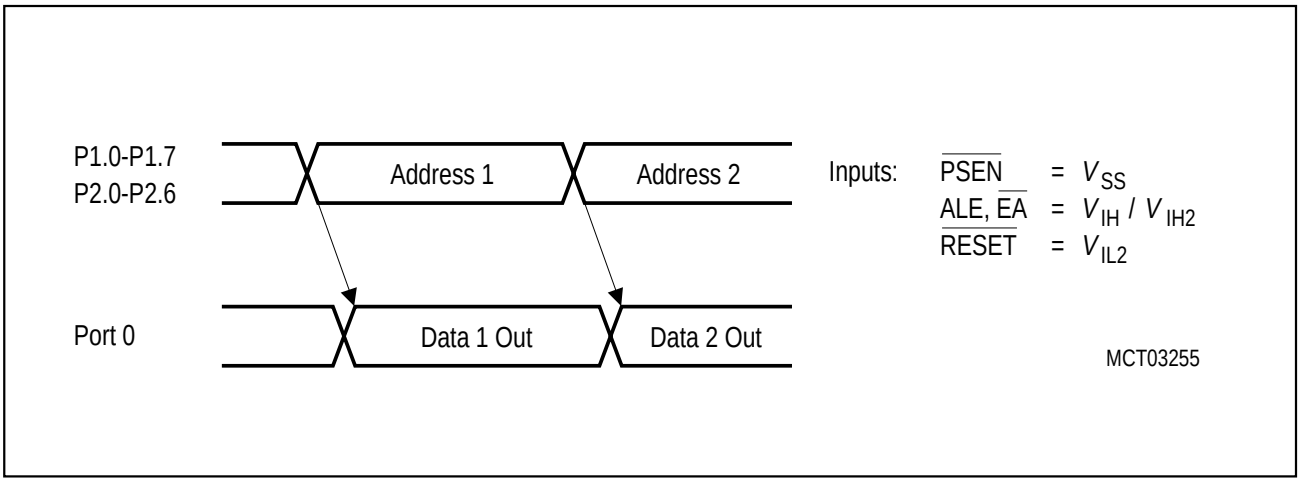


Figure 4-3
ROM Verification Mode 1

ROM verification mode 1 is selected if the inputs $\overline{\text{PSEN}}$, ALE, $\overline{\text{EA}}$, and $\overline{\text{RESET}}$ are put to the specified logic level. Then the 14-bit address of the internal ROM byte to be read is applied to the port 1 and port 2 lines. After a delay time, port 0 outputs the content of the addressed ROM cell. In ROM verification mode 1, the C515A must be provided with a system clock at the XTAL pins and pullup resistors on the port 0 lines.

4.6.2 Protected ROM Mode

If the ROM is protected, the ROM verification mode 2 as shown in **figure 4-4** is used to verify the contents of the ROM. The detailed timing characteristics of the ROM verification mode is shown in the AC specifications (chapter 10).

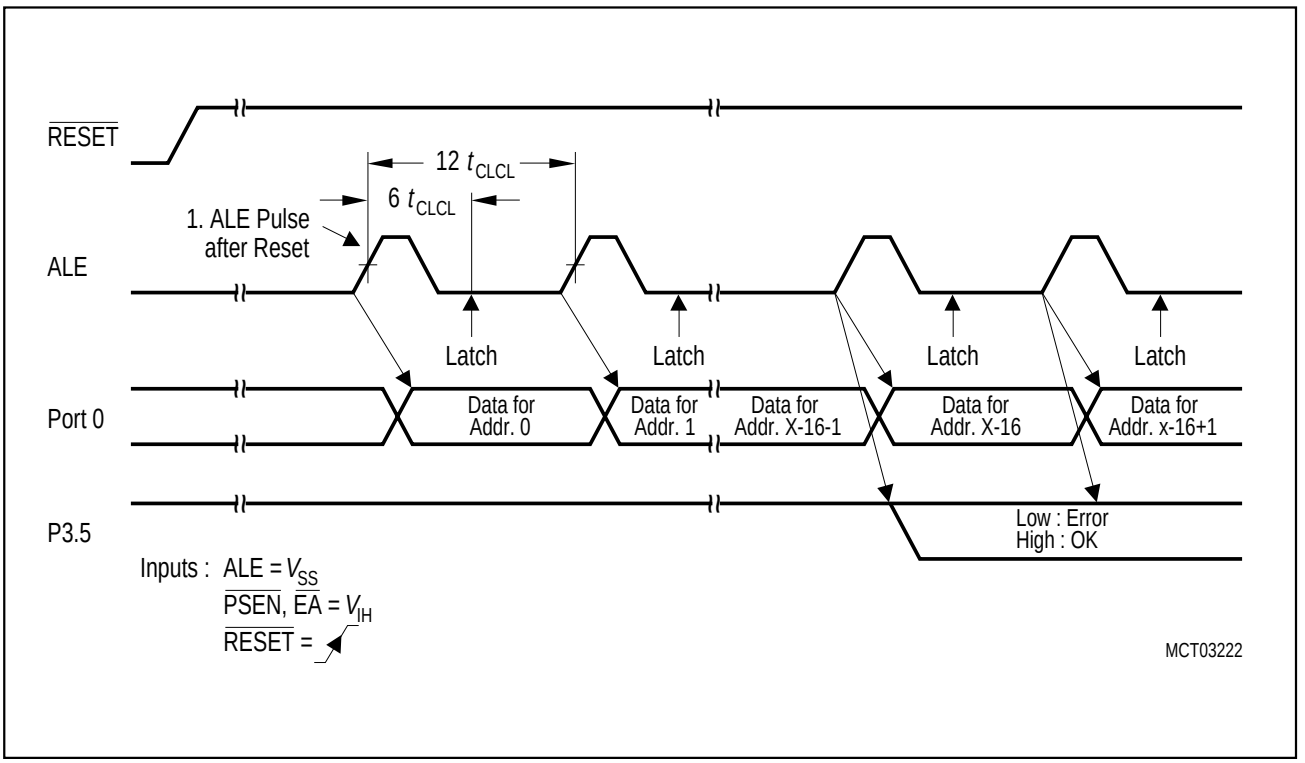


Figure 4-4
ROM Verification Mode 2

ROM verification mode 2 is selected if the inputs $\overline{PSEN}$, $\overline{EA}$, and ALE are put to the specified logic levels. With $\overline{RESET}$ going inactive, the ROM verification mode 2 sequence is started. The C515A outputs an ALE signal with a period of 12 t_{CLCL} and expects data bytes at port 0. The data bytes at port 0 are assigned to the ROM addresses in the following way :

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

The C515A-1R does not output any address information during the ROM 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 is put onto the data bus with the first rising edge of ALE. With each following ALE pulse the ROM 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 3 CLP after ALE rising edge) and compared internally with the ROM content of the actual address. If an verify error is detected, the error

condition is stored internally. After each 16th data byte the cumulated verify result (pass or fail) of the last 16 verify operations is output at P3.5. This means that P3.5 stays at static level (low for fail and high for pass) during the 16 bytes are checked. In ROM verification mode 2, the C515A-4R must be provided with a system clock at the XTAL pins

Figure 4-5 shows an application example of an external circuitry which allows to verify a protected ROM inside the C515A-4R in ROM verification mode 2. With $\overline{\text{RESET}}$ going inactive, the C515A-4R starts the ROM 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. The verify detect logic typically displays the pass/fail information of the verify operation. P3.5 can be latched with the falling edge of ALE.

When the last byte of the internal ROM has been handled, the C515A-4R starts generating a $\overline{\text{PSEN}}$ signal. This signal or the CY signal of the address counter indicate to the verify detect logic the end of the internal ROM verification.

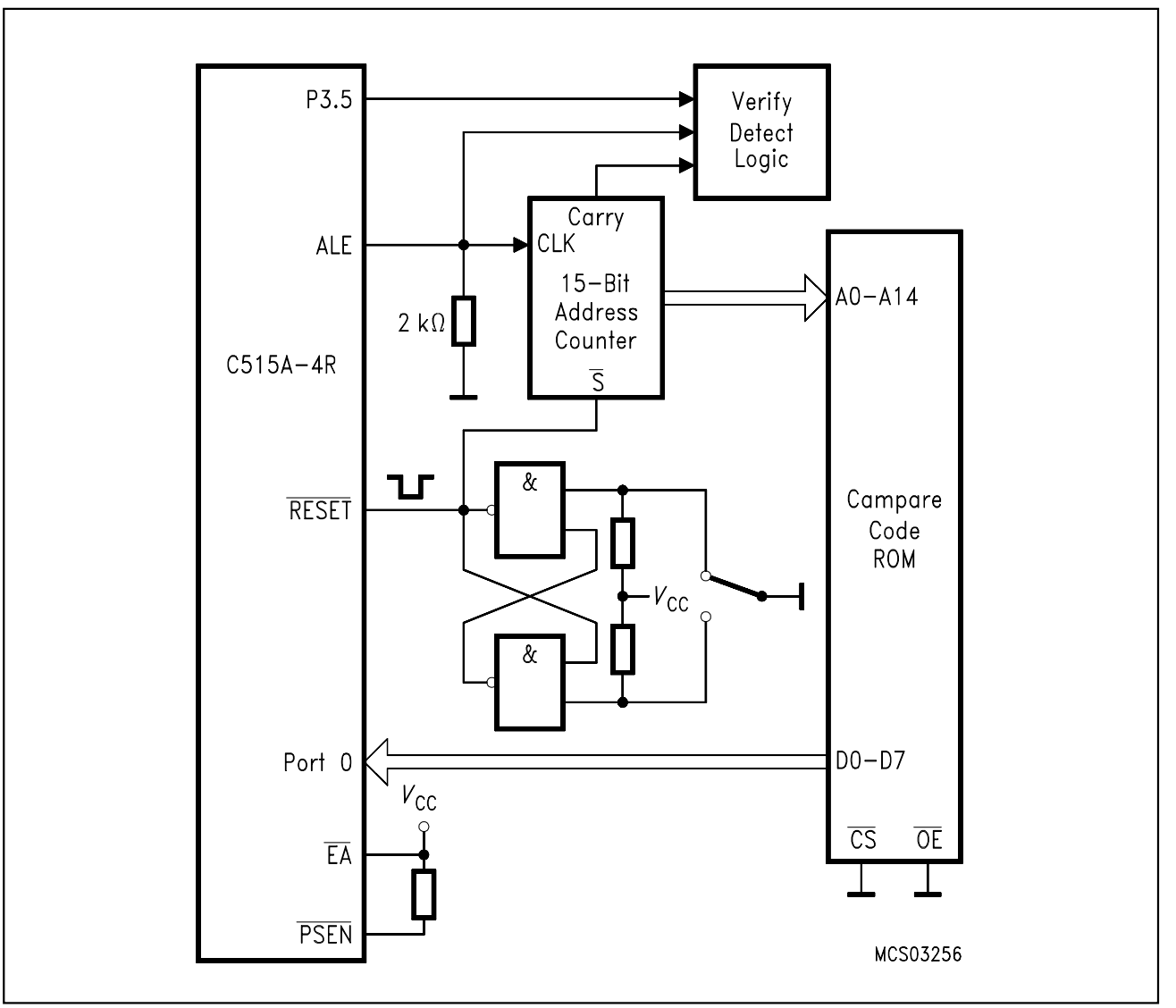


Figure 4-5
ROM Verification Mode 2 - External Circuitry Example

5 Reset and System Clock Operation

5.1 Hardware Reset Operation

The hardware reset function incorporated in the C515A allows for an easy automatic start-up at a minimum of additional hardware and forces the controller to a predefined default state. The hardware reset function can also be used during normal operation in order to restart the device. This is particularly done when the power-down mode is to be terminated.

Additional to the hardware reset, which is applied externally to the C515A, there are two internal reset sources, the watchdog timer and the oscillator watchdog. This chapter deals only with the external hardware reset.

The reset input is an active low input. An internal Schmitt trigger is used at the input for noise rejection. Since the reset is synchronized internally, the $\overline{\text{RESET}}$ pin must be held low for at least two machine cycles (24 oscillator periods) while the oscillator is running. With the oscillator running the internal reset is executed during the second machine cycle and is repeated every cycle until $\overline{\text{RESET}}$ goes high again.

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

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

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

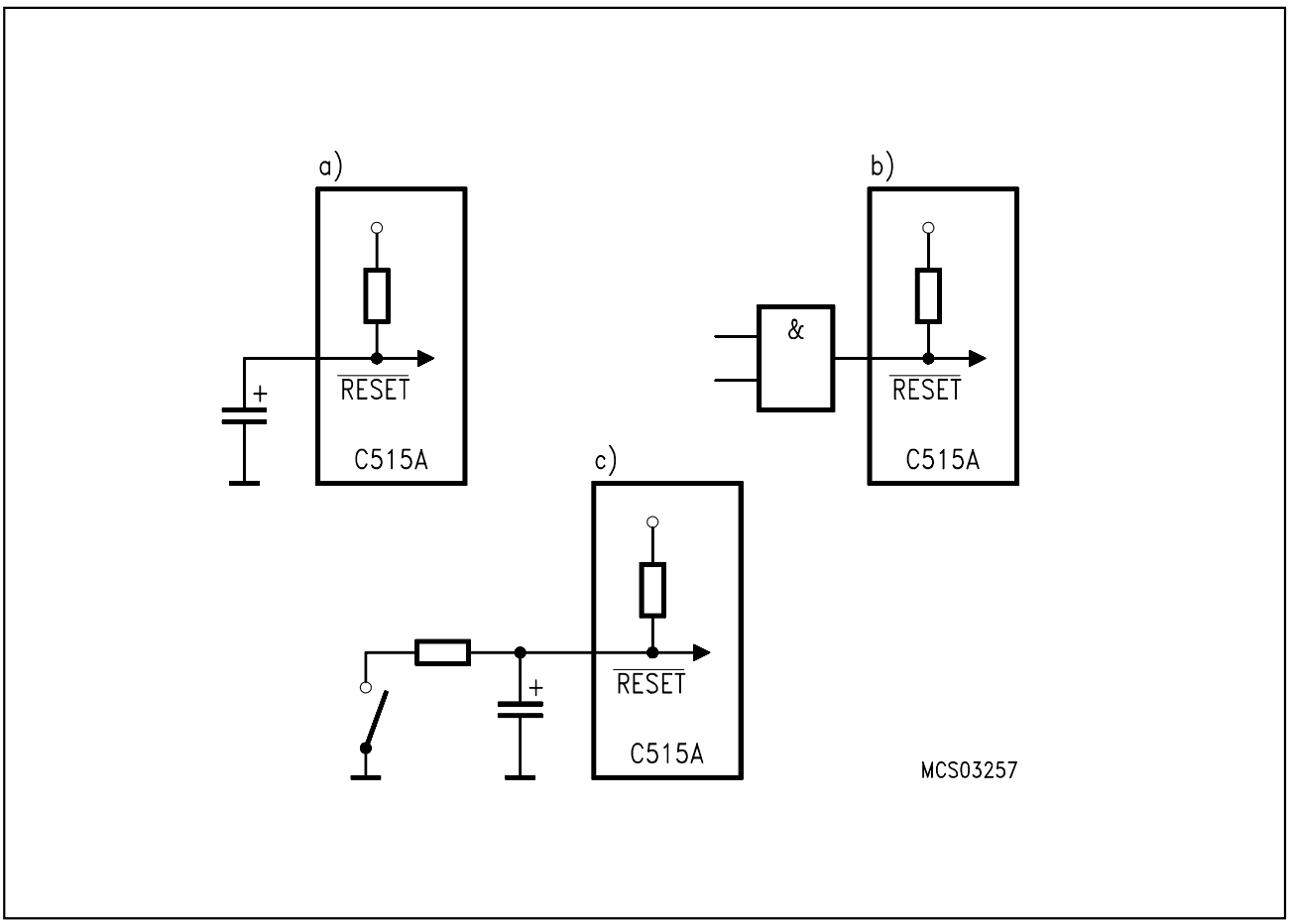


Figure 5-6
Reset Circuitries

A correct reset leaves the processor in a defined state. The program execution starts at location 0000_H. After reset is internally accomplished the port latches of ports 0 to 5 are set 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, 4 and 5) output a one (1). Port 2 lines output a zero (or one) after reset, if $\overline{EA}$ is held low (or high). Port 6 is an input-only port. It has no internal latch and therefore the contents of the special function registers P6 depend on the levels applied to port 6.

The content of the internal RAM of the C515A is not affected by a reset. After power-up the content is undefined, while it remains unchanged during a reset if the power supply is not turned off.

5.2 Fast Internal Reset after Power-On

The C515A uses the oscillator watchdog unit for a fast internal reset procedure after power-on. **Figure 5-1** shows the power-on sequence under control of the oscillator watchdog.

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

In the C515A the oscillator watchdog unit avoids this situation. In this case, after power-on the oscillator watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 microseconds). In the following the watchdog circuitry detects a failure condition for the on-chip oscillator because this has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is detected the watchdog uses the RC oscillator output as clock source for the chip rather than the on-chip oscillator's output. This allows correct resetting of the part and brings also all ports to the defined state (see **figure 5-7**).

Under worst case conditions (fast V_{CC} rise time - e.g. 1µs, measured from V_{CC} = 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_{CC} below 4.25V (lower specification limit). Therefore, at slower V_{CC} rise times the delay time will be less than the two values given above.

After the on-chip oscillator has finally started, the oscillator watchdog detects the correct function; then the watchdog still holds the reset active for a time period of max. 768 cycles of the RC oscillator clock in order to allow the oscillation of the on-chip oscillator to stabilize (**figure 5-7, II**). Subsequently the clock is supplied by the on-chip oscillator and the oscillator watchdog's reset request is released (**figure 5-7, III**). However, an externally applied reset still remains active (**figure 5-7, IV**) and the device does not start program execution (**figure 5-7, V**) before the external reset is also released.

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

- Termination of software power down mode
- Reset of the status flag OWDS that is set by the oscillator watchdog during the power up sequence.

Using a crystal or ceramic resonator for clock generation, the external reset signal must be held active at least until the on-chip oscillator has started and the internal watchdog reset phase is completed (after phase III in **figure 5-7**). When an external clock generator is used, phase II is very short. Therefore, an external reset time of typically 1 ms is sufficent in most applications.

Generally, for reset time generation at power-on an external capacitor can be applied to the $\overline{\text{RESET}}$ pin.

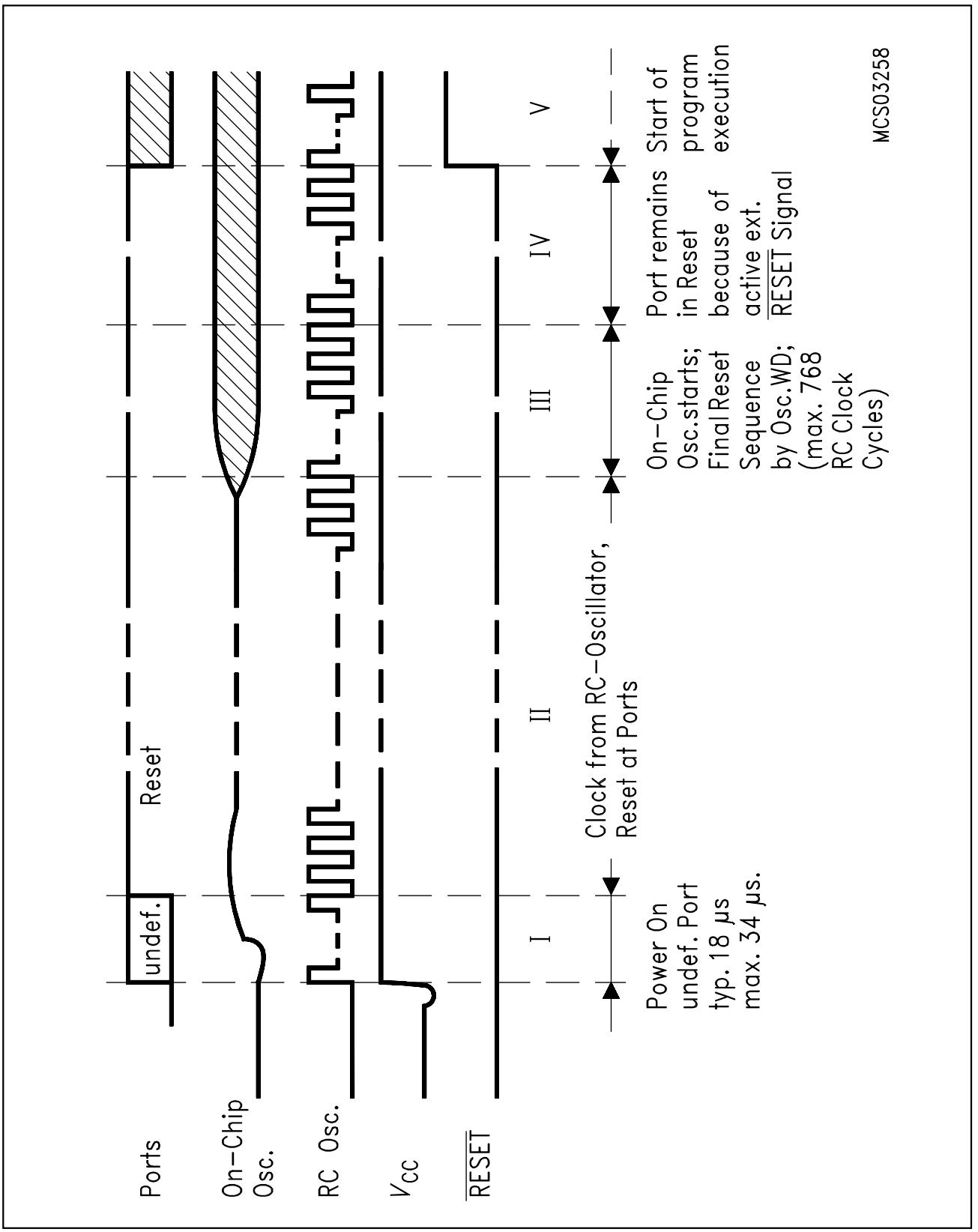


Figure 5-7
Power-On Reset of the C515A

5.3 Hardware Reset Timing

This section describes the timing of the hardware reset signal.

The input pin $\overline{\text{RESET}}$ is sampled once during each machine cycle. This happens in state 5 phase 2. Thus, the external reset signal is synchronized to the internal CPU timing. When the reset is found active (low level) the internal reset procedure is started. It needs two complete machine cycles to put the complete device to its correct reset state, i.e. all special function registers contain their default values, the port latches contain 1's etc. Note that this reset procedure is also performed if there is no clock available at the device. (This is done by the oscillator watchdog, which provides an auxiliary clock for performing a perfect reset without clock at the XTAL1 and XTAL2 pins). The $\overline{\text{RESET}}$ signal must be active for at least two machine cycles; after this time the C515A 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 5-8 shows this timing for a configuration with $\overline{\text{EA}}$ = 0 (external program memory). Thus, between the release of the $\overline{\text{RESET}}$ 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.

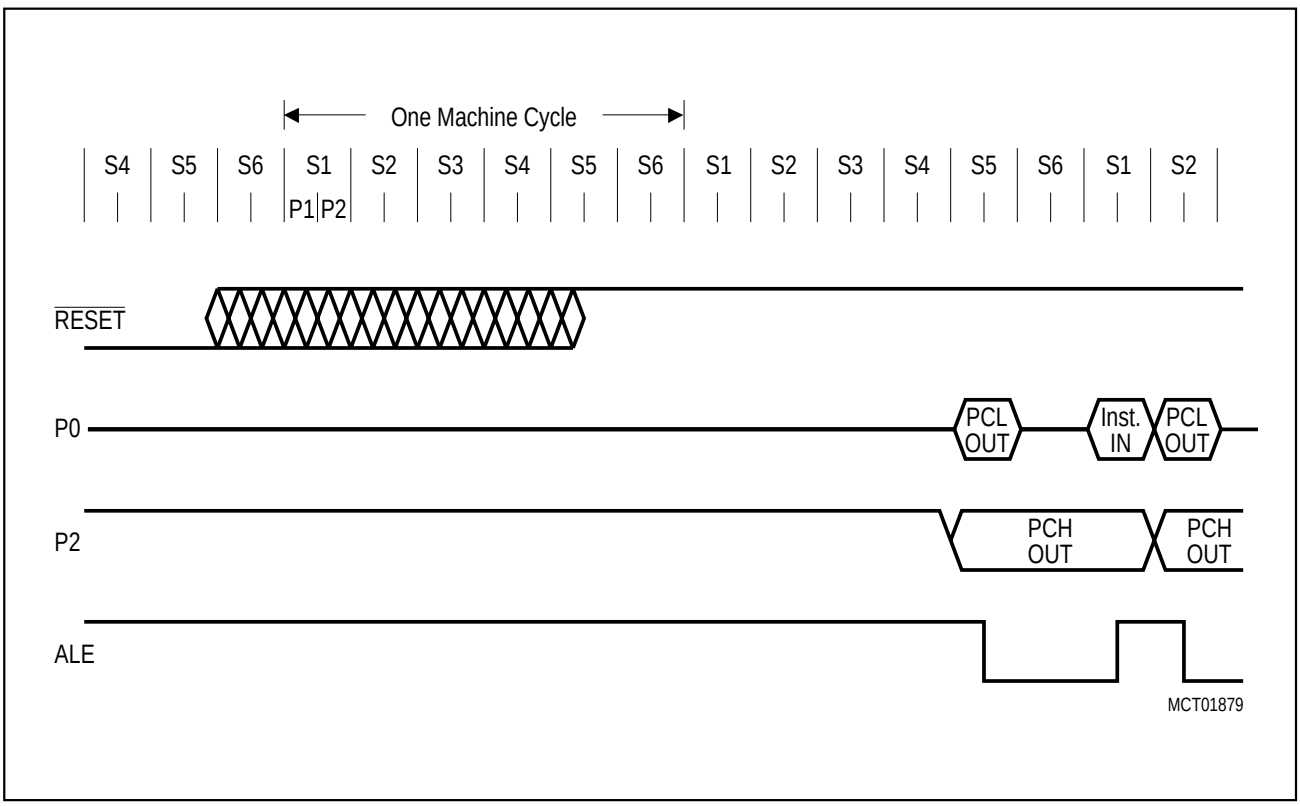


Figure 5-8
CPU Timing after Reset

5.4 Oscillator and Clock Circuit

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

Figure 5-9 shows the recommended oscillator circuit.

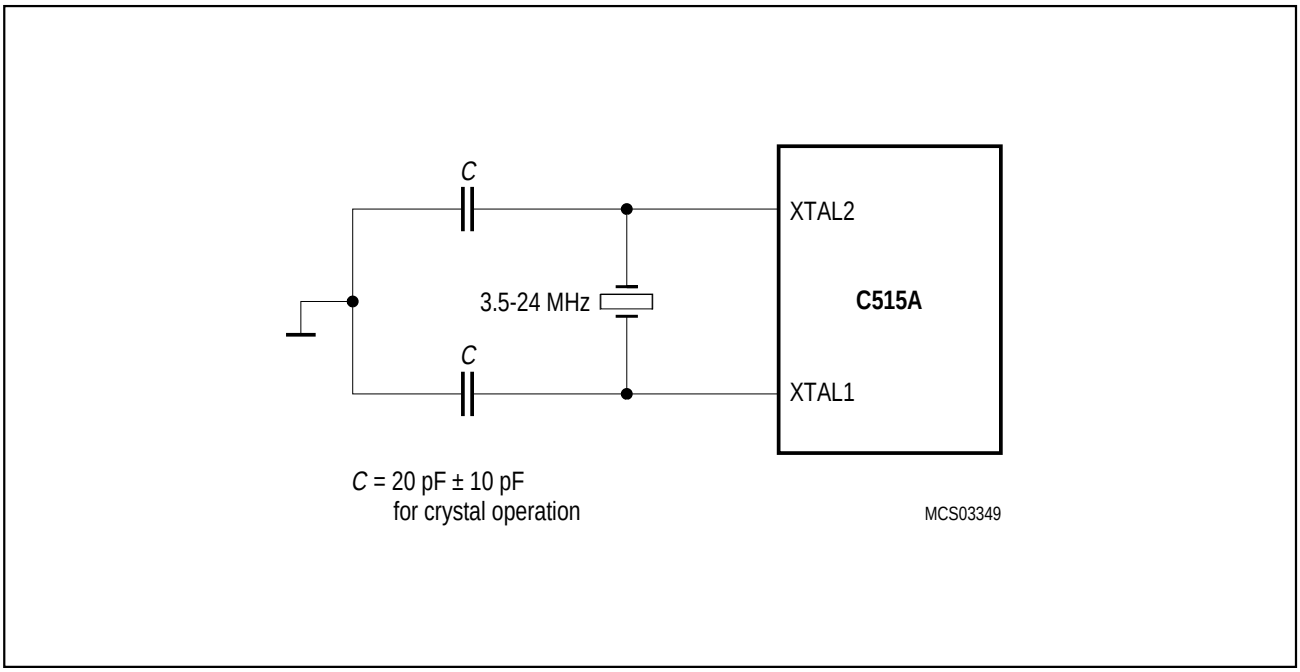


Figure 5-9
Recommended Oscillator Circuit

In this application the on-chip oscillator is used as a crystal-controlled, positive-reactance oscillator (a more detailed schematic is given in **figure 5-10**). It is operated in its fundamental response mode as an inductive reactor in parallel resonance with a capacitor external to the chip. The crystal specifications and capacitances are non-critical. In this circuit 20 pF can be used as single capacitance at any frequency together with a good quality crystal. A ceramic resonator can be used in place of the crystal in cost-critical applications. If a ceramic resonator is used, the two capacitors normally have different values depending on the oscillator frequency. We recommend consulting the manufacturer of the ceramic resonator for value specifications of these capacitors.

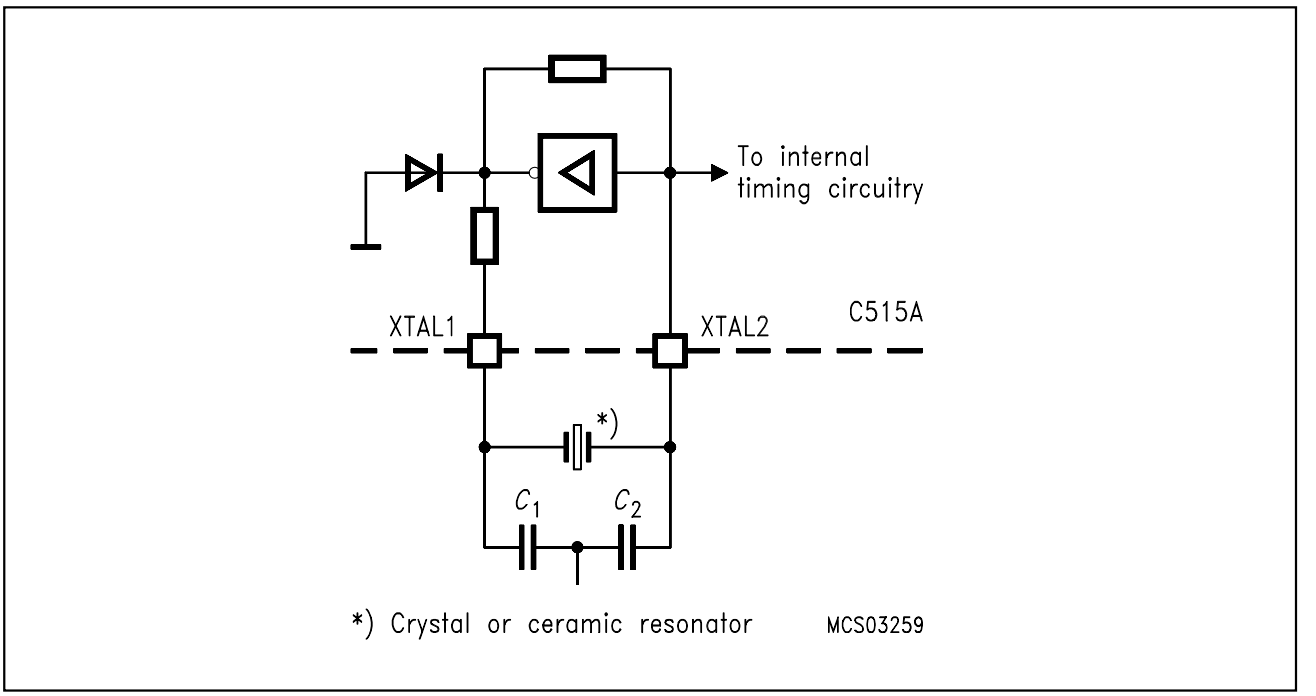


Figure 5-10
On-Chip Oscillator Circuiry

To drive the C515A with an external clock source, the external clock signal has to be applied to XTAL2, as shown in **figure 5-11**. XTAL1 has to be left unconnected. A pullup resistor is suggested (to increase the noise margin), but is optional if V_{OH} of the driving gate corresponds to the V_{IH2} specification of XTAL2.

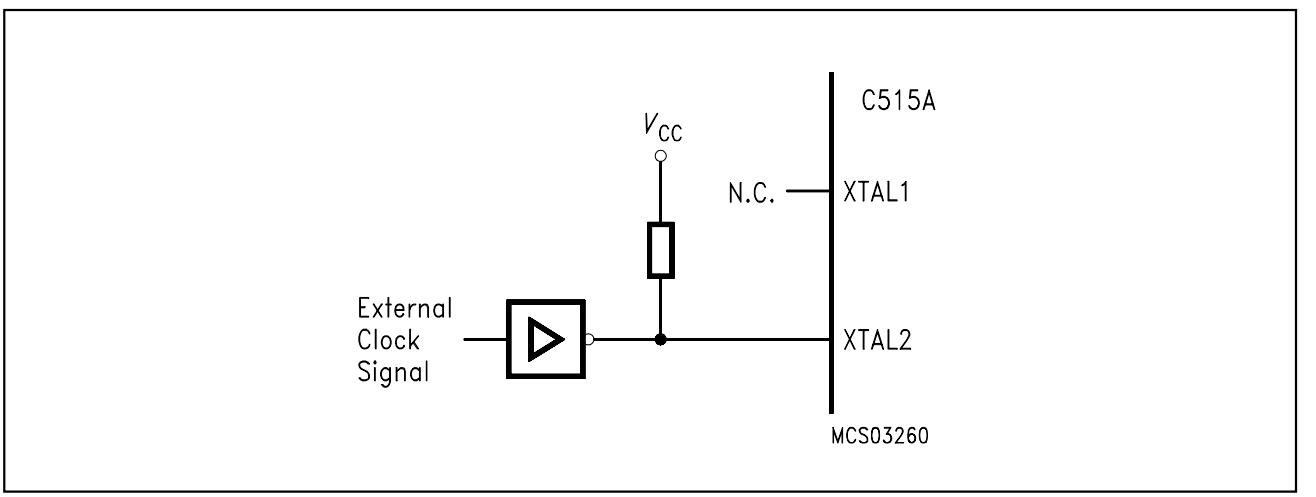


Figure 5-11
External Clock Source

5.5 System Clock Output

For peripheral devices requiring a system clock, the C515A provides a clock output signal derived from the oscillator frequency as an alternate output function on pin P1.6/CLKOUT. If bit CLK is set (bit 6 of special function register ADCON0), a clock signal with 1/12 of the oscillator frequency is gated to pin P1.6/CLKOUT. To use this function the port pin must be programmed to a one (1), which is also the default after reset.

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

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

The shaded bits are not used for clock output control.

Bit	Function
CLK	Clockout enable bit When set, pin P1.6/CLKOUT outputs the system clock which is 1/12 of the oscillator frequency.

The system clock is high during S3P1 and S3P2 of every machine cycle and low during all other states. Thus, the duty cycle of the clock signal is 1:6. Associated with a MOVX instruction the system clock coincides with the last state (S3) in which a $\overline{RD}$ or $\overline{WR}$ signal is active. A timing diagram of the system clock output is shown in **figure 5-12**.

Note : During slow-down operation the frequency of the CLKOUT signal is divided by 8.

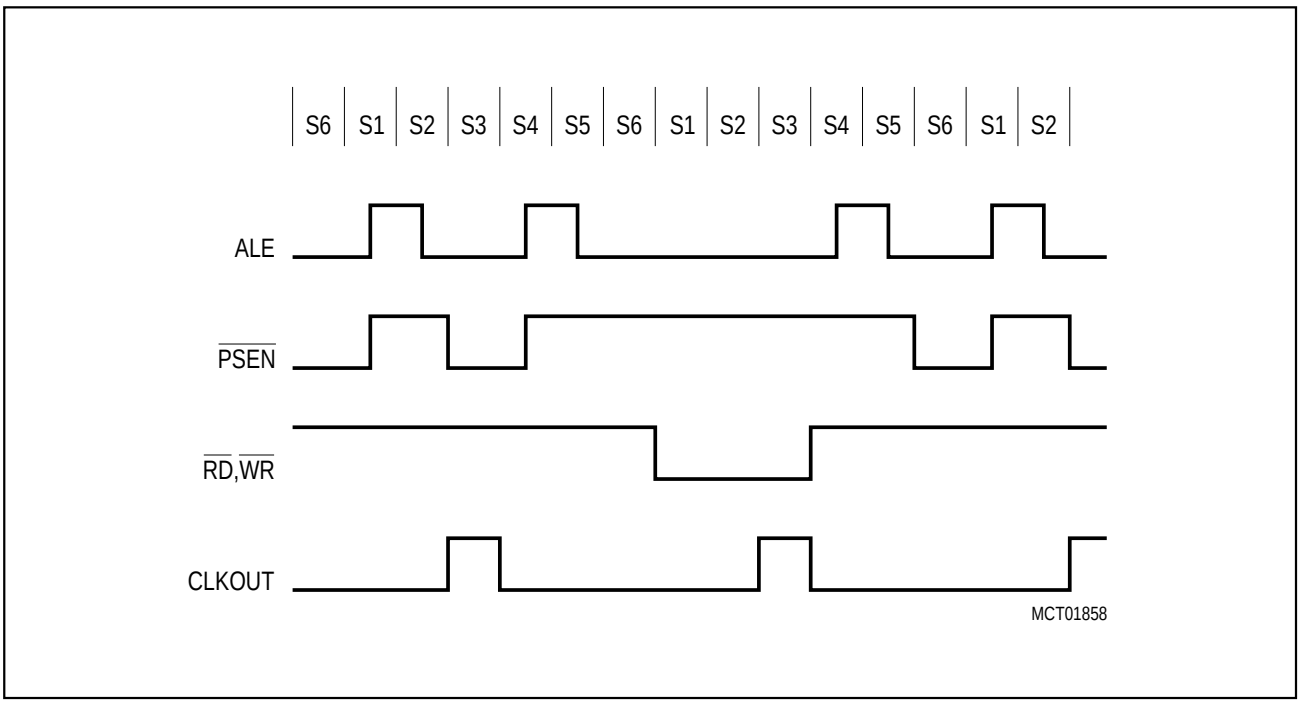


Figure 5-12
Timing Diagram - System Clock Output

6 On-Chip Peripheral Components

This chapter gives detailed information about all on-chip peripherals of the C515A except for the integrated interrupt controller, which is described separately in chapter 7.

6.1 Parallel I/O

The C515A has six 8-bit digital I/O ports and one 8-bit input port for analog/digital input. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-bidirectional I/O ports with internal pullup resistors. That means, when configured as inputs, ports 1 to 5 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

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

6.1.1 Port Structures

The C515A generally allows digital I/O on 48 lines grouped into 6 bidirectional C501 compatible 8-bit ports and one 8-bit analog/digital input port. Each port bit (except port 6) consists of a latch, an output driver and an input buffer. Read and write accesses to the I/O ports P0 to P5 are performed via their corresponding special function registers. Depending on the specific ports, multiple functions are assigned to the port pins. These alternate functions of the port pins are listed in **table 6-1**.

When port 6 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 6.4**). Port 6 lines may also be used as digital inputs. In this case they are addressed as an input port via SFR P6. Since port 6 has no internal latch, the contents of SFR P6 only depends on the levels applied to the input lines. It makes no sense to output a value to these input-only port by writing to the SFR P6. This will have no effect.

Table 6-1
Alternate Functions of Port 1 and 3

Port	Alternate Functions	Description
P1.0	$\overline{\text{INT3}}$ / CC0	External Interrupt 3 input / Capture/compare 0 input/output
P1.1	INT4 / CC1	External Interrupt 4 input / Capture/compare 1 input/output
P1.2	INT5 / CC2	External Interrupt 5 input / Capture/compare 2 input/output
P1.3	INT6 / CC3	External Interrupt 6 input / Capture/compare 3 input/output
P1.4	$\overline{\text{INT2}}$	External Interrupt 2 input
P1.5	T2EX	Timer 2 external reload/trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external count input
P3.0	RxD	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TxD	Serial port's transmitter data output (asynchronous) or data clock output (synchronous)
P3.2	$\overline{\text{INT0}}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external count input
P3.5	T1	Timer 1 external count input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe

6.1.2 Standard I/O Port Circuitry

Figure 6-1 shows a functional diagram of a typical bit latch and I/O buffer, which is the core of each of the six I/O-ports. The bit latch (one bit in the port's SFR) is represented as a type-D flip-flop, which will clock in a value from the internal bus in response to a "write-to-latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read-latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read-pin" signal from the CPU. Some instructions that read from a port (i.e. from the corresponding port SFR P0 to P4) activate the "read-latch" signal, while others activate the "read-pin" signal.

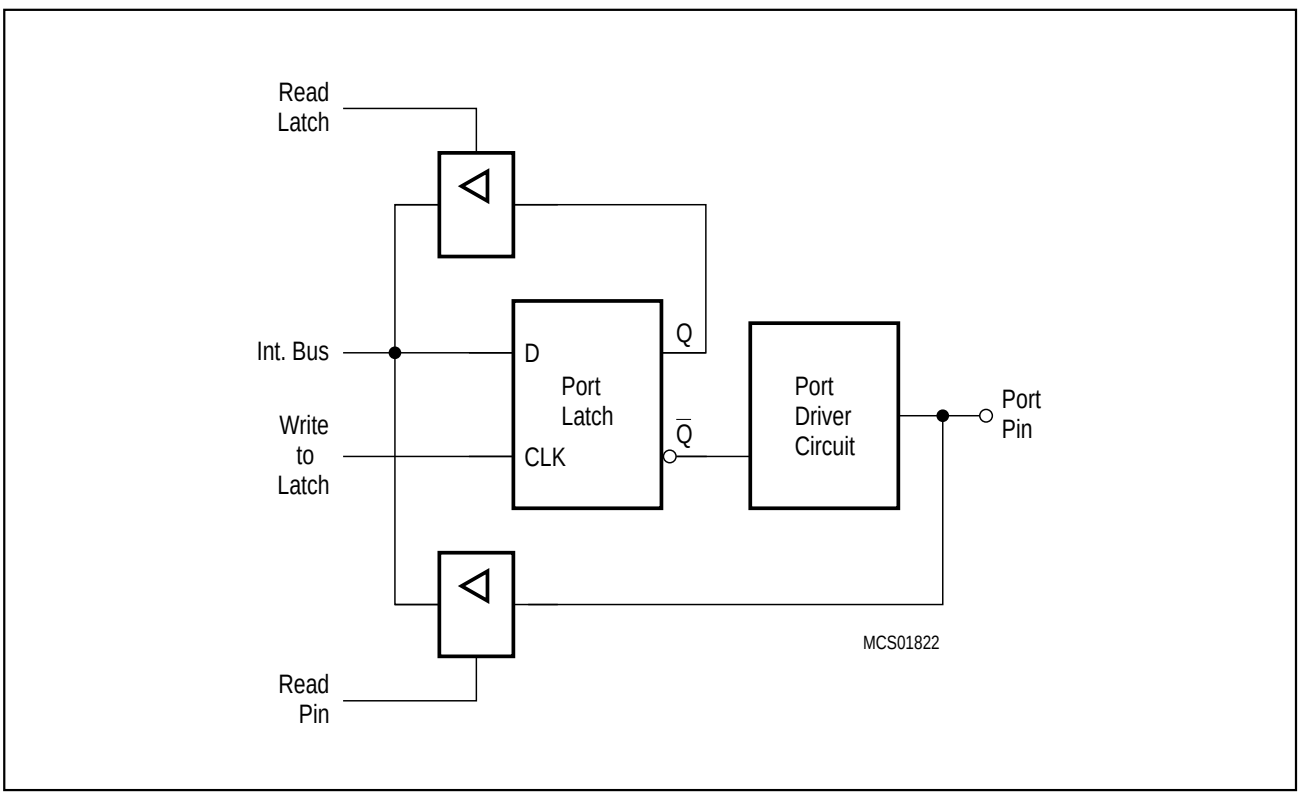


Figure 6-1
Basic Structure of a Port Circuitry

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

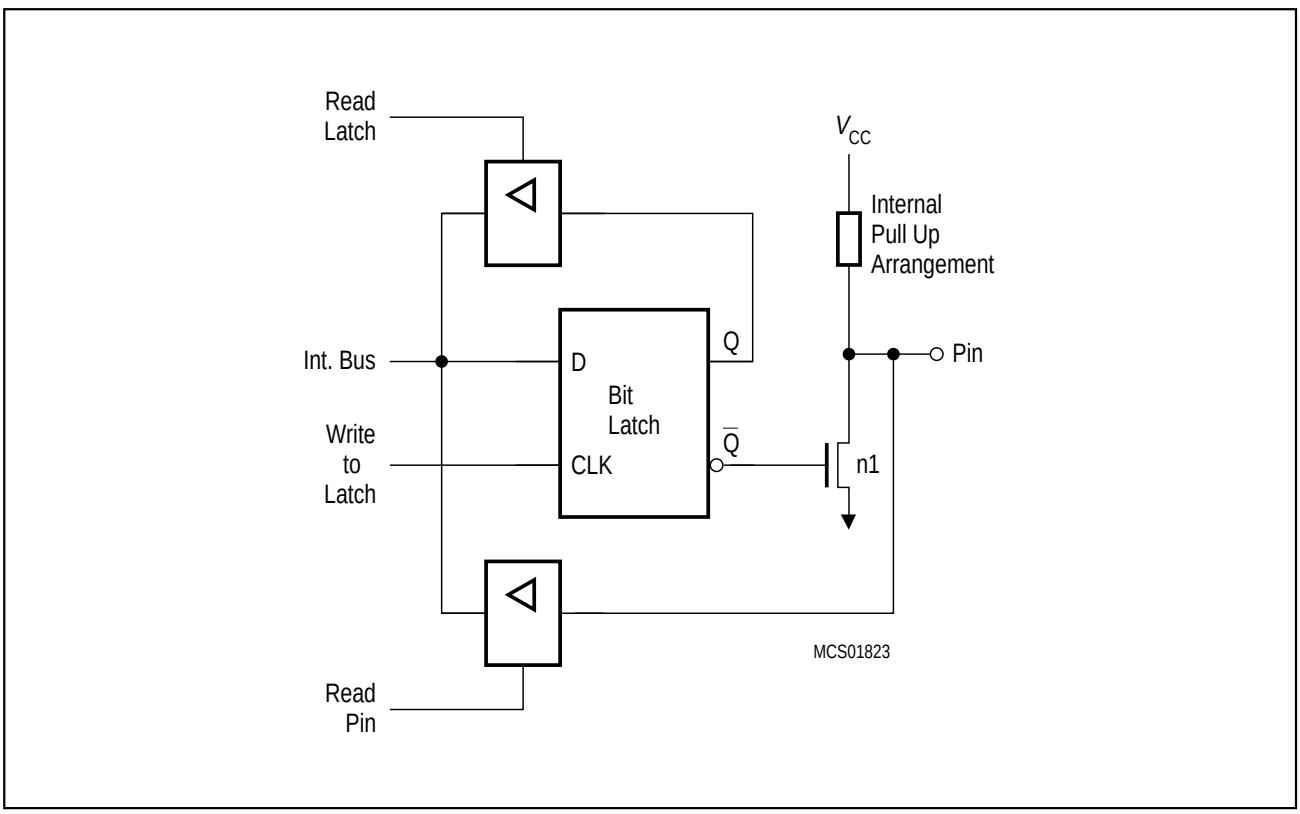


Figure 6-2
Basic Output Driver Circuit of Ports 1 to 5

6.1.2.1 Port 0 Circuitry

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

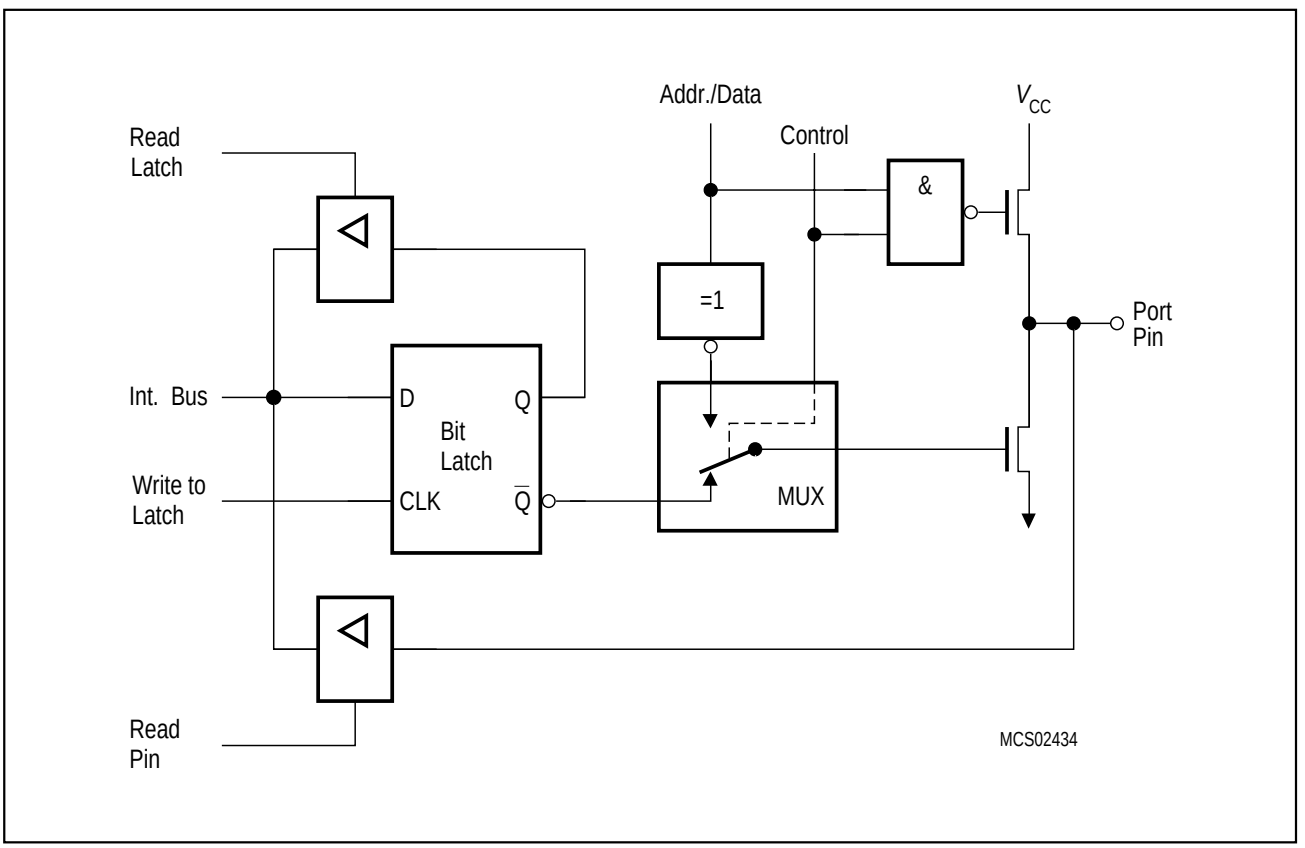


Figure 6-3
Port 0 Circuitry

6.1.2.2 Port 1, Port 3 to Port 5 Circuitry

The pins of ports 1, 3, 4, and 5 are multifunctional. They are port pins and also serve to implement special features as listed in **table 6-1**.

Figure 6-4 shows a functional diagram of a port latch with alternate function. To pass the alternate function to the output pin and vice versa, however, the gate between the latch and driver circuit must be open. Thus, to use the alternate input or output functions, the corresponding bit latch in the port SFR has to contain a one (1); otherwise the pulldown FET is on and the port pin is stuck at 0. After reset all port latches contain ones (1).

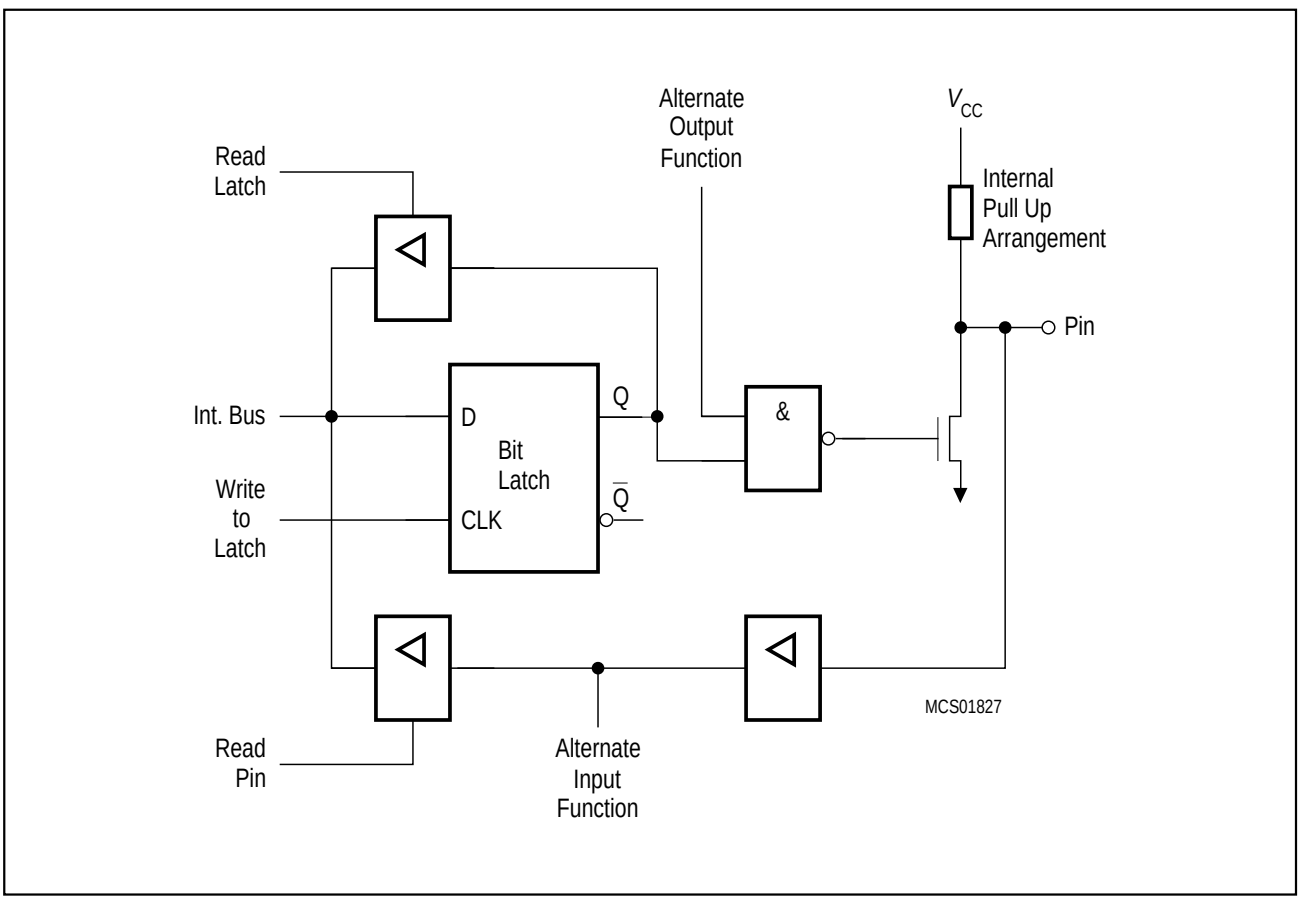


Figure 6-4
Ports 1, 3, 4, and 5

6.1.2.3 Port 2 Circuitry

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

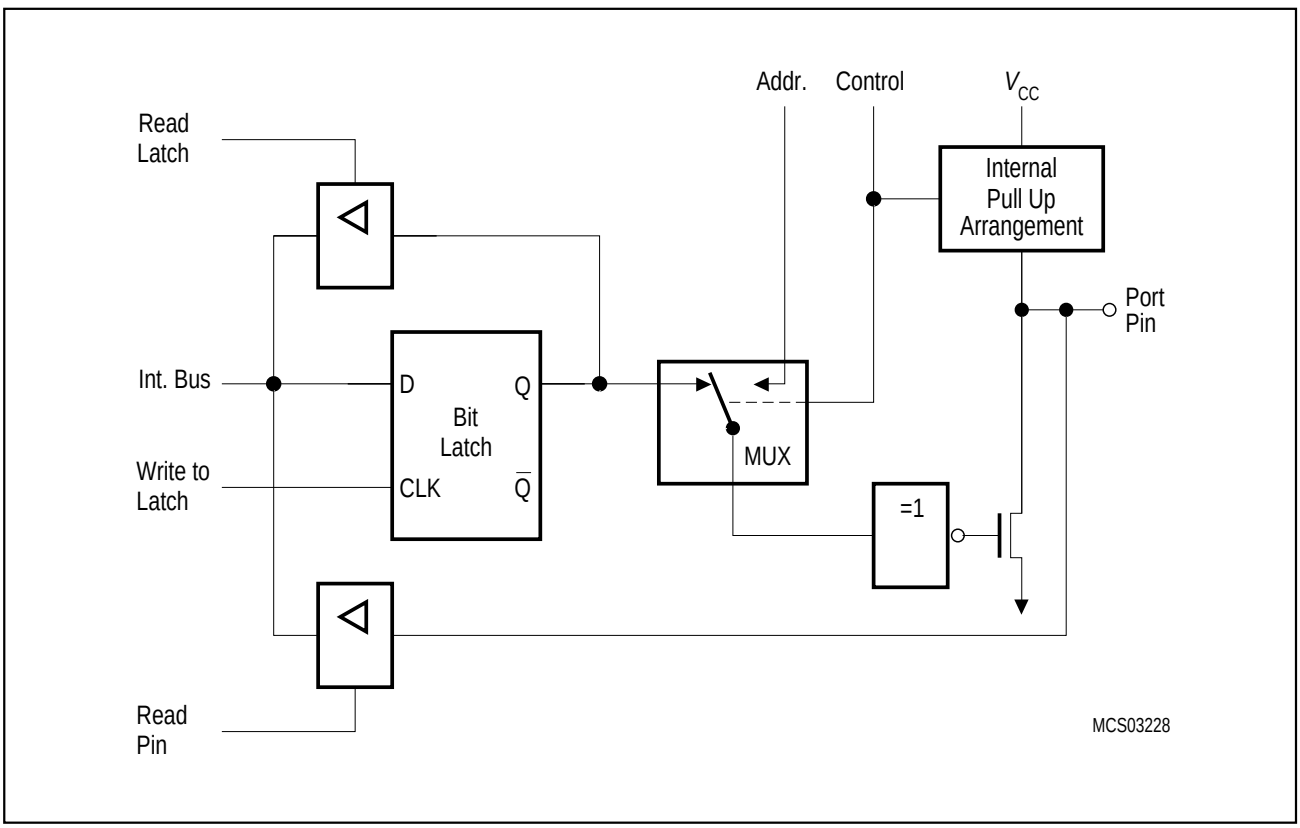


Figure 6-5
Port 2 Circuitry

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

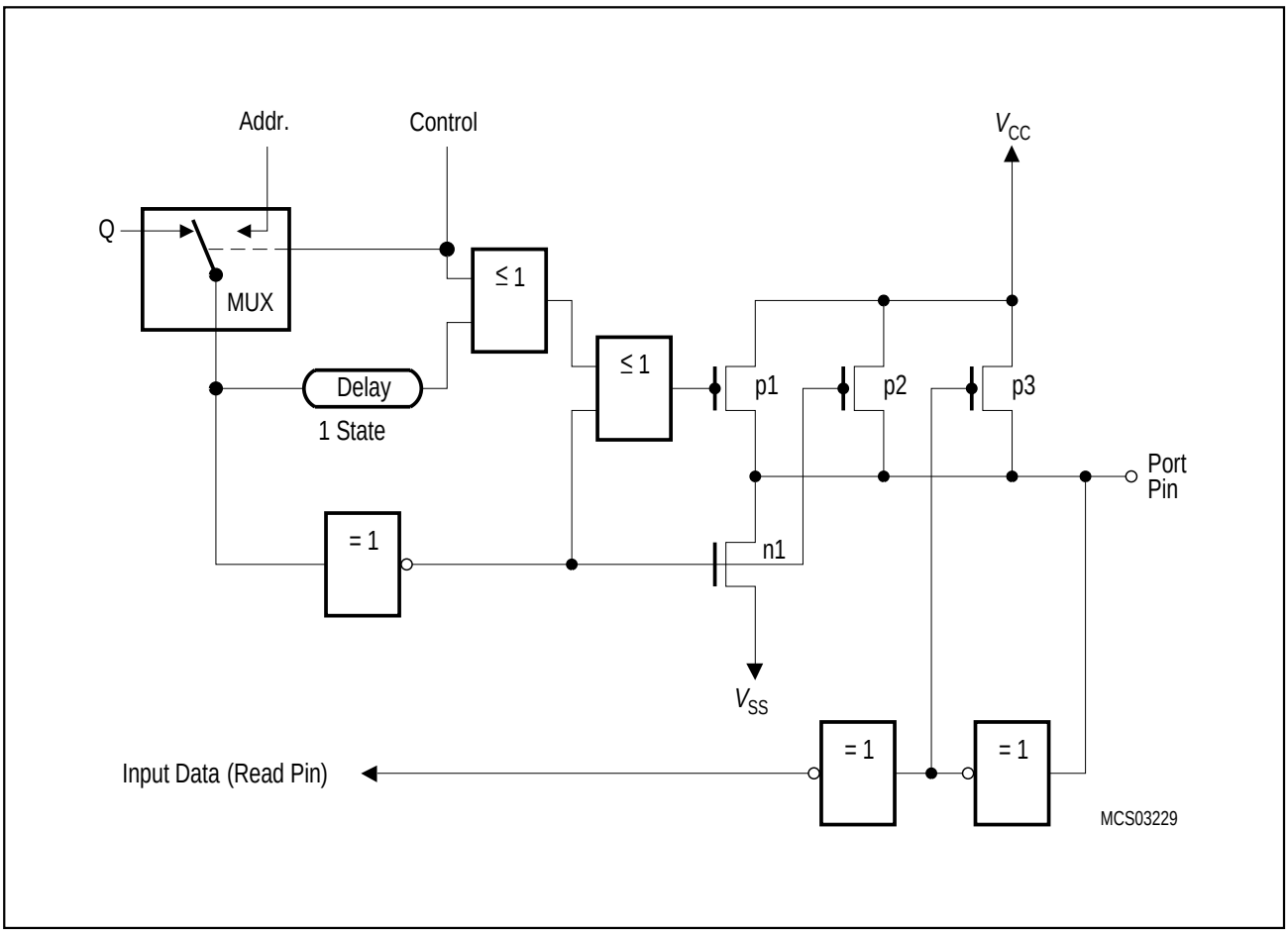


Figure 6-6
Port 2 Pull-up Arrangement

Port 2 in I/O function works similar to the standard port driver circuitry (section 6.1.2.4) whereas in address output function it works similar to Port 0 circuitry.

6.1.2.4 Detailed Output Driver Circuitry

In fact, the pullups mentioned before and included in **figure 6-2, 6-4** and **6-5** are pullup arrangements.

Figure 6-7 shows the detailed output driver (pullup arrangement) circuit of the the port 1 and 3 to 5 port lines. The basic circuitry of these ports is shown in **figure 6-4**. The pullup arrangement of these port lines has one n-channel pulldown FET and three pullup FETs:

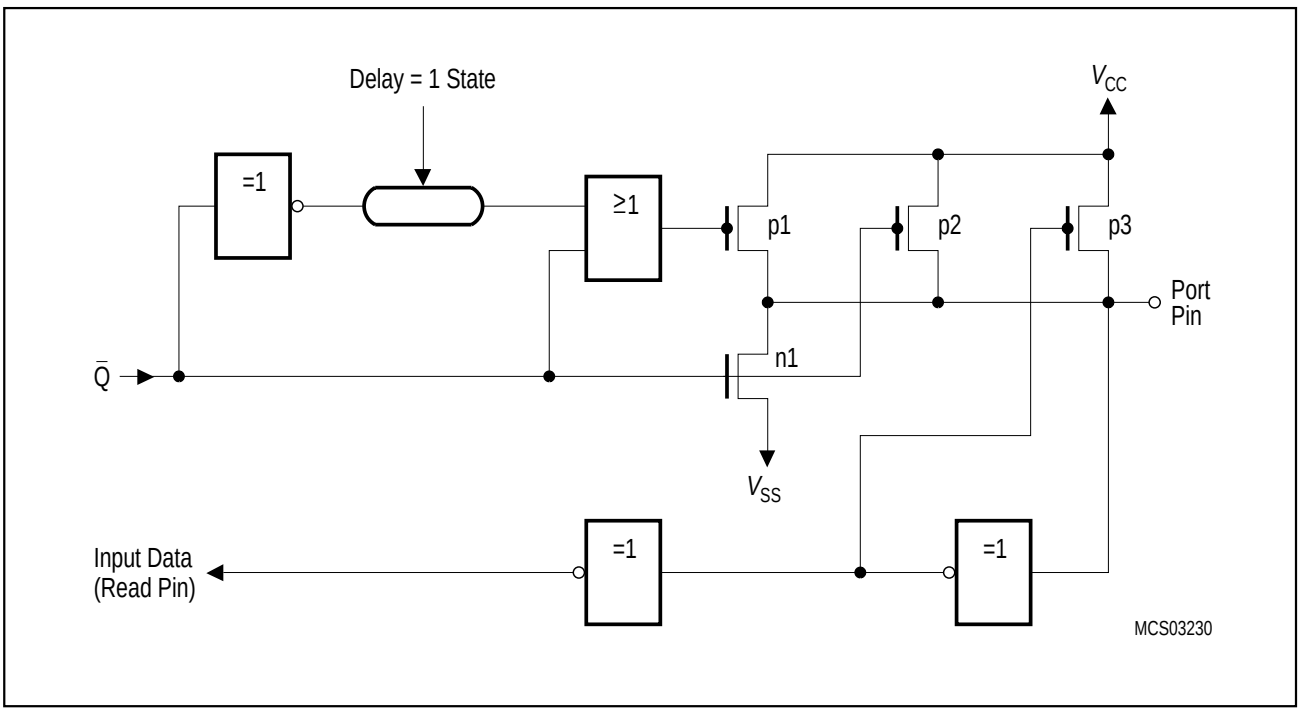


Figure 6-7
Driver Circuit of Ports 1, 3 to 6

- The **pulldown FET n1** is of n-channel type. It is a very strong driver transistor which is capable of sinking high currents (I_{OL}); it is only activated if a "0" is programmed to the port pin. A short circuit to V_{CC} must be avoided if the transistor is turned on, since the high current might destroy the FET. This also means that no “0“ must be programmed into the latch of a pin that is used as input.
- The **pullup FET p1** is of p-channel type. It is activated for one state (S1) if a 0-to-1 transition is programmed to the port pin, i.e. a "1" is programmed to the port latch which contained a "0". The extra pullup can drive a similar current as the pulldown FET n1. This provides a fast transition of the logic levels at the pin.
- The **pullup FET p2** is of p-channel type. It is always activated when a "1" is in the port latch, thus providing the logic high output level. This pullup FET sources a much lower current than p1; therefore the pin may also be tied to ground, e.g. when used as input with logic low input level.
- The **pullup FET p3** is of p-channel type. It is only activated if the voltage at the port pin is higher than approximately 1.0 to 1.5 V. This provides an additional pullup current if a logic high level shall be output at the pin (and the voltage is not forced lower than approximately 1.0 to 1.5 V). However, this transistor is turned off if the pin is driven to a logic low level, e.g when used as input. In this configuration only the weak pullup FET p2 is active, which sources

the current I_{IL} . If, in addition, the pullup FET p3 is activated, a higher current can be sourced (I_{TL}). Thus, an additional power consumption can be avoided if port pins are used as inputs with a low level applied. However, the driving capability is stronger if a logic high level is output.

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

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

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

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

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

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

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

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

When an instruction reads a value from a port pin (e.g. MOV A, P1) the port pin is actually sampled in state 5 phase 1 or phase 2 depending on port and alternate functions. **Figure 6-8** 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", e.g. when used as counter input. In this case an "edge" is detected when the sampled value differs from the value that was sampled the cycle before. Therefore, there must be met certain reqirements on the pulse length of signals in order to avoid signal "edges" not being detected. The minimum time period of high and low level is one machine cycle, which guarantees that this logic level is noticed by the port at least once.

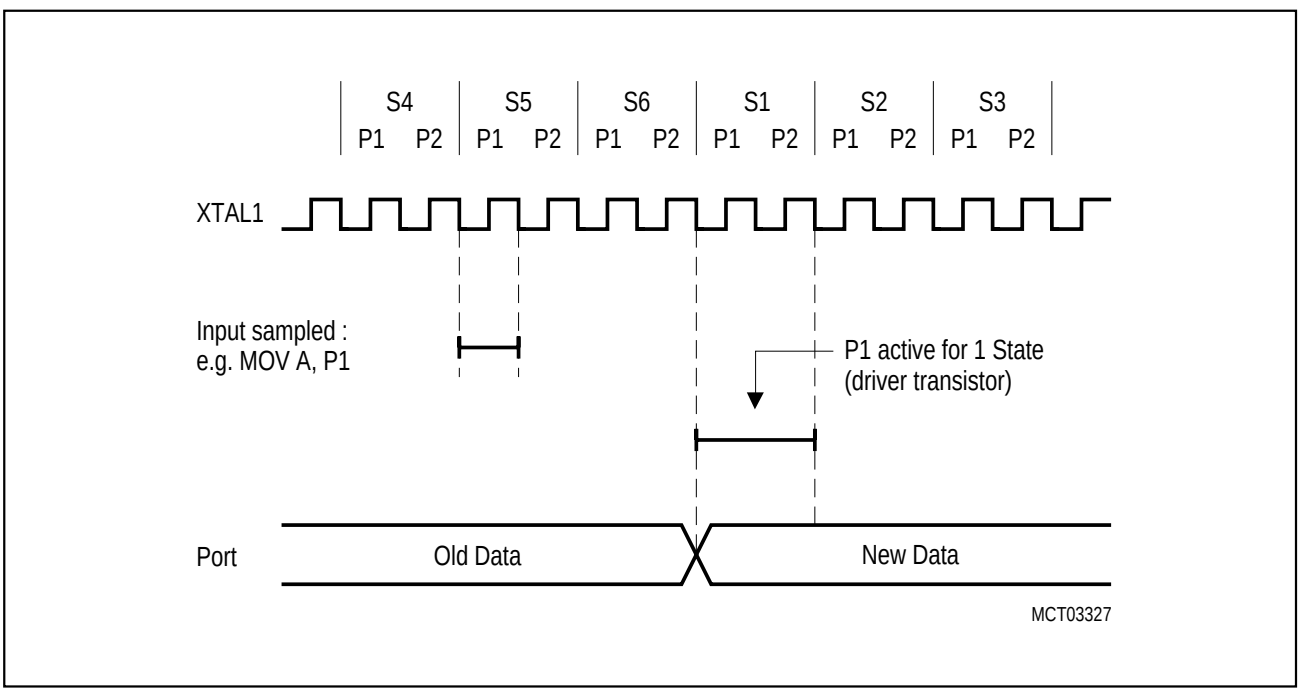


Figure 6-8
Port Timing

6.1.4 Port Loading and Interfacing

The output buffers of ports 1 to 5 can drive TTL inputs directly. The maximum port load which still guarantees correct logic output levels can be be looked up in the DC characteristics in the Data Sheet of the C515A or in chapter 10 of this User's Manual. The corresponding parameters are V_{OL} and V_{OH}.

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

When used as inputs it must be noted that the ports 1 to 5 are not floating but have internal pullup transistors. The driving devices must be capable of sinking a sufficient current if a logic low level shall be applied to the port pin (the parameters I_{TL} and I_{IL} in the DC characteristics specify these currents). Port 0 as well as port 1 programmed to analog input function, however, have floating inputs when used for digital input.

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

Some port-reading instructions read the latch and others read the pin. The instructions reading the latch rather than the pin read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write"- instructions, which are listed in **table 6-2**. If the destination is a port or a port pin, these instructions read the latch rather than the pin. Note that all other instructions which can be used to read a port, exclusively read the port pin. In any case, reading from latch or pin, resp., is performed by reading the SFR P0, P2 and P3; for example, "MOV A, P3" reads the value from port 3 pins, while "ANL P3, #0AAH" reads from the latch, modifies the value and writes it back to the latch.

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

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

Instruction	Function
ANL	Logic AND; e.g. ANL P1, A
ORL	Logic OR; e.g. ORL P2, A
XRL	Logic exclusive OR; e.g. XRL P3, A
JBC	Jump if bit is set and clear bit; e.g. JBC P1.1, LABEL
CPL	Complement bit; e.g. CPL P3.0
INC	Increment byte; e.g. INC P4
DEC	Decrement byte; e.g. DEC P5
DJNZ	Decrement and jump if not zero; e.g. DJNZ P3, LABEL
MOV Px.y,C	Move carry bit to bit y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

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

6.2 Timers/Counters

The C515A contains three general purpose 16-bit timers/counters, timer 0, 1, 2, and the compare imer which are useful in many applications for timing and counting.

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

In "counter" function, the register is incremented in response to a 1-to-0 transition (falling edge) at its corresponding external input pin, T0 or T1 (alternate functions of P3.4 and P3.5, resp.). In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

6.2.1 Timer/Counter 0 and 1

Timer / counter 0 and 1 of the C515A 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) which may be combined to one timer configuration depending on the mode that is established. The functions of the timers are controlled by two special function registers TCON and TMOD.

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

6.2.1.1 Timer/Counter 0 and 1 Registers

Totally six special function registers control the timer/counter 0 and 1 operation :

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

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

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

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

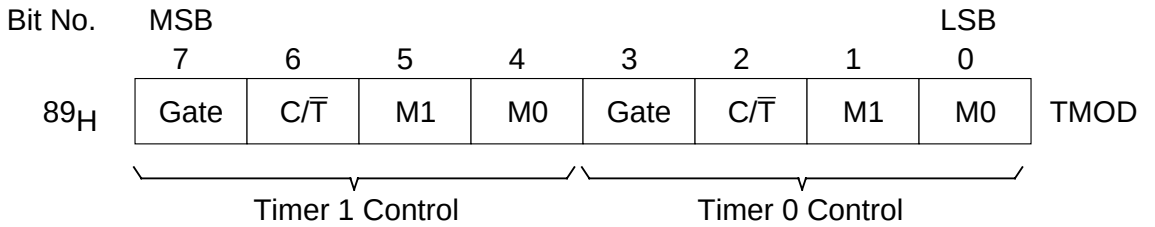
Special Function Register TCON (Address 88_H) **Reset Value : 00_H**

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

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

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

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



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

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

6.2.1.2 Mode 0

Putting either timer/counter 0,1 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 $\overline{INT0}$ = 1 (setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements). TR0 is a control bit in the special function register TCON; Gate is in TMOD.

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

Mode 0 operation is the same for timer 0 as for timer 1. Substitute TR0, TF0, TH0, TL0 and INT0 for the corresponding timer 1 signals in **figure 6-9**. There are two different gate bits, one for timer 1 (TMOD.7) and one for timer 0 (TMOD.3).

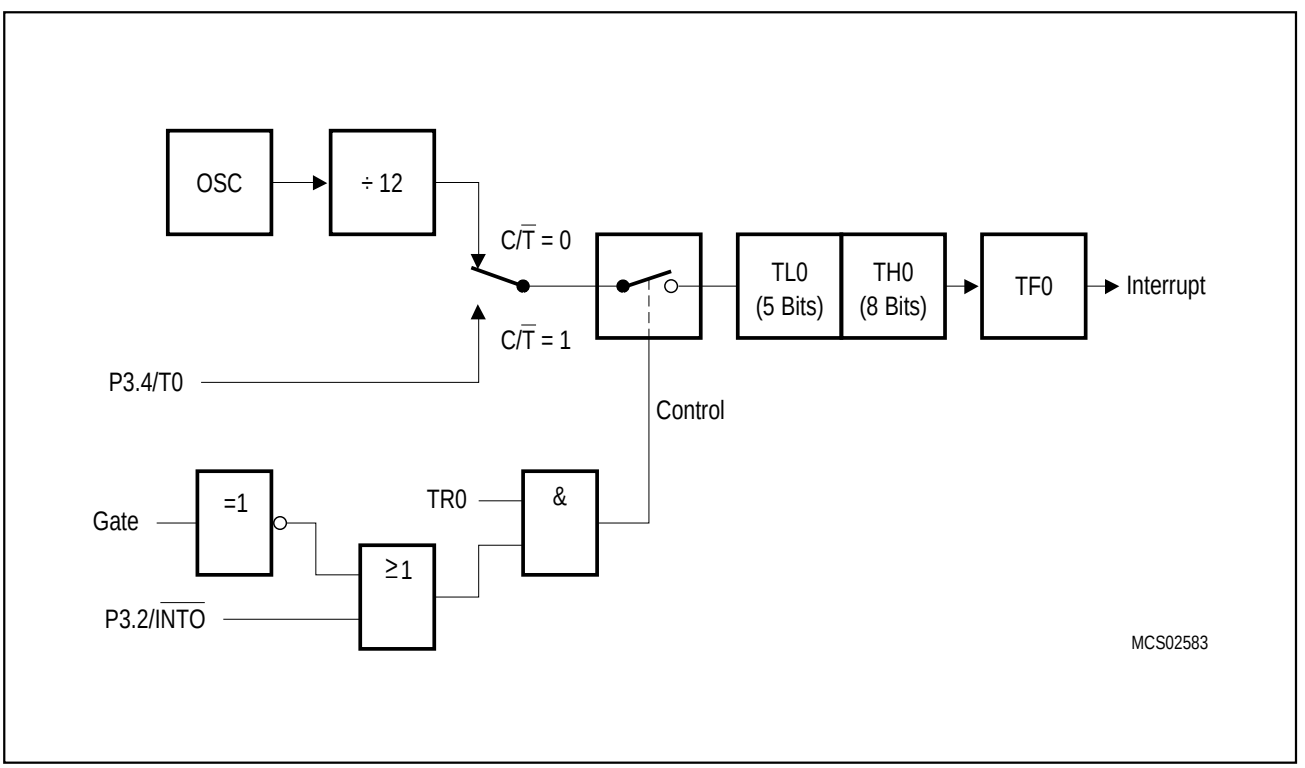


Figure 6-9
Timer/Counter 0, Mode 0: 13-Bit Timer/Counter

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**.

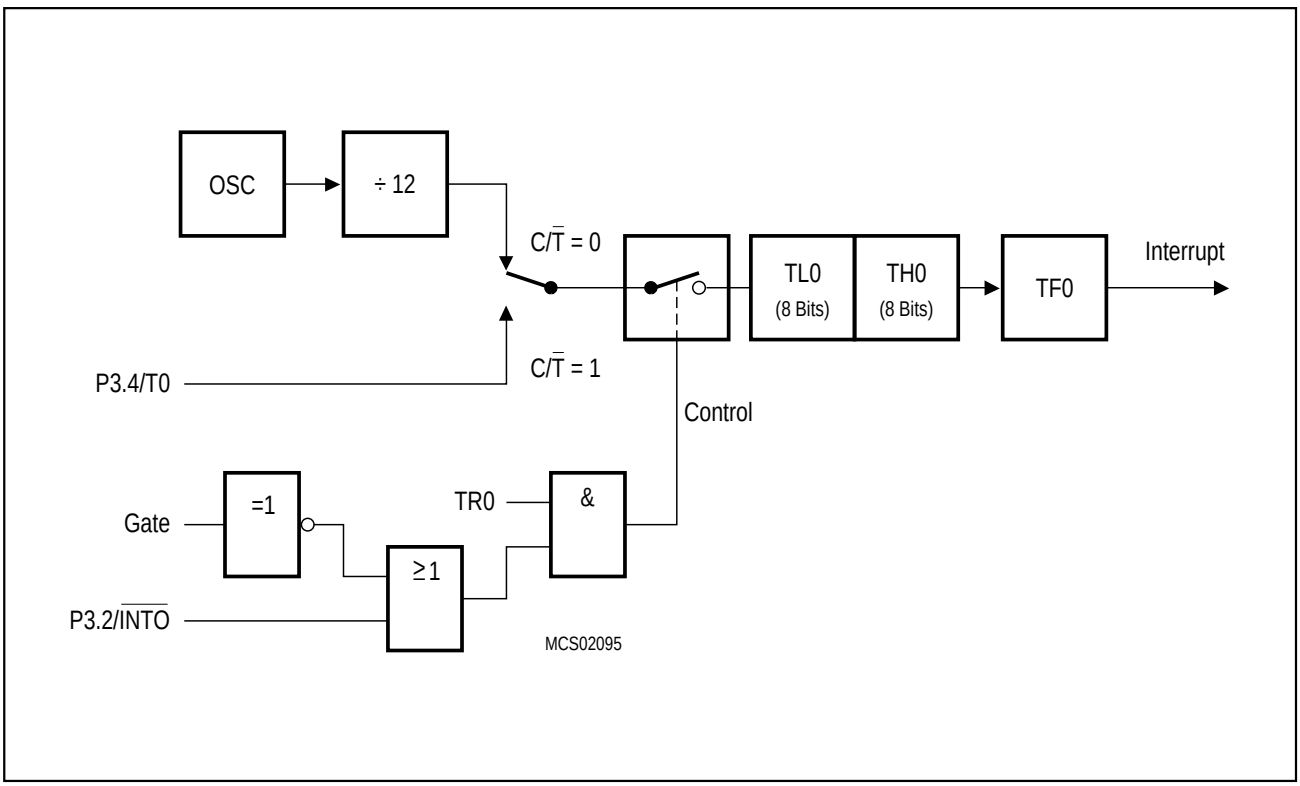


Figure 6-10
Timer/Counter 0, Mode 1: 16-Bit Timer/Counter

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.

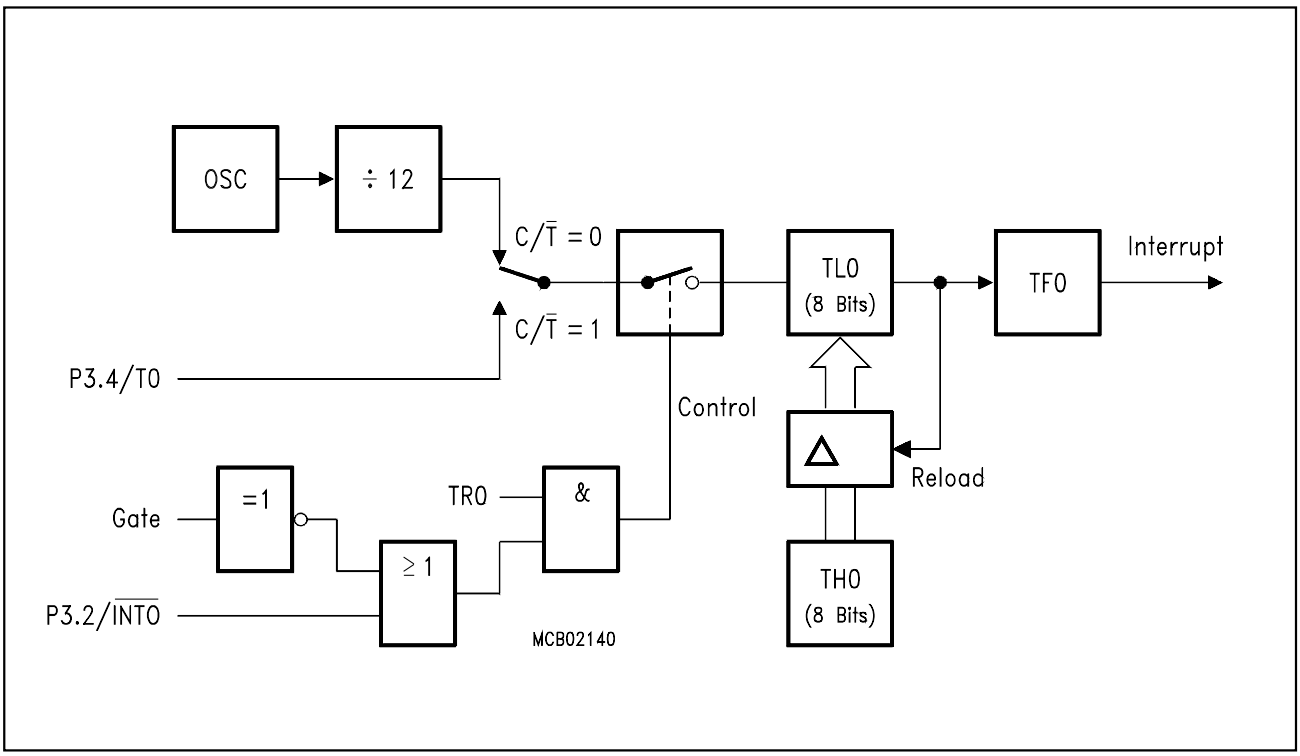


Figure 6-11
Timer/Counter 0,1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

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 seperate counters. The logic for mode 3 on timer 0 is shown in **figure 6-12**. TL0 uses the timer 0 control bits: C/$\overline{T}$, Gate, TR0, $\overline{INT0}$ and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the "timer 1" interrupt.

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

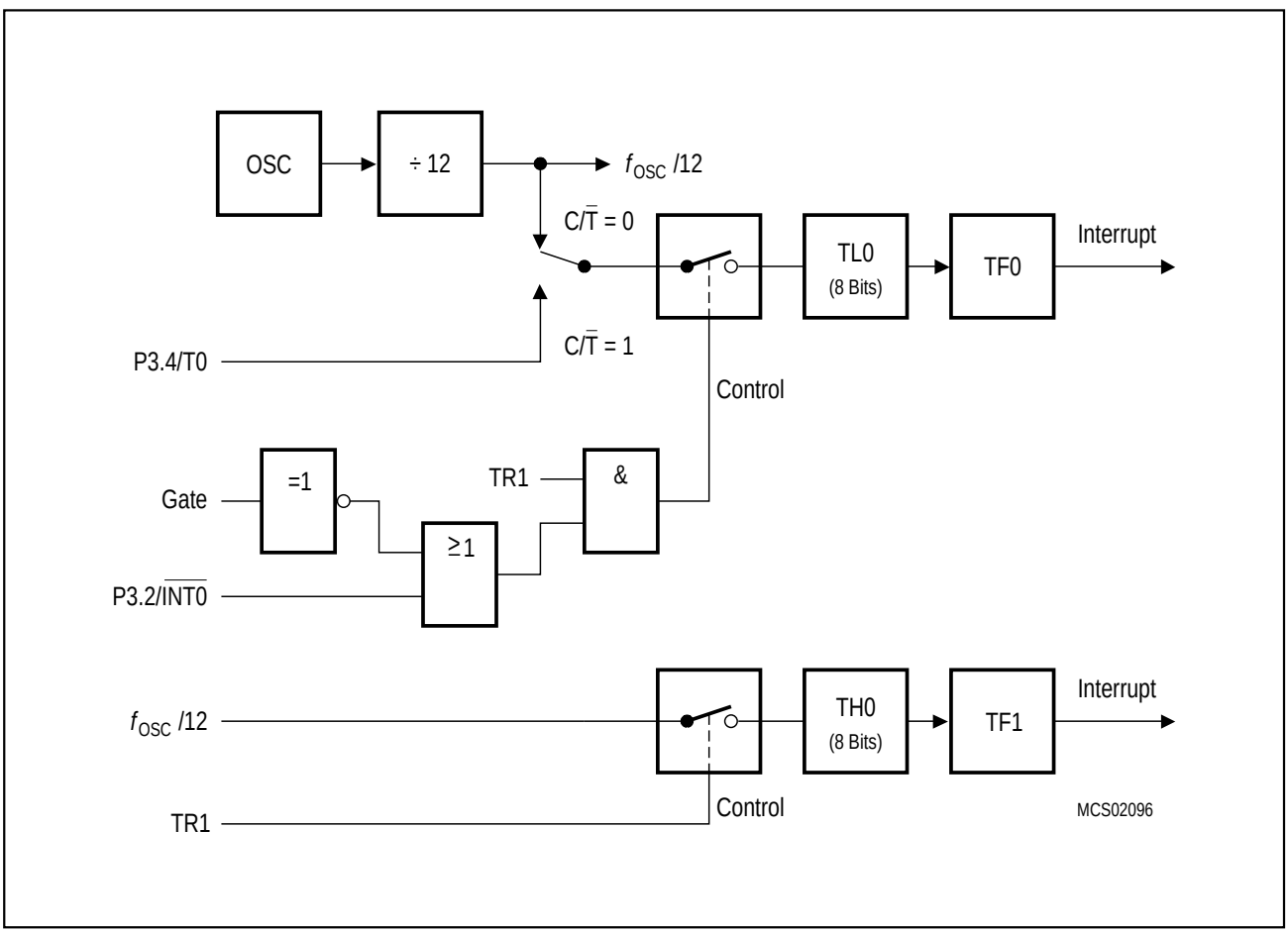


Figure 6-12
Timer/Counter 0, Mode 3: Two 8-Bit Timers/Counters

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

The timer 2 with additional compare/capture/reload features is one of the most powerful peripheral units of the C515A. lt can be used for all kinds of 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 (ignition/injection-control, anti-lock-brake ...) as weil as industrial applications (DC-, three-phase AC- and stepper-motor control, frequency generation, digital-to-analog conversion, process control ...). Please note that this timer is not equivalent to timer 2 of the C501.

The C515A timer 2 in combination with the compare/capture/reload registers allows the following operating modes:

- Compare : up to 4 PWM signals with 65535 steps at maximum, and 500 ns resolution
- Capture : up to 4 high speed capture inputs with 500 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 used for timer 2 control are located as multifunctional port functions at port 1 (see **table 6-3**).

Table 6-3
Alternate Port Functions of Timer 2

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

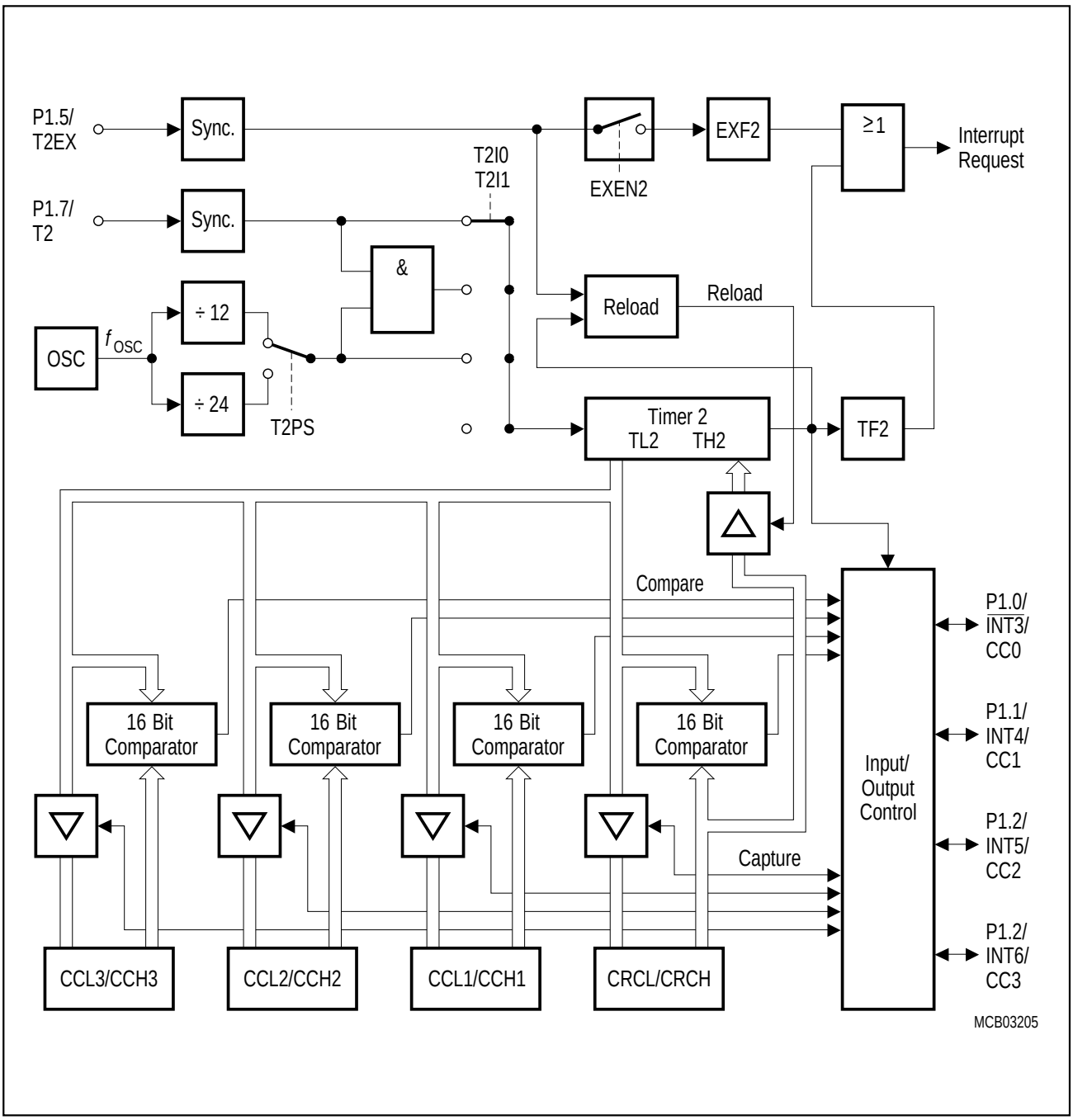


Figure 6-13
Timer 2 Block Diagram

6.2.2.1 Timer 2 Registers

This chapter describes all timer 2 related special function registers of timer 2. The interrupt related SFRs are also included in this section. **Table 6-4** summarizes all timer 2 SFRs.

Table 6-4
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
CCL1	Compare / capture register 1, low byte	$C2_H$
CCH1	Compare / capture register 1, high byte	$C3_H$
CCL2	Compare / capture register 2, low byte	$C4_H$
CCH2	Compare / capture register 2, high byte	$C5_H$
CCL3	Compare / capture register 3, low byte	$C6_H$
CCH3	Compare / capture register 3, high byte	$C7_H$
IEN0	Interrupt enable register 0	$A8_H$
IEN1	Interrupt enable register 1	$B8_H$
IRCON	Interrupt control register	$C0_H$

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

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

Bit No.	MSB 7	6	5	4	3	2	1	LSB 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 bit is not used in controlling timer/counter 2.

Bit	Function
T2PS	Prescaler select bit When set, timer 2 is clocked in the “timer“ or “gated timer“ function with 1/24 of the oscillator frequency. When cleared, timer 2 is clocked with 1/12 of the oscillator frequency. T2PS must be 0 for the counter operation of timer 2.
I3FR	External interrupt 3 falling/rising edge flag Used for capture function in combination with register CRC. If set, a capture to register CRC (if enabled) will occur on a positive transition at pin P1.0 / $\overline{INT3}$ / CC0. If I3FR is cleared, a capture will occur on a negative transition.
T2R1 T2R0	Timer 2 reload mode selection

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

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

T2I1	**T2I0**	**Function**
0	0	No input selected, timer 2 stops
0	1	Timer function : input frequency = $f_{osc}/12$ (T2PS = 0) or $f_{osc}/24$ (T2PS = 1)
1	0	Counter function : external input signal at pin P1.7 / T2
1	1	Gated timer function : input controlled by pin P1.7 / T2

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

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

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

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

	MSB							LSB	
Bit No.	AF_H	AE_H	AD_H	AC_H	AB_H	AA_H	$A9_H$	$A8_H$	
$A8_H$	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	IEN0
Bit No.	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1
Bit No.	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	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.
EXEN2	Timer 2 external reload interrupt enable If EXEN2 = 0, the timer 2 external reload interrupt is disabled. If EXEN2 = 1, the timer 2 external reload interrupt is enabled. The external reload function is not affected by EXEN2.
EXF2	Timer 2 external reload flag EXF2 is set when a reload is caused by a falling edge on pin T2EX while EXEN2 = 1. If ET2 in IEN0 is set (timer 2 interrupt enabled), EXF2 = 1 will cause an interrupt. EXF2 can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.
TF2	Timer 2 overflow flag Set by a timer 2 overflow and must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.

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

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

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

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

COCAH2 COCAL2	Compare/capture mode for CC register 2

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

COCAH1 COCAL1	Compare/capture mode for CC register 1

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

COCAH0 COCAL0	Compare/capture mode for CRC register

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

6.2.2.2 Timer 2 Operation

The timer 2, which is a 16-bit-wide register, can operate as timer, event counter, or gated timer. The detailed operation is described below.

Timer Mode

In timer function, the count rate is derived from the oscillator frequency. A prescaler offers the possibility of selecting a count rate of 1/12 or 1/24 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. lf T2PS is cleared, the input frequency is 1/12 of the oscillator frequency. if T2PS is set, the 2:1 prescaler gates 1/24 of the oscillator frequency to the timer.

Gated Timer Mode

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

Event Counter Mode

In the counter function, the timer 2 is incremented in response to a 1-to-0 transition at its corresponding external input pin P1.7 / T2 In this function, the external input is sampled every machine cycle. When the sampled inputs show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the timer register in the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for at least one full machine cycle.

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

In either case, no matter whether timer 2 is configured as timer, event counter, or gated timer, a rolling-over of the count from all 1’s to all 0’s sets the timer overflow flag TF2 in SFR IRCON, which can generate an interrupt.

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

Reload of Tirner 2

The reload mode for timer 2 is selected by bits T2R0 and T2R1 in SFR T2CON. **Figure 6-14** shows the configuration of timer 2 in reload mode.

Mode 0 : When timer 2 rolls over from all l'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 register, which is preset by software. The reload will happen in the same machine cycle in which TF2 is set, thus overwriting the count value 0000_H.

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

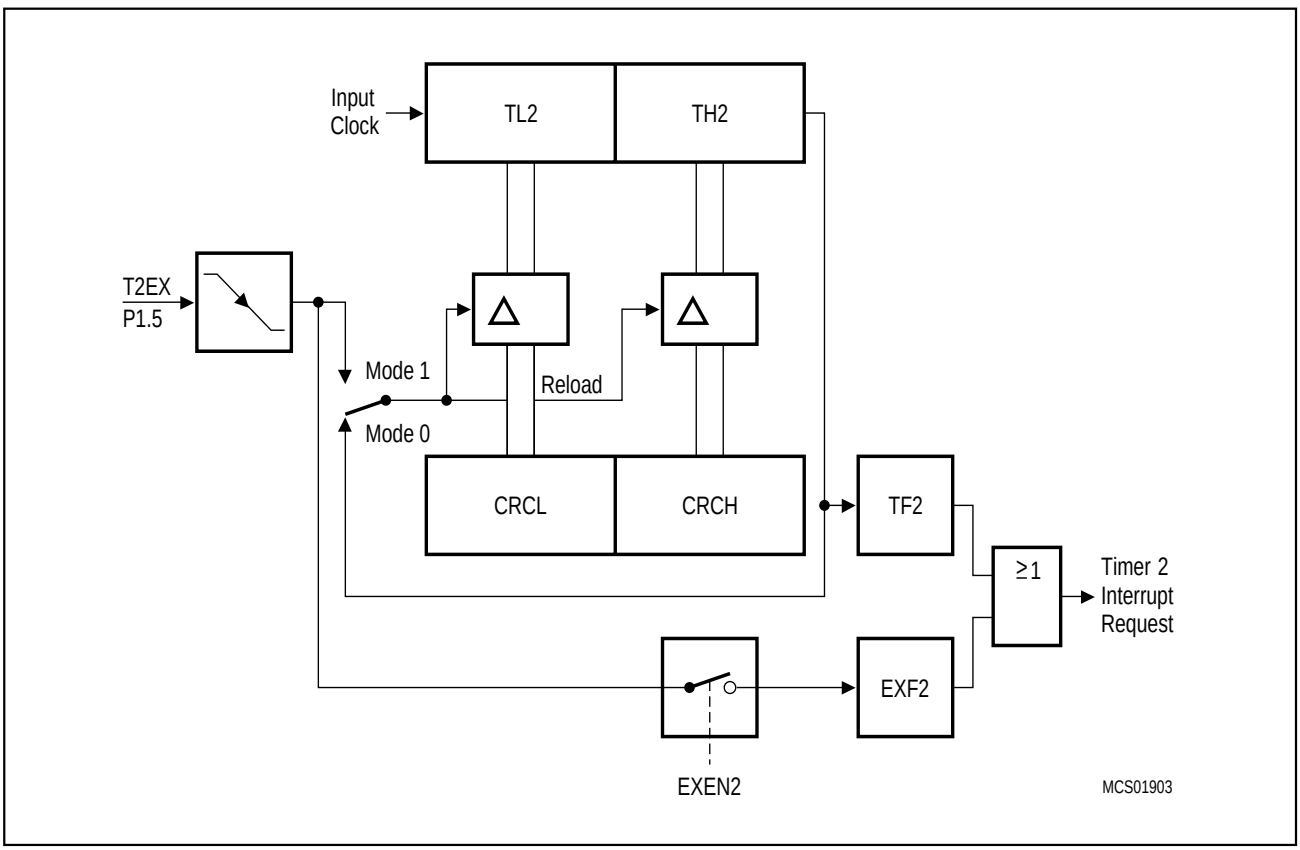


Figure 6-14
Timer 2 in Reload Mode

6.2.2.3 Compare Function of Registers CRC, CC1 to CC3

The compare function of a timer/register combination 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 'time stamp' at which a dedicated output reacts in a predefined way (either with a positive or negative transition). Variation of this 'time stamp' somehow changes the wave of a rectangular output signal at a port pin. This may - as a variation of the duty cycle of a periodic signal - be used for pulse width modulation as well as for a continually controlled generation of any kind of square wave forms. 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, CC1 to CC3 are multifunctional as they additionally provide a capture, compare or reload capability (the 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, upon matching the timer and compare register contents, the output signal changes from low to high. lt 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 **6-17** illustrate the function of compare mode 0.

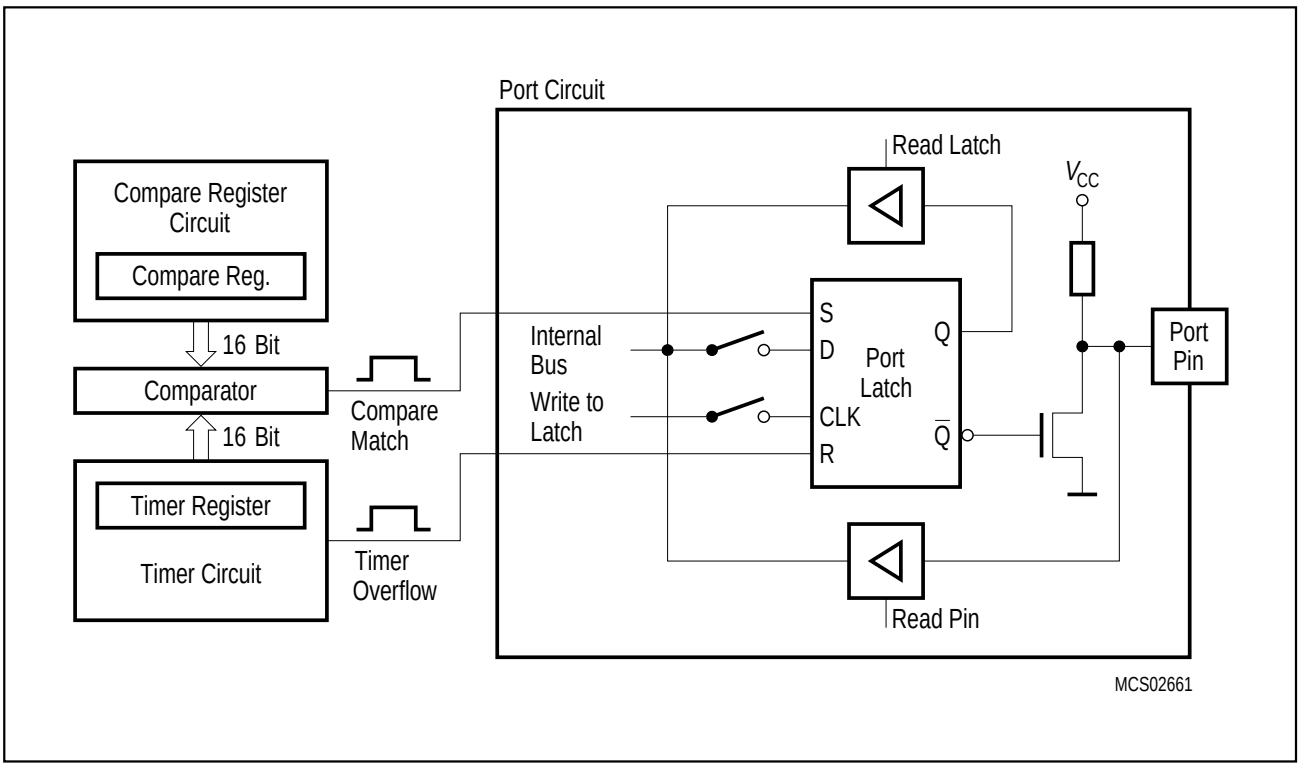


Figure 6-15
Port Latch in Compare Mode 0

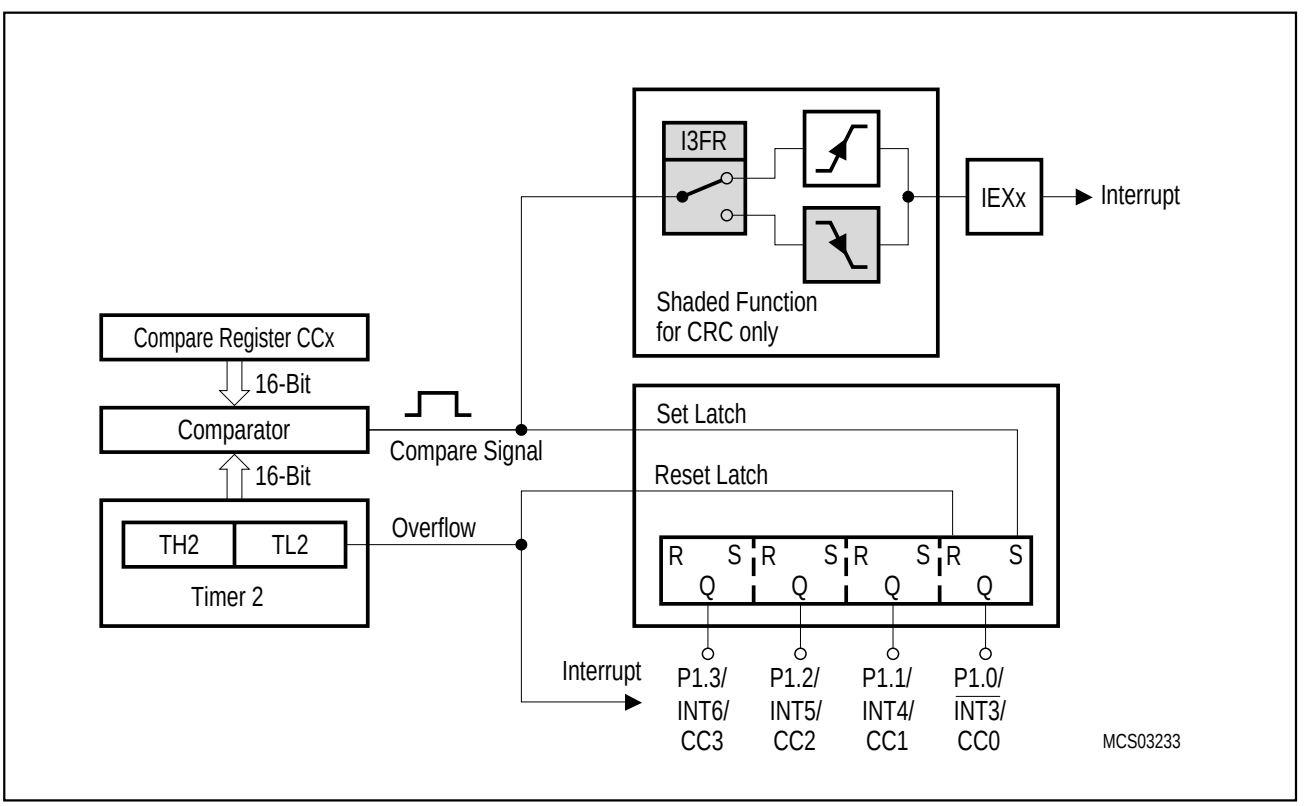


Figure 6-16
Timer 2 with Registers CCx in Compare Mode 0

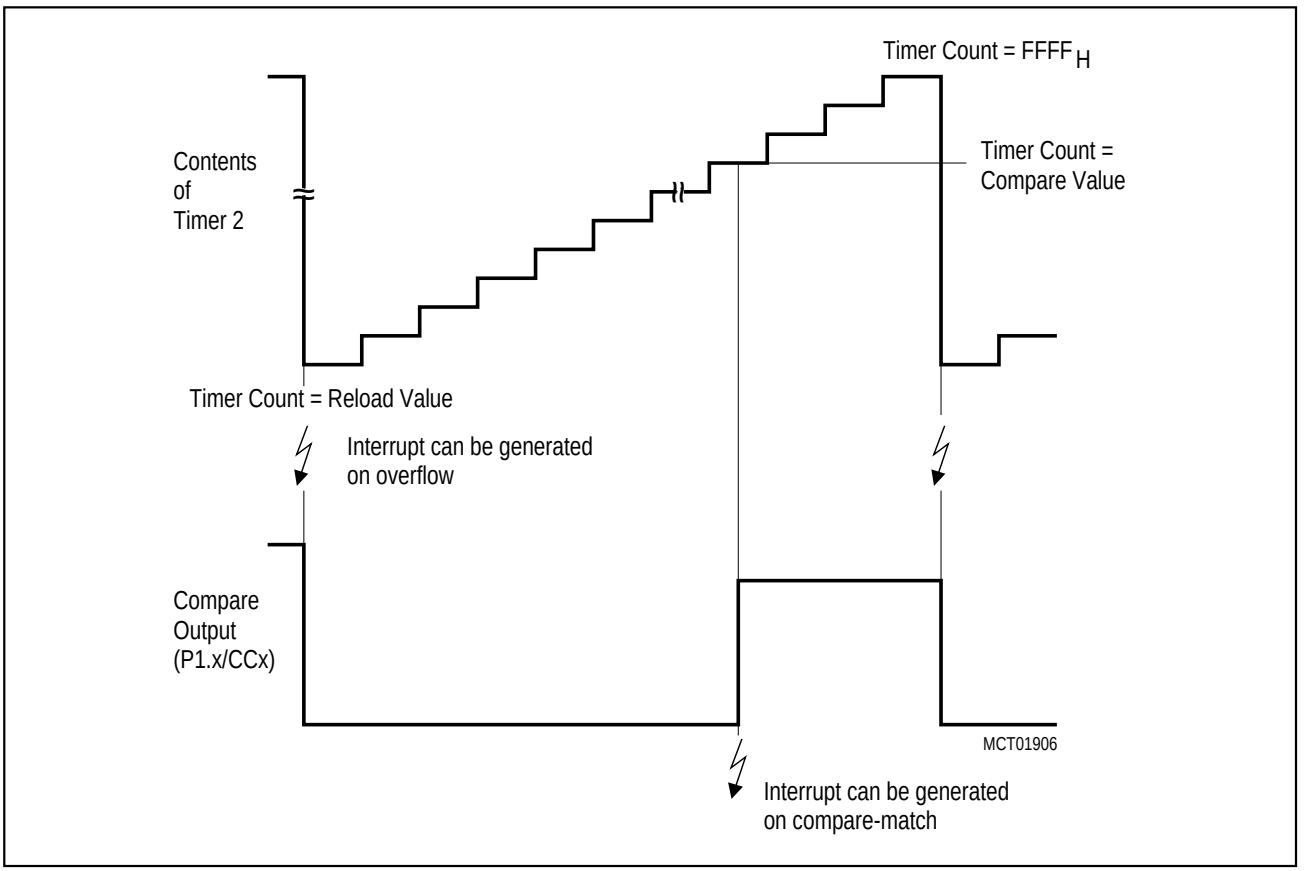


Figure 6-17
Function of Compare Mode 0

6.2.2.3.2 Modulation Range in Compare Mode 0

Generally it can be said that for every PWM generation in compare mode 0 with n-bit wide compare registers there are 2^n different settings for the duty cycle. Starting with a constant low level (0%duty cycle) as the first setting, the maximum possible duty cycle then would 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. In a timer 2/CCx register configuration in compare mode 0 this spike is divided into two halves: one at the beginning when the contents of the compare register is equal to the reload value of the timer; the other half when the compare register is equal to the maximum value of the timer register (here: $FFFF_H$). Please refer to **figure 6-18** where the maximum and minimum duty cycle of a compare output signal is illustrated. Timer 2 is incremented with the machine clock (fosc/12), thus at 24-MHz operating frequency, these spikes are both approx. 250 ns long.

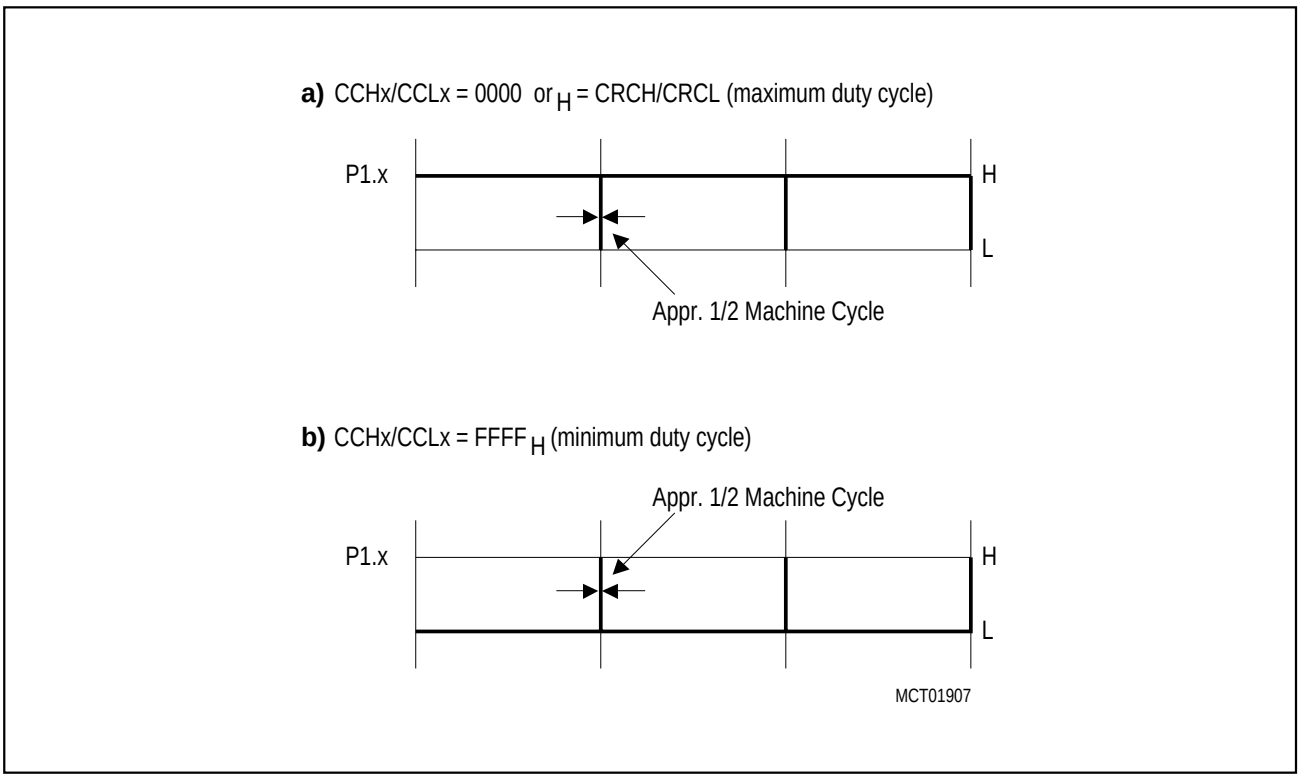


Figure 6-18
Modulation Range of a PWM Signal, generated with a Timer 2/CCx Register Combination in Compare Mode 0*

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

Example:

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

Restriction of modulation range = 1 / 256 x 2 x 100% = 0.195%

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

6.2.2.3.3 Compare Mode 1

In compare mode 1, the software adaptively determines the transition of the output signal. lt is commonly used when output signals are not related to a constant signal perlod (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.

lf compare mode 1 is enabled and the software writes to the appropriate output latch at the port, the new value will not appear at the output pin until the next compare match occurs. Thus, one can choose whether the output signal is to make a new transition (1-to-0 or 0-to-1, depending on the actual pinlevel) or 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 - as usual - read the pin of the corresponding compare output.

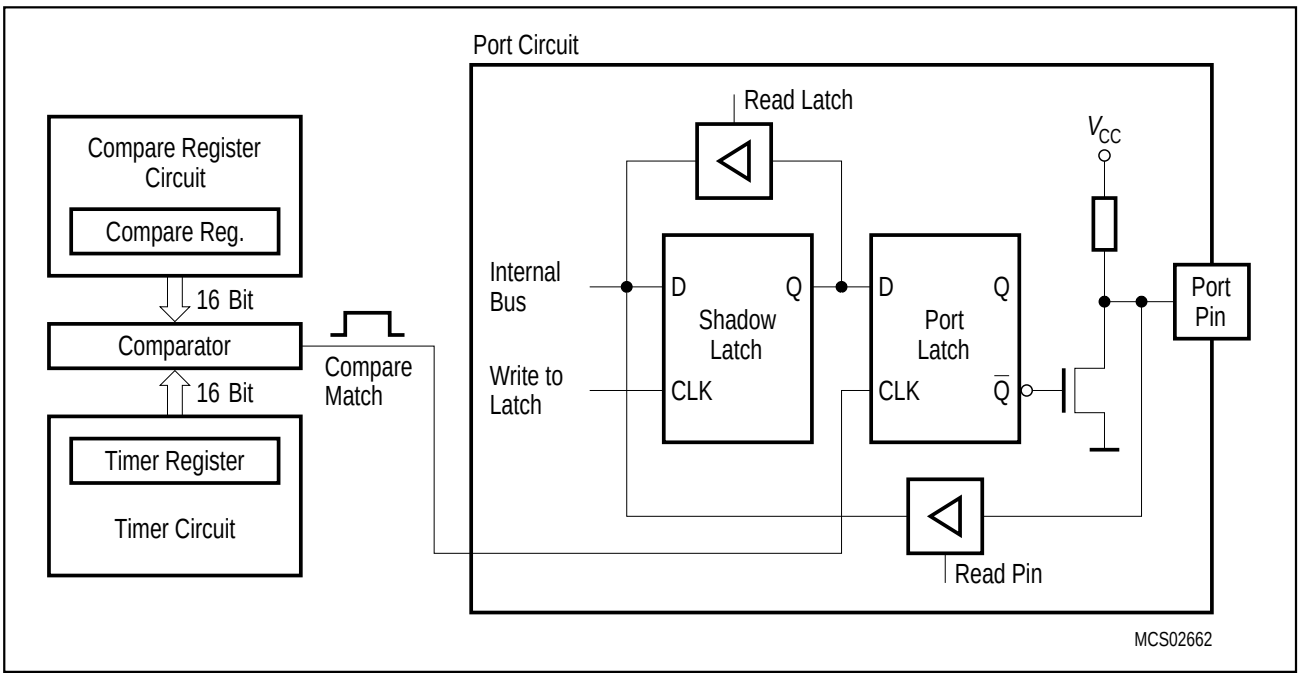


Figure 6-19
Port Latch in Compare Mode 1

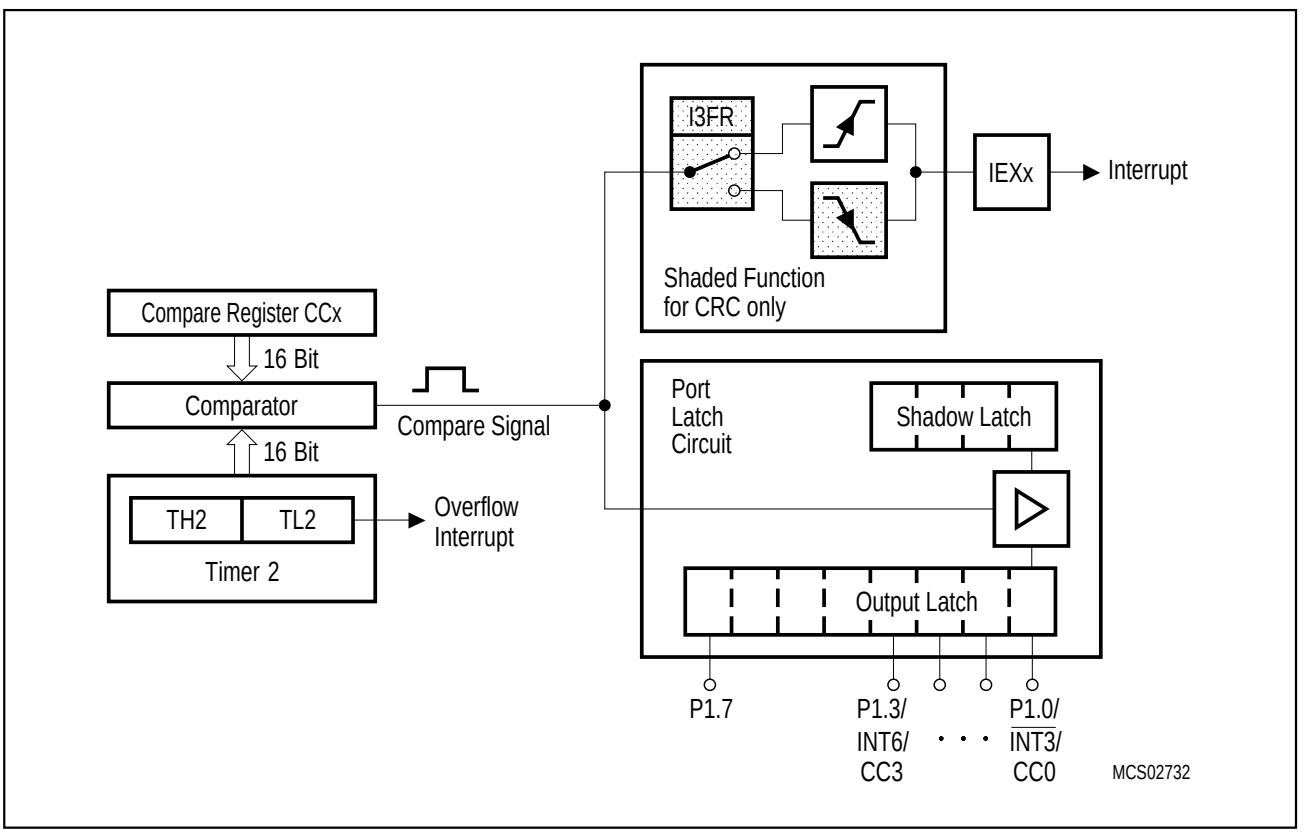


Figure 6-20
Timer 2 with Registers CCx in Compare Mode 1

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

6.2.2.4 Using Interrupts in Combination with the Compare Function

The compare service of registers CRC, CC1, CC2 and CC3 is assigned to alternate output functions at port pins P1.0 to P1.3. Another option of these pins is that they can be used as external interrupt inputs. However, when using the port lines as compare outputs 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 - of course provided 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

Firstly, 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.

Secondly, and this is the most interesting advantage of the compare feature, the output pin is exclusively controlled by hardware therefore 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 suff icient space of time.

Please note two special cases where a program using compare interrupts could show a "surprising" behavior :

The first 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 (= 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, you can select whether an interrupt should be generated when the compare signal goes active or inactive, depending on the status of bit I3FR in T2CON.

Initializing the interrupt to be negative transition triggered is advisable in the above case. Then the compare signal is already inactive and any write access to the port latch just changes the contents of the "shadow-latch".

Please note that for CC registers 1 to 3 an interrupt is always requested when the compare signal goes active.

The second configuration which should be noted is when compare function is combined with negative transition activated interrupts. lf the port latch of port P1.0 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 lf the request flag is cleared by software after the compare is activated and before the external interrupt is enabled.

6.2.2.5 Capture Function

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

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

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

In both cases the appropriate port 1 pin is used as input and the port latch must be programmed to contain a 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 is latched to the appropriate capture register in the cycle following the one in which the transition was identified.

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

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

Figure 6-21 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.

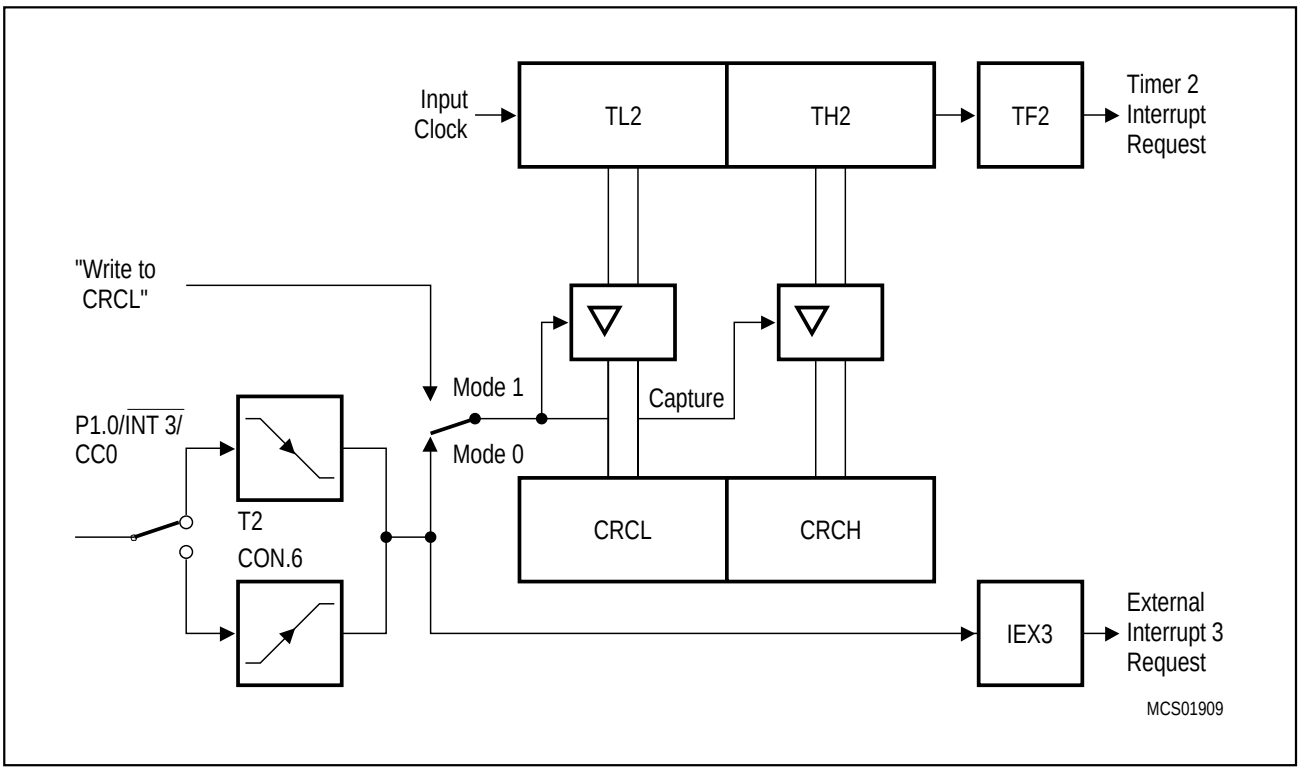


Figure 6-21
Timer 2 - Capture with Register CRC

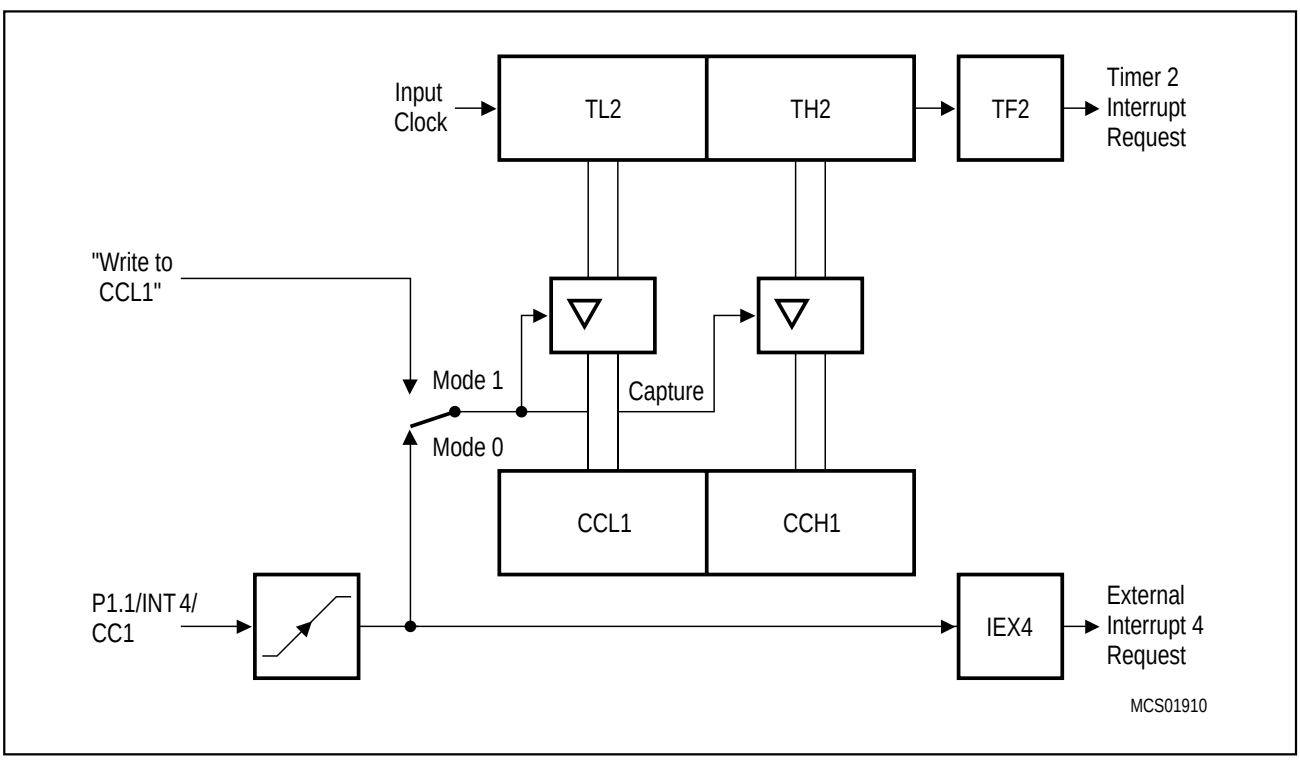


Figure 6-22
Timer 2 - Capture with Registers CC1 to CC3

6.3 Serial Interface

The serial port of the C515A is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register (however, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at special function register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes (one synchronous mode, three asynchronous modes). The baud rate clock for the serial port is derived from the oscillator frequency (mode 0, 2) or generated either by timer 1 or by a dedicated baud rate generator (mode 1, 3).

Mode 0, Shift Register (Synchronous) Mode:

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 data bits are transmitted/ received: (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency. (See **section 6.3.4** for more detailed information)

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

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

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

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

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

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

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1. The serial interfaces also provide interrupt requests when a transmission or a reception of a frame has completed. The corresponding interrupt request flags for serial interface 0 are TI or RI, resp. See **chapter 7** of this user manual for more details about the interrupt structure. The interrupt request flags TI and RI can also be used for polling the serial interface 0 if the serial interrupt is not to be used (i.e. serial interrupt 0 not enabled).

6.3.1 Multiprocessor Communication

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

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

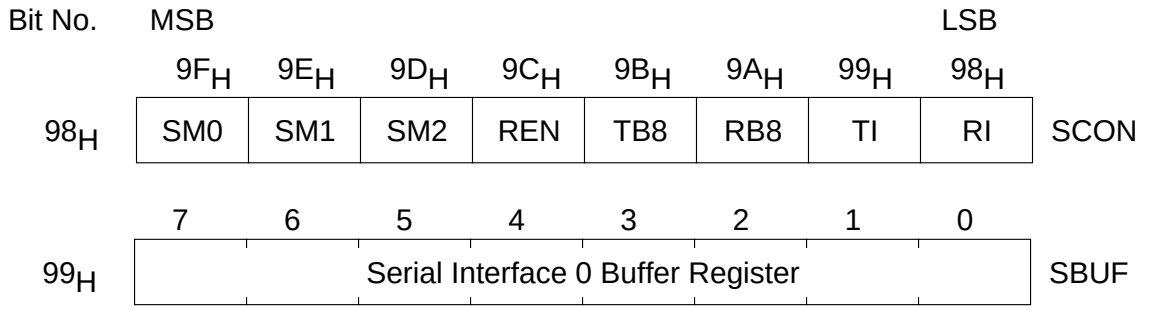
SM2 has no effect in mode 0. SM2 can be used in mode 1 to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

6.3.2 Serial Port Registers

The serial port control and status register is the 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 0. Writing to SBUF loads the transmit register and initiates transmission. Reading out SBUF accesses a physically separate receive register.

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



Bit	Function
SM0 SM1	Serial port 0 operating mode selection bits

SM0	SM1	Selected operating mode
0	0	Serial mode 0 : Shift register, fixed baud rate (f_{OSC}/12)
0	1	Serial mode 1 : 8-bit UART, variable baud rate
1	0	Serial mode 2 : 9-bit UART, fixed baud rate (f_{OSC}/32 or f_{OSC}/64)
1	1	Serial mode 3 : 9-bit UART, variable baud rate

SM2	Enable serial port multiprocessor communication in modes 2 and 3 In mode 2 or 3, if SM2 is set to 1 then RI0 will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	Serial port receiver enable Enables serial reception. Set by software to enable serial reception. Cleared by software to disable serial reception.
TB8	Serial port transmitter bit 9 TB8 is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	Serial port receiver bit 9 In modes 2 and 3, RB8 is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	Serial port transmitter interrupt flag TI is set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. TI must be cleared by software.
RI	Serial port receiver interrupt flag RI is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (exception see SM2). RI must be cleared by software.

6.3.3 Baud Rate Generation

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

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

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

Bit No.	MSB							LSB	
	DF_H	DE_H	DD_H	DC_H	DB_H	DA_H	$D9_H$	$D8_H$	
$D8_H$	BD	CLK	ADEX	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 in controlling the serial port baud rate.

Bit	Function
BD	Baud rate generator enable When set, the baud rate of the serial interface is derived from the dedicated programmable baud rate generator. When cleared (default after reset), baud rate is derived from the timer 1 overflow rate.
SMOD	Double baud rate When set, the baud rate of serial interface 0 in modes 1, 2, 3 is doubled. After reset this bit is cleared.

Figure 6-23 shows the configuration for the baud rate generation of the serial port.

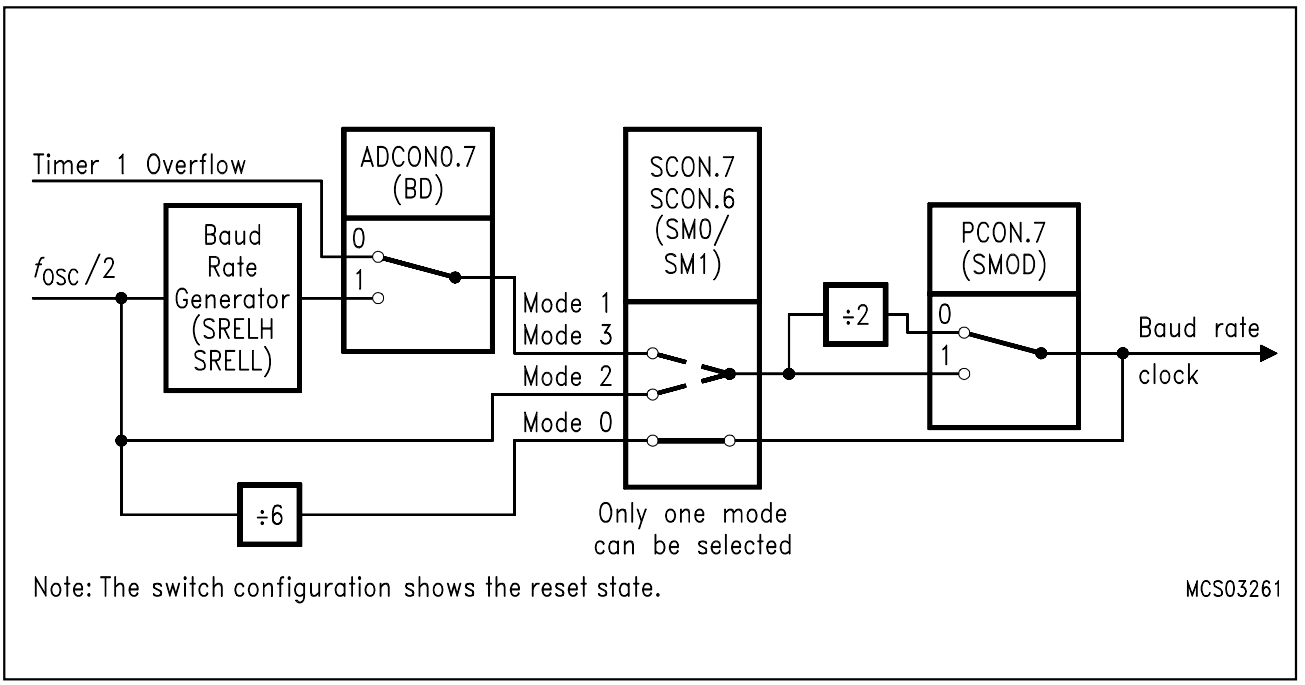


Figure 6-23
Baud Rate Generation for the Serial Port

Depending on the programmed operating mode different paths are selected for the baud rate clock generation. **Table 6-5** shows the dependencies of the serial port baud rate clock generation from the 3 control bits and from the mode which is selected in the special function register SCON.

Table 6-5
Serial Interface 0 - Baud Rate Dependencies

Serial Interface 0 Operating Modes	Active Control Bits		Baud Rate Clock Generation
	BD	SMOD	
Mode 0 (Shift Register)	–	–	Fixed baud rate clock fosc/12
Mode 1 (8-bit UART) Mode 3 (9-bit UART)	X	X	BD=0 : Timer 1 overflow is used for baud rate generation; SMOD controls a divide-by-2 option. BD=1 : Baud rate generator is used for baud rate generation; SMOD controls a divide-by-2 option .
Mode 2 (9-bit UART)	–	X	Fixed baud rate clock fosc/32 (SMOD=1) or fosc/64 (SMOD=0)

6.3.3.1 Baud Rate in Mode 0

The baud rate in mode 0 is fixed to :

$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

6.3.3.2 Baud Rate in Mode 2

The baud rate 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 baud rate is 1/64 of the oscillator frequency.
If SMOD = 1, the baud rate is 1/32 of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{\text{SMOD}}}{64} \times \text{oscillator frequency}$$

6.3.3.3 Baud Rate in Mode 1 and 3

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

6.3.3.3.1 Using the Programmable Baud Rate Generator

In modes 1 and 3, the C515A can use an internal baud rate 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 affects the input and output clock signal of the baud rate 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 baud rate generator is $f_{OSC}/2$.

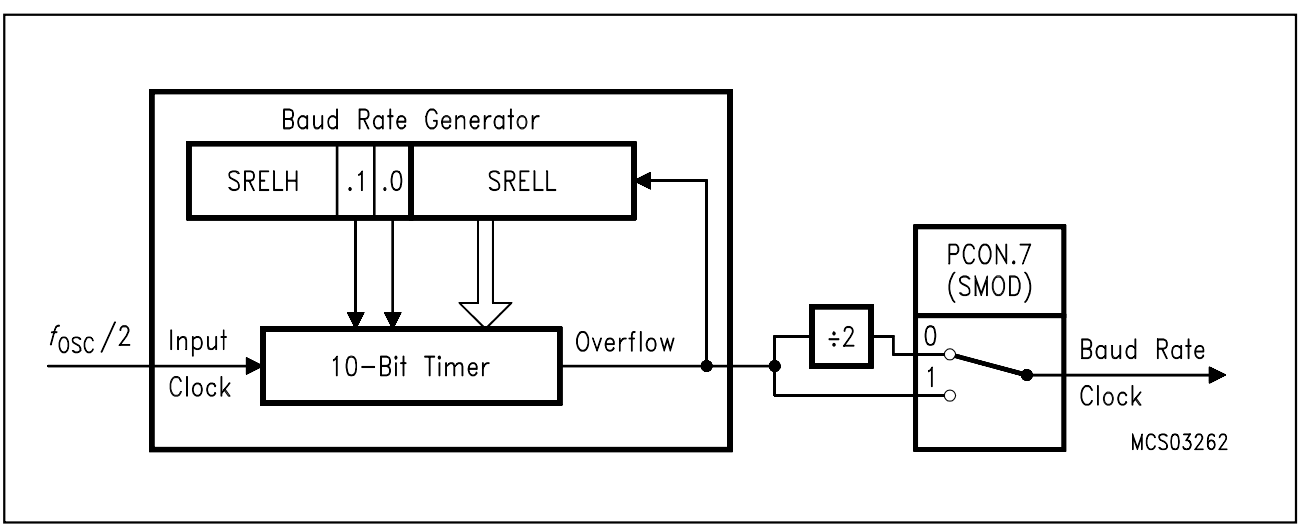
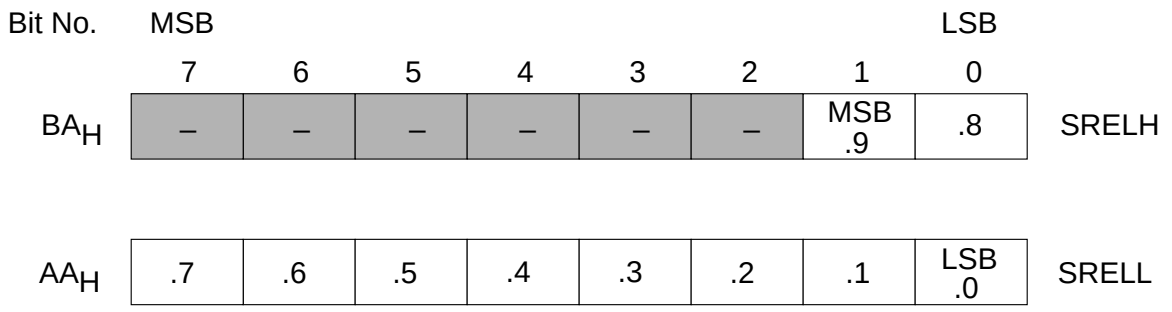


Figure 6-24
Serial Port Input Clock when using the Programmable Baud Rate Generator

The baud rate 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 baud rate 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 : $D9_H$**



The shaded bits are don't care 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.

After reset SRELH and SRELL have a reload value of $3D9_H$. With this reload value the baud rate generator has an overflow rate of input clock / 39 (compatible to C515). With 12 MHz oscillator frequency, the commonly used baud rates 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) are available (with 0.16 % deviation).

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

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \times \text{oscillator frequency}}{64 \times (\text{baud rate generator overflow rate})}$$

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

6.3.3.3.2 Using Timer 1 to Generate Baud Rates

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

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

The timer 1 interrupt is usually disabled in this application. Timer 1 itself can be configured for either "timer" or "counter" operation, 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 baud rate is given by the formula:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \times \text{oscillator frequency}}{32 \times 12 \times (256 - (TH1))}$$

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

Table 6-6 lists various commonly used baud rates and shows how these baud rates can be obtained from timer 1 or from the baud rate generator.

Table 6-6
Commonly used Baud Rates

Baud Rate		f_{osc} (MHz)	SMOD	BD	Timer 1	
					Mode	Reload Value
Mode 1, 3 :	62.5 Kbaud	12.0	1	0	2	FF_H
	125 Kbaud	24.0	1	0	2	FF_H
	19.5 Kbaud	11.059	1	0	2	FD_H
	9.6 Kbaud	11.059	0	0	2	FD_H
	4.8 Kbaud	11.059	0	0	2	FA_H
	2.4 Kbaud	11.059	0	0	2	$F4_H$
	1.2 Kbaud	11.059	0	0	2	$E8_H$
	110 Baud	6.0	0	0	2	72_H
	110 Baud	12.0	0	0	1	$FEEB_H$
					Baud Rate Generator Reload value	
	375 Kbaud	12.0	1	1	$3FF_H$	
	562.5 Kbaud	18.0	1	1	$3FF_H$	
	750 Kbaud	24.0	1	1	$3FF_H$	
	9.6 Kbaud	12.0	1	1	$3D9_H$	
	9.6 Kbaud	18.0	1	1	$3C5_H$	
	9.6 Kbaud	24.0	1	1	$3B2_H$	
Mode 0 :	1 Mbaud	12.0	-	-	-	-
	1.5 Mbaud	18.0	-	-	-	-
	2 Mbaud	24.0	-	-	-	-
Mode 2 :	187.5 Kbaud	12.0	0	-	-	-
	375 Kbaud	12.0	1	-	-	-
	281 Kbaud	18.0	0	-	-	-
	562.5 Kbaud	18.0	1	-	-	-
	375 Kbaud	24.0	0	-	-	-
	750 Kbaud	24.0	1	-	-	-

6.3.4 Details about Mode 0

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

Figure 6-25 shows a simplified functional diagram of the serial port in mode 0. The associated timing is illustrated in **figure 6-26**.

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

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

As data bits shift out to the right, zeroes come in from the left. When the MSB of the data byte is at the output position of the shift register, then the1 that was initially loaded into the 9th position, is just left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "Write-to-SBUF".

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

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted one position to the left. 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 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

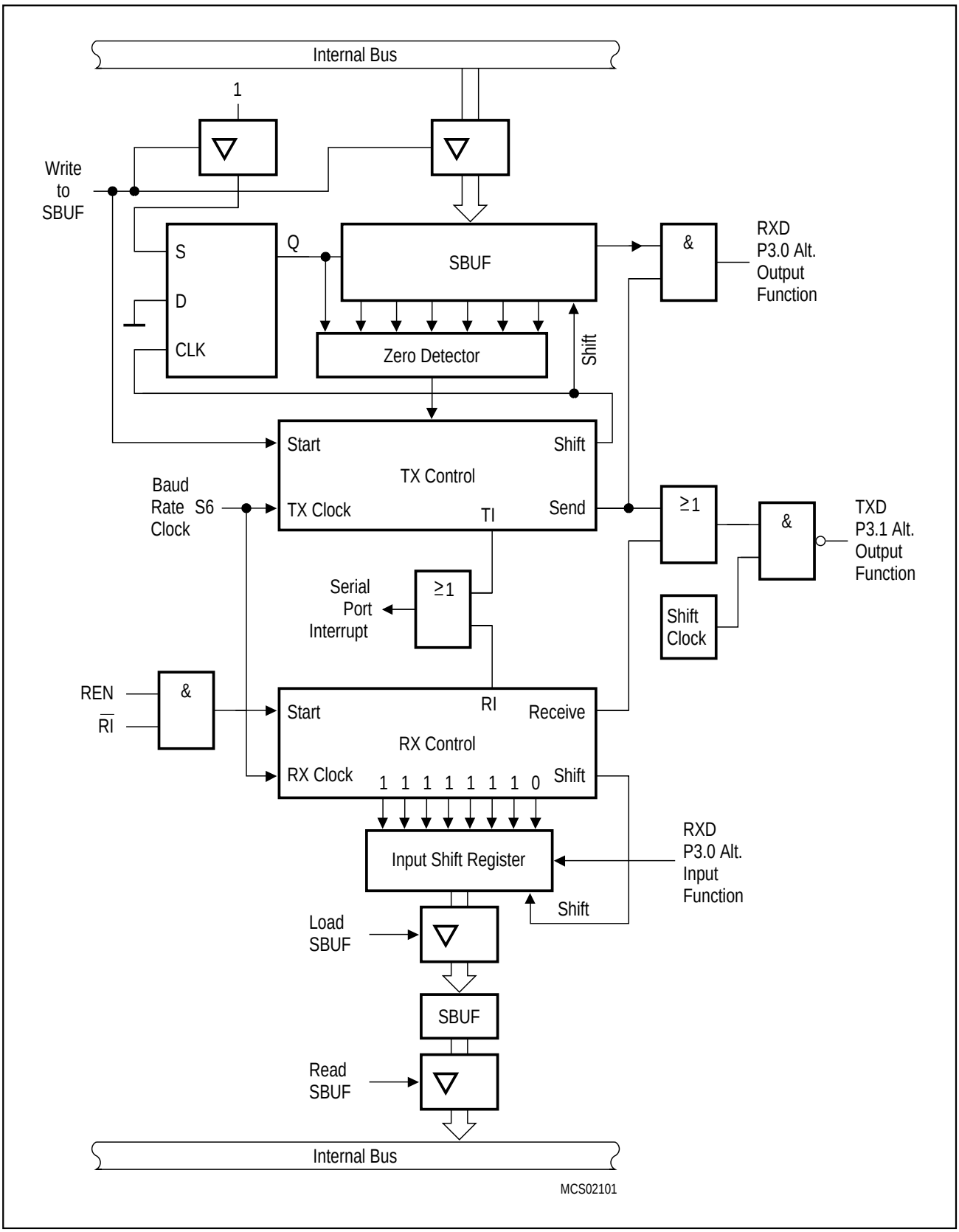


Figure 6-25
Serial Interface, Mode 0, Functional Diagram

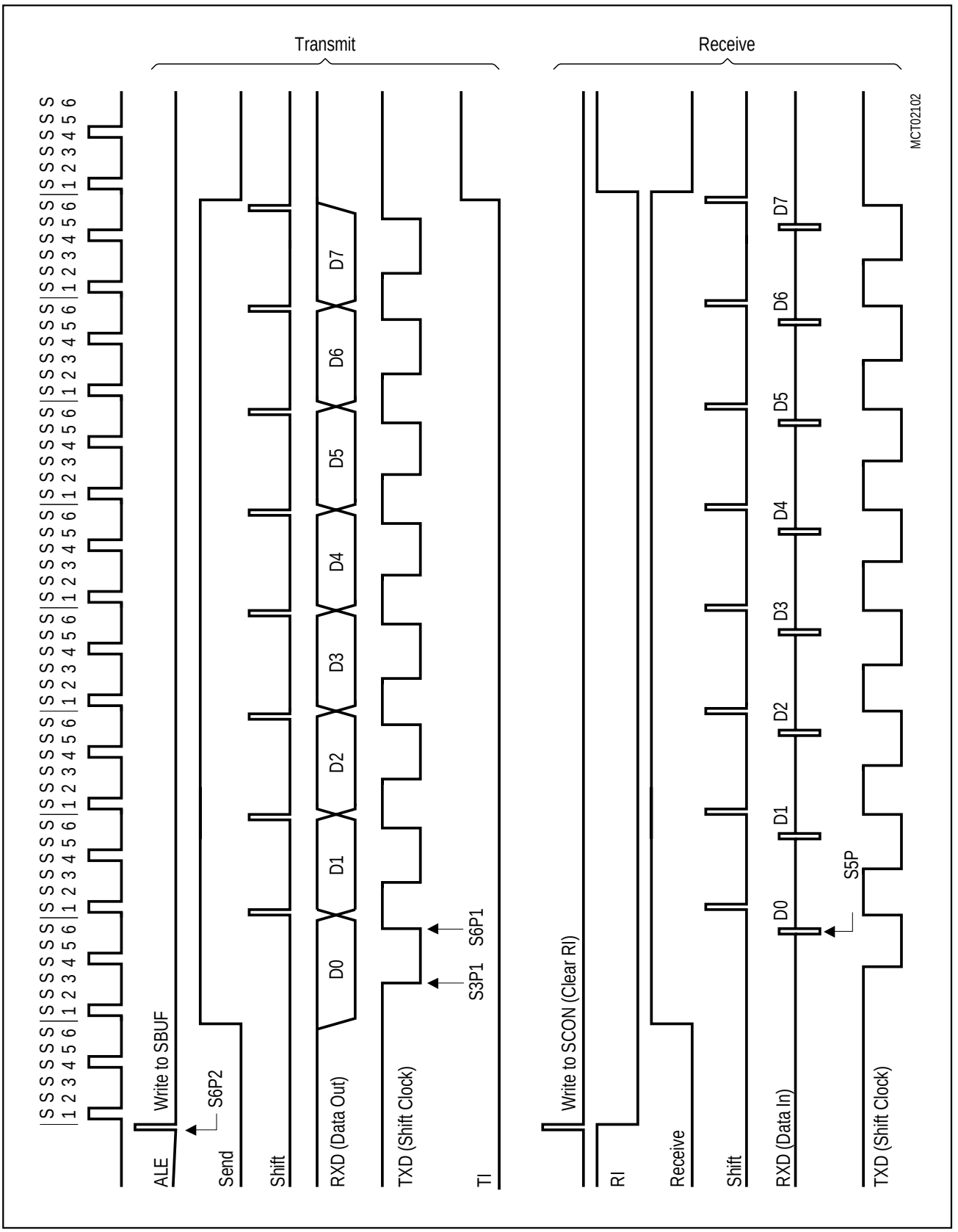


Figure 6-26
Serial Interface, Mode 0, Timing Diagram

6.3.5 Details about Mode 1

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

Figure 6-27 shows a simplified functional diagram of the serial port in mode 1. The associated timings for transmit/receive are illustrated in **figure 6-28**.

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

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

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

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

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

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

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

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

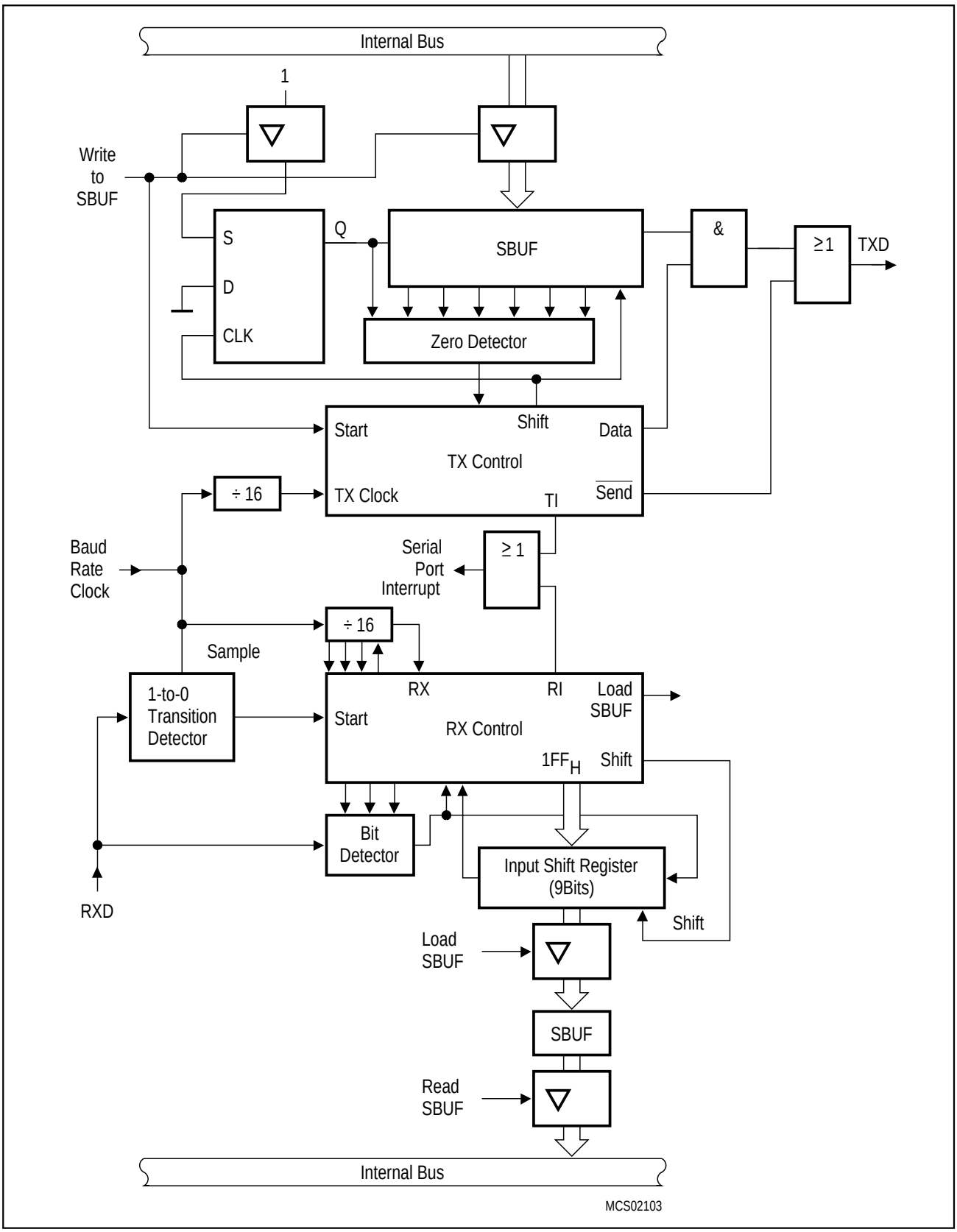


Figure 6-27
Serial Interface, Mode 1, Functional Diagram

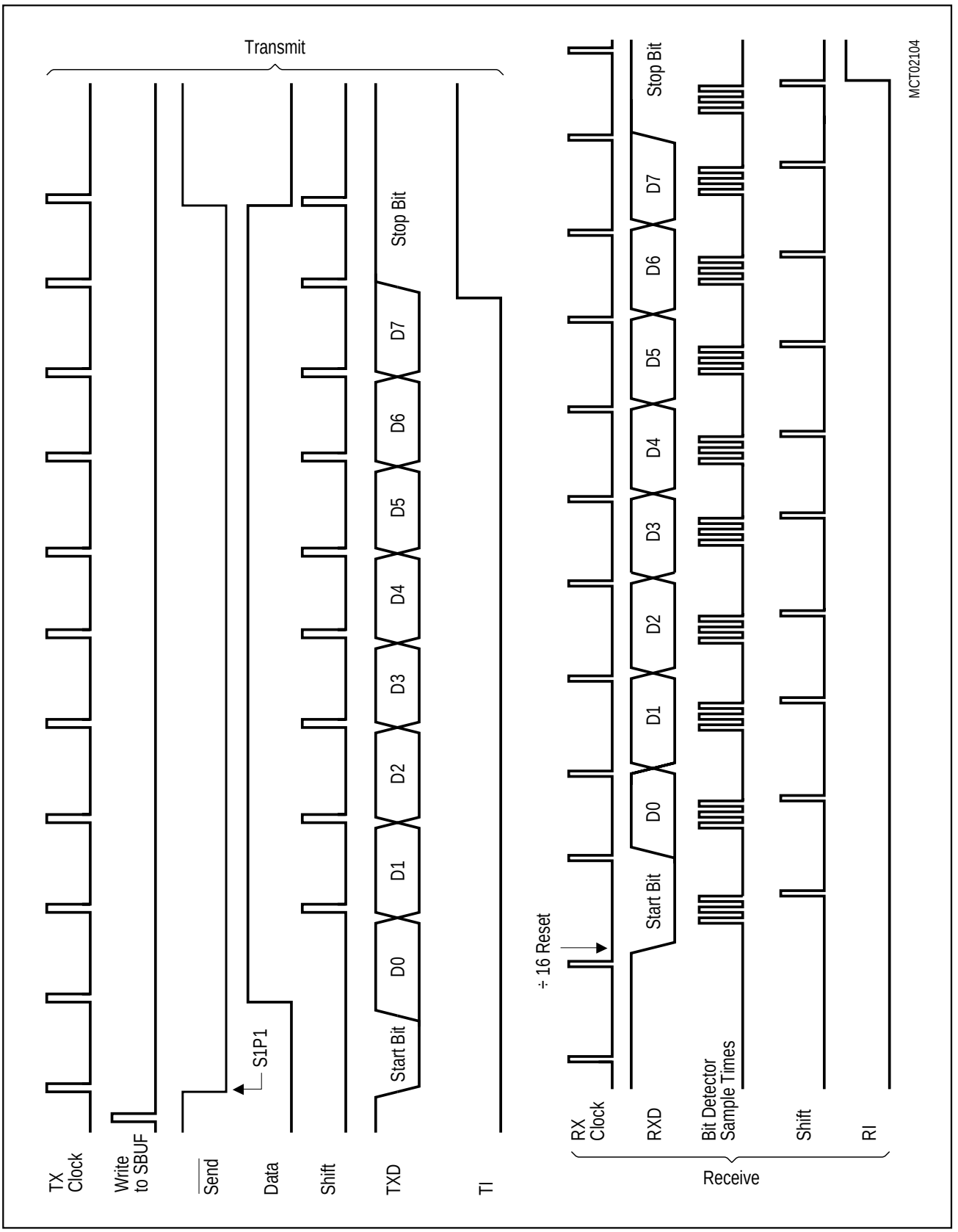


Figure 6-28
Serial Interface, Mode 1, Timing Diagram

6.3.6 Details about Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is generated by either using timer 1 or the internal baud rate generator.

Figure 6-29 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 6-30**.

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

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

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

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

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

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

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

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

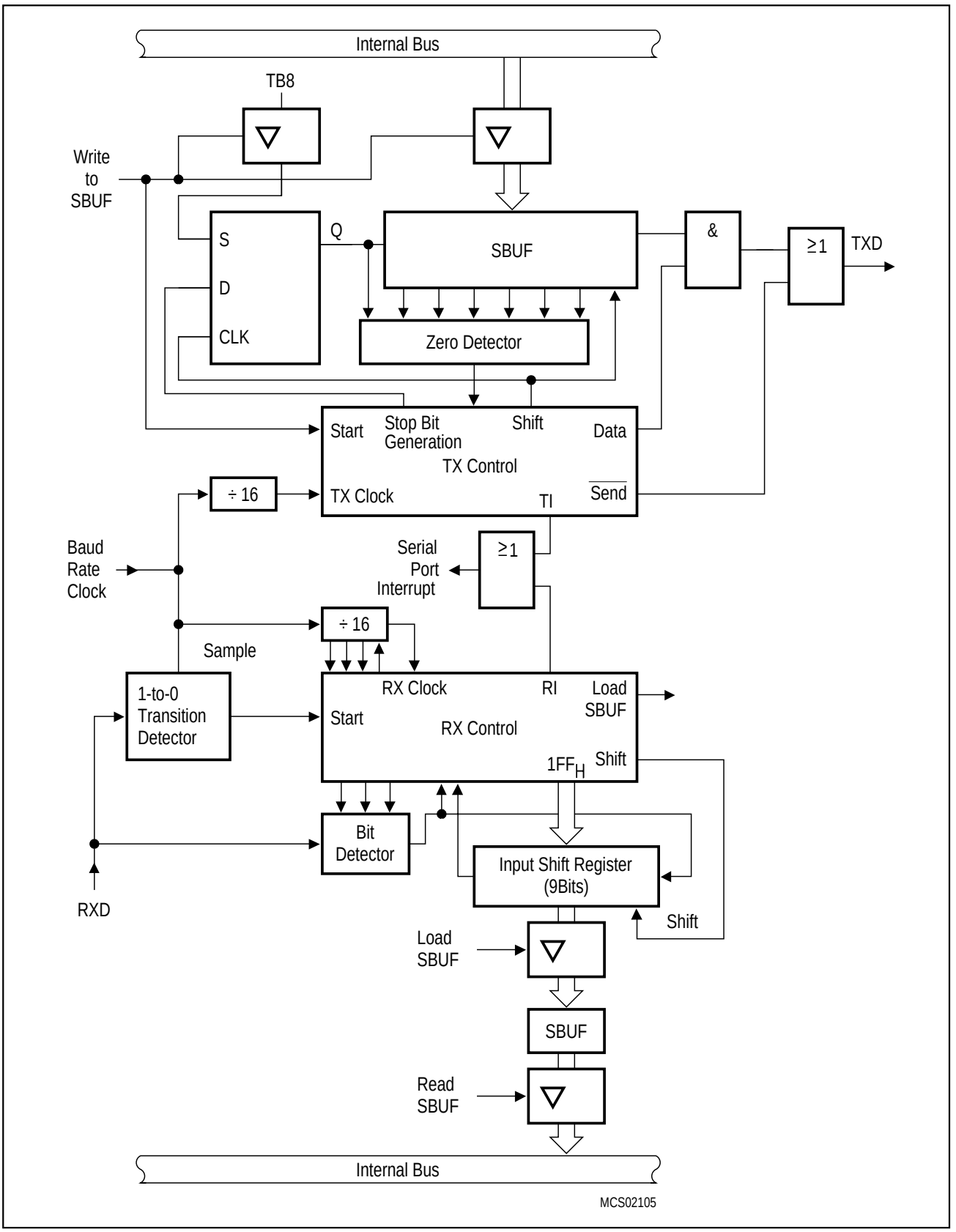


Figure 6-29
Serial Interface, Mode 2 and 3, Functional Diagram

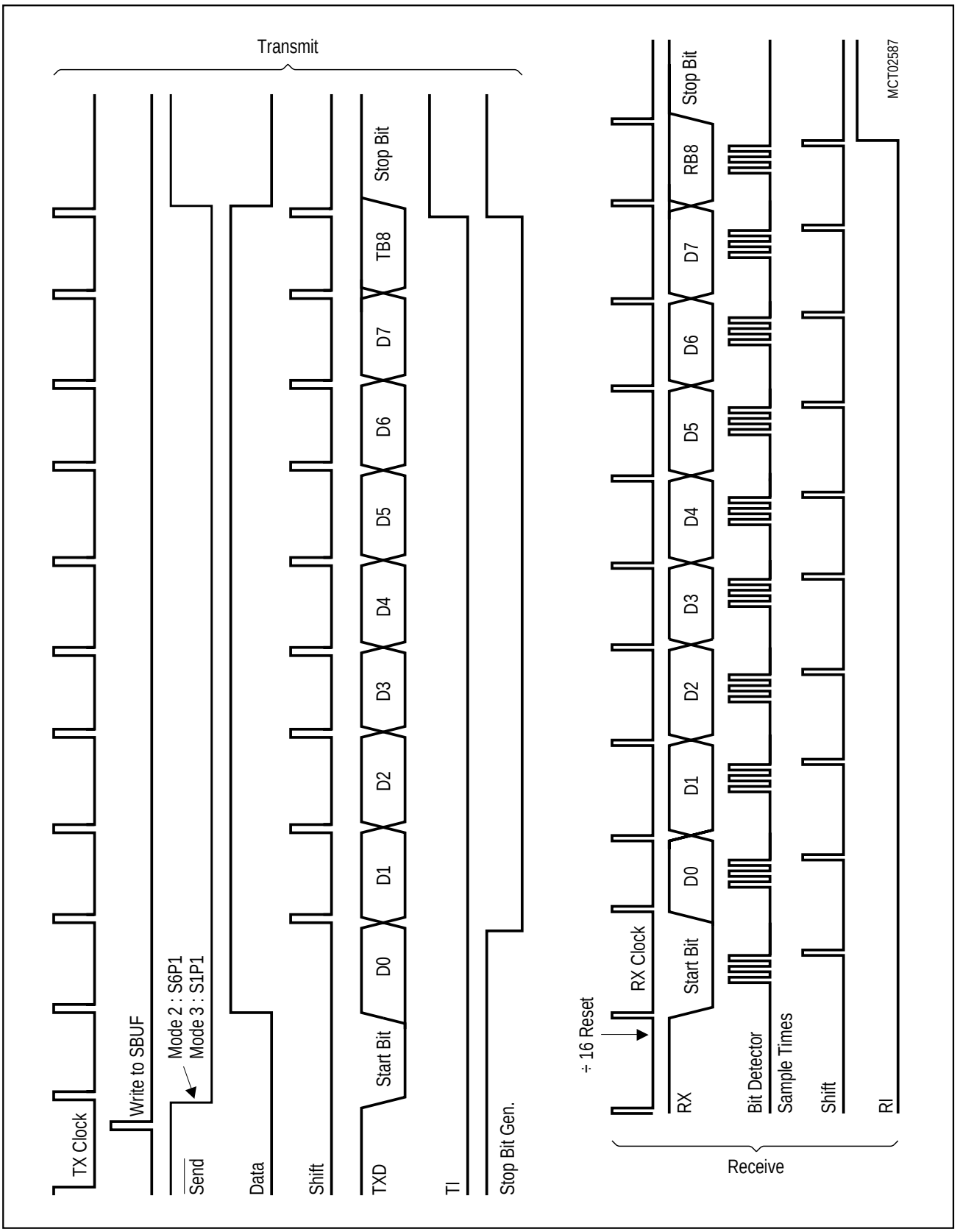


Figure 6-30
Serial Interface, Mode 2 and 3, Timing Diagram

6.4 A/D Converter

The C515A 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 multiplexed input channels (port 6), which can also be used as digital inputs
- 10-bit resolution
- Single or continuous conversion mode
- Internal or external 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 voltage range has to be held on a fixed value within the specifications. The main functional blocks of the A/D converter are shown in **figure 6-31**.

6.4.1 A/D Converter Operation

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

An externally controlled conversion can be achieved by setting the bit ADEX. In this mode one single A/D conversion is triggered by a 1-to-0 transition at pin P4.0/$\overline{\text{ADST}}$ (when ADM is 0). P4.0/$\overline{\text{ADST}}$ is sampled during S5P2 of every machine cycle. When the samples show a logic high in one cycle and a logic low in the next cycle the transition is detected and the A/D conversion is started. When ADM and ADEX is set, a continuous conversion is started when pin P4.0/$\overline{\text{ADST}}$ sees a low level. Only if no A/D conversion (single or continuous) has occurred after the last reset operation, a 1-to-0 transition is required at pin P4.0/$\overline{\text{ADST}}$ for starting the continuous conversion mode externally. The continuous A/D conversion is stopped when the pin P4.0/$\overline{\text{ADST}}$ goes back to high level. The last running A/D conversion during P4.0/$\overline{\text{ADST}}$ low level will be completed.

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 can be 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 6 is a dual purpose input port. lf the input voltage meets the specified logic levels, it can also be used as digital inputs regardless of whether the pin levels are sampled by the A/D converter at the same time.

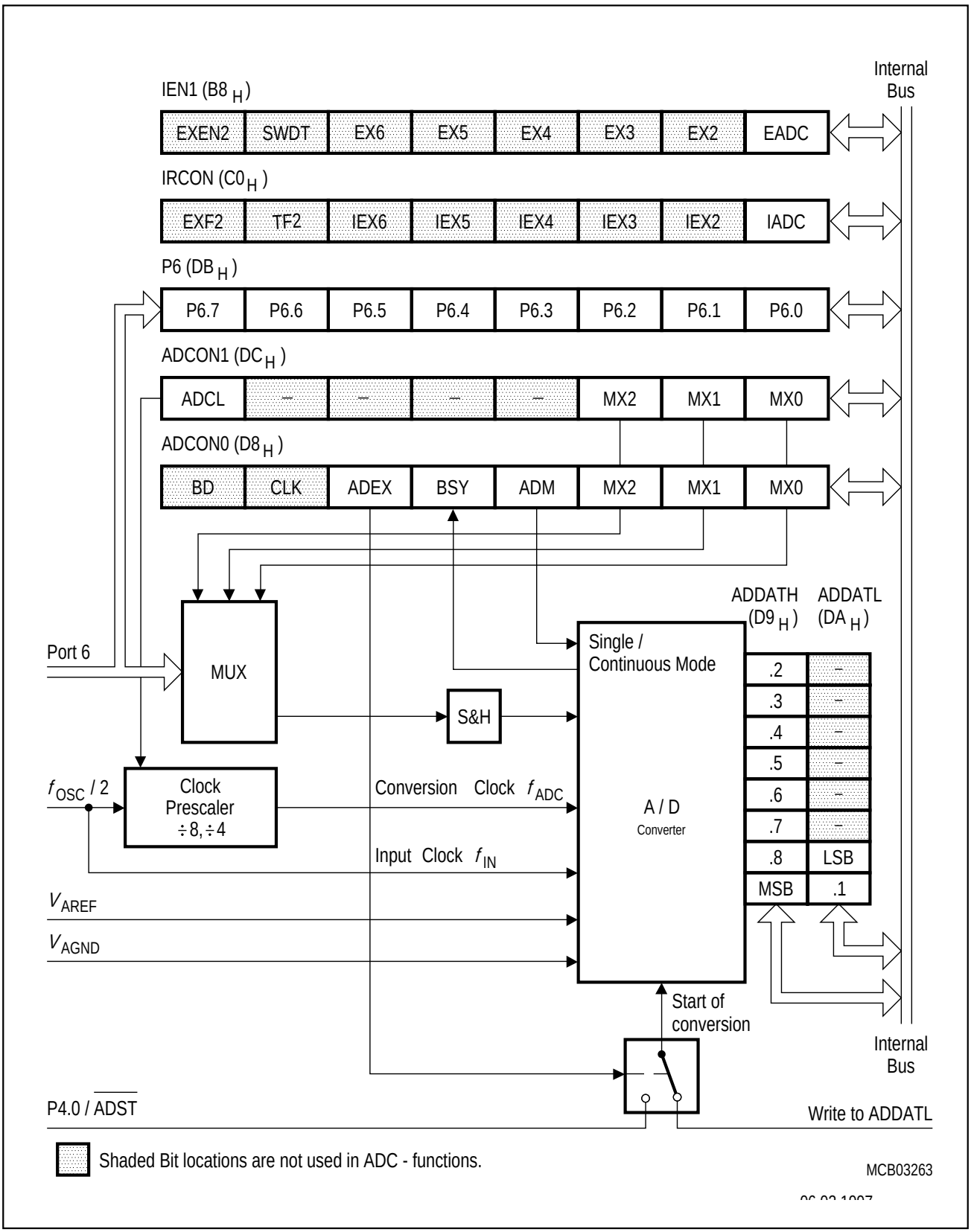


Figure 6-31
Block Diagram of the A/D Converter

6.4.2 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 : $00XXXXXX_B$**

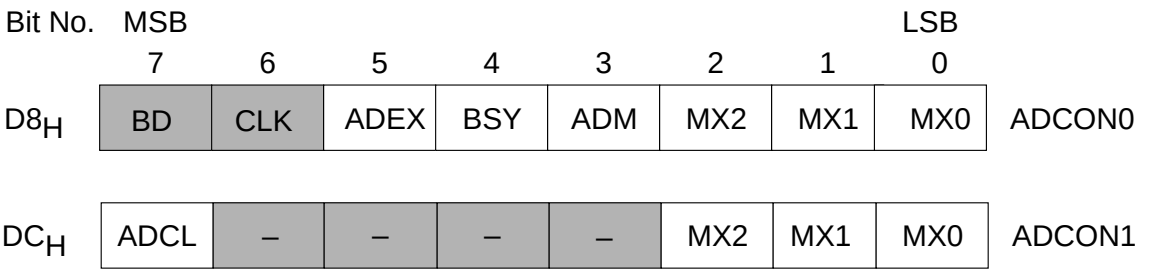
Bit No.	MSB 7	6	5	4	3	2	1	LSB 0	
$D9_H$	MSB .9	.8	.7	.6	.5	.4	.3	.2	ADDATH
DA_H	.1	LSB .0	–	–	–	–	–	–	ADDATL

The shaded bits are not used for A/D converter purposes.

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 register must be read. If an 8-bit conversion result is required, only the reading of ADDATH is necessary. The data remain in ADDATH/ADDATL until it is overwritten by the next converted data. ADDAT can be read or written under software control. lf the A/D converter of the C515A is not used, register ADDATH can be used as an additional general purpose register.

Each A/D conversion is started by writing to SFR ADDATL with dummy data. If continuous conversion is selected, ADDATL must be written only once to start continuous conversion.

Special Function Registers ADCON0 (Address $D8_H$) **Reset Value : 00_H**
Special Function Registers ADCON1 (Address DC_H) **Reset Value : $0XXXX000_B$**



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

Bit	Function
–	Reserved bits for future use
ADEX	Internal / external start of converrsion When set, the external start of an A/D conversion by a falling edge at pin P4.0 / $\overline{ADST}$ is enabled.
BSY	Busy flag This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is finished.
ADM	A/D conversion mode When set, a continuous A/D conversion is selected. If cleared, the converter stops after one A/D conversion.
MX2 - MX0	A/D converter input channel select bits Bits MX2-0 can be written or read either in ADCON0 or ADCON1. The channel selection done by writing to ADCON 1(0) overwrites the selection in ADCON 0(1) when ADCON 1(0) is written after ADCON 0(1). The analog inputs are selected according the following table :

MX2	MX1	MX0	Selected Analog Input
0	0	0	P6.0 / AIN0
0	0	1	P6.1 / AIN1
0	1	0	P6.2 / AIN2
0	1	1	P6.3 / AIN3
1	0	0	P6.2 / AIN4
1	0	1	P6.3 / AIN5
1	1	0	P6.4 / AIN6
1	1	1	P6.5 / AIN7

Bit	Function
ADCL	A/D converter clock prescaler selection ADCL selects the prescaler ratio for the A/D conversion clock f_{ADC}. Depending on the clock rate f_{OSC} of the C515A, f_{ADC} must be adjusted in a way that the resulting f_{ADC} clock is less or equal 2 MHz. The prescaler ratio is selected according the following table :

ADCL	Prescaler Ratio
0	divide by 4 (default after reset)
1	divide by 8

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

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

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

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

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

	MSB							LSB	
Bit No.	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

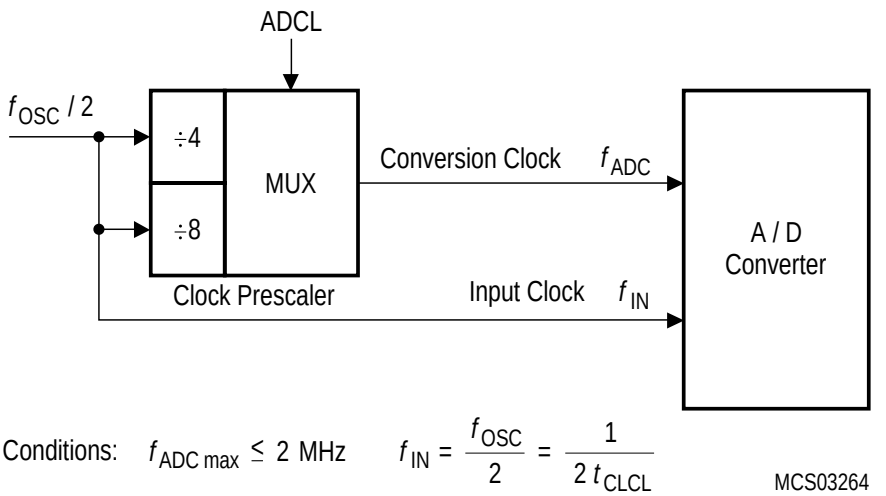
The shaded bits are not used for A/D converter control.

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.

6.4.3 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}). Both clock signals are derived from the C515A system clock f_{OSC} which is applied at the XTAL pins via the ADC clock prescaler as shown in **figure 6-32**. The input clock f_{IN} is always f_{OSC}/2 while the conversion clock must be adapted to the input clock f_{OSC}. The conversion clock is limited to a maximum frequency of 2 MHz. Therefore, the ADC clock prescaler must be programmed to a value which assures that the conversion clock does not exceed 2 MHz. The prescaler ratio is selected by the bit ADCL of SFR ADCON1.

The table in **figure 6-32** shows the prescaler ratio which must be selected for typical system clock rates. Up to 16 MHz system clock the prescaler ratio 4 is selected. Using a system clock greater than 16 MHz (max. 24 MHz) the prescaler ratio of at least 8 must be selected. The prescaler ratio 8 is recommended when the input impedance of the analog source is to high to reach the maximum accuracy.



MCU System Clock Rate (f_{OSC})	f_{IN} [MHz]	Prescaler Ratio	f_{ADC} [MHz]	ADCL
3.5 MHz	1.75	÷ 4	.438	0
12 MHz	6	÷ 4	1.5	0
16 MHz	8	÷ 4	2	0
18 MHz	9	÷ 8	1.125	1
24 MHz	12	÷ 8	1.5	1

Figure 6-32
A/D Converter Clock Selection

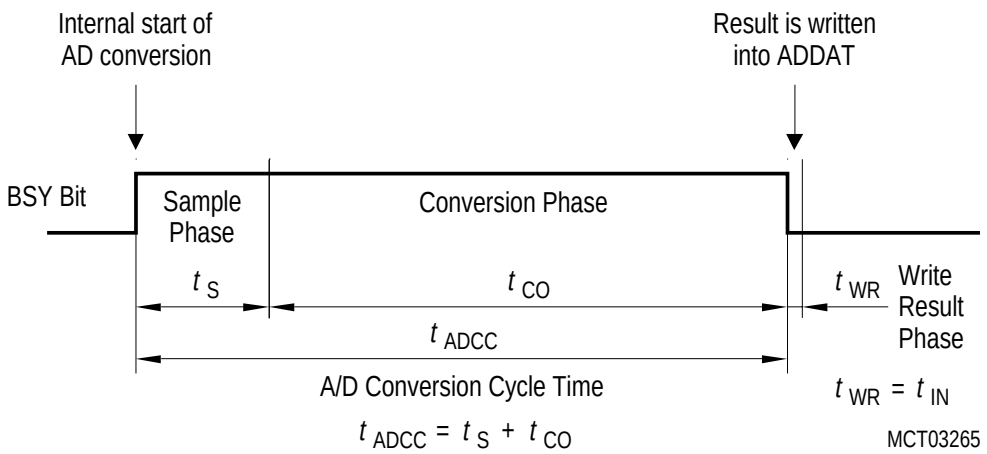
The duration of an A/D conversion is a multiple of the period of the f_{IN} clock signal. The calculation of the A/D conversion time is shown in the next section.

6.4.4 A/D Conversion Timing

An A/D conversion is internally 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 (including 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 three phase times t_S and t_{CO}. The duration of the three phases of an A/D conversion is specified by its specific timing parameter as shown in **figure 6-33**.



Selected Divider Ratio	t_S	t_{CO}	t_{ADCC}
÷ 4	8 x t_{IN}	40 x t_{IN}	48 x t_{IN}
÷ 8	16 x t_{IN}	80 x t_{IN}	96 x t_{IN}

Figure 6-33
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. With 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 also a calibration takes place. During this calibration alternating offset and linearity calibration cycles are executed (see also **section 6.4.5**). At the end of the conversion 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 6-34 shows how an A/D conversion is embedded into the microcontroller cycle scheme using the relation 12 x t_{IN} = 1 instruction cycle. It also shows the behaviour of the busy flag (BSY) and the interrupt flag (IADC) during an A/D conversion.

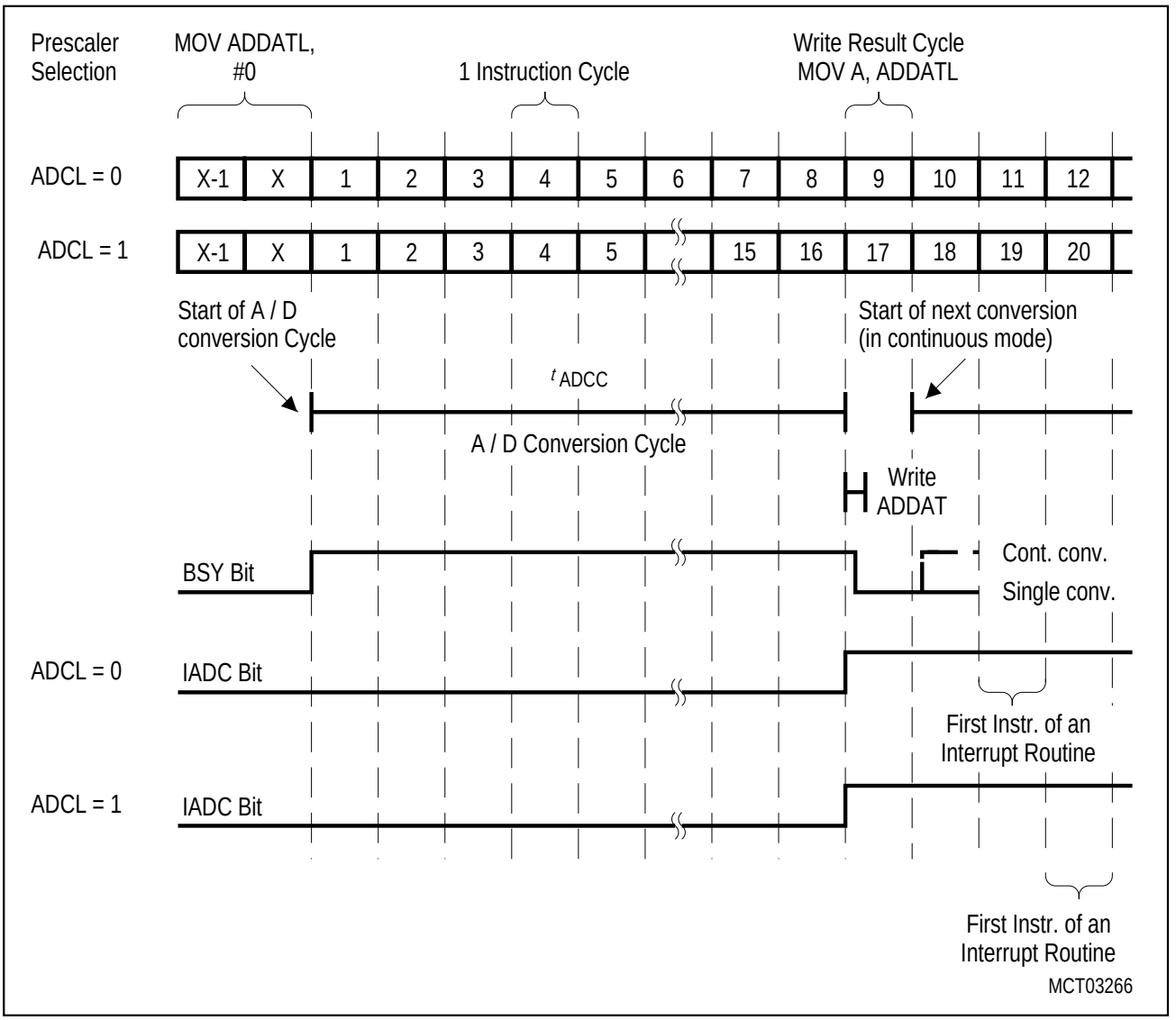


Figure 6-34
A/D Conversion Timing in Relation to Processor Cycles

Depending on the selected prescaler ratio (see **figure 6-32**), two different relationships between machine cycles and A/D conversion are possible. The A/D conversion is always started with the beginning of a processor cycle when it has been started by writing SFR ADDATL with dummy data or after an high-to-low transition has been detcted at P4.0 / $\overline{\text{ADST}}$. The ADDATL write operation may take one or two machine cycles. In **figure 6-34**, the instruction MOV ADDATL,#00 starts the A/D conversion (machine cycle X-1 and X). The total A/D conversion is finished with the end of the 8th or 16th machine cycle after the A/D conversion start. In the next machine cycle the conversion result is written into the ADDAT registers and can be read in the same cycle by an instruction (e.g.

MOV A,ADDATL). If continuous conversion is selected (bit ADM set), the next conversion is started with the beginning of the machine cycle which follows the writre result cycle.

The BSY bit is set at the beginning of the first A/D conversion machine cycle and reset at the beginning of the write result cycle. If continuous conversion is selected, BSY is again set with the beginning of the machine cycle 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 priorized 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 software methods to handle the A/D conversion in the C515A.

- 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 C515A 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 C515A and the selected divider ratio of the A/D converter prescaler the total time of an A/D conversion is calculated according **figure 6-33** and **table 6-7. Figure 6-35** 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 16 MHz.

Table 6-7
A/D Conversion Time for Dedicated System Clock Rates

f_{OSC} [MHz]	Prescaler Ratio	f_{ADC} [MHz]	Sample Time t_S [µs]		Total Conversion Time t_{ADCC} [µs]	
3.5	÷ 4	.438	4.57	= 8 x t_{IN}	27.4	= 48 x t_{IN}
8	÷ 4	1	2		12	
12	÷ 4	1.5	1.33		8	
16	÷ 4	2	1		6	
18	÷ 8	1.125	1.78	= 16 x t_{IN}	10.67	= 96 x t_{IN}
24	÷ 8	1.5	1.3		8	

Note : The prescaler ratios in **table 6-7** are mimimum values. At system clock rates (f_{OSC}) up to 16 MHz the divider ratio 4 and 8 can be used. At system clock rates greater than 16 MHz only the divider ratio 8 can be used. Using higher divider ratios than required increases the total conversion time but can be useful in applications which have voltage sources with higher input resistances for the analog inputs (increased sample phase).

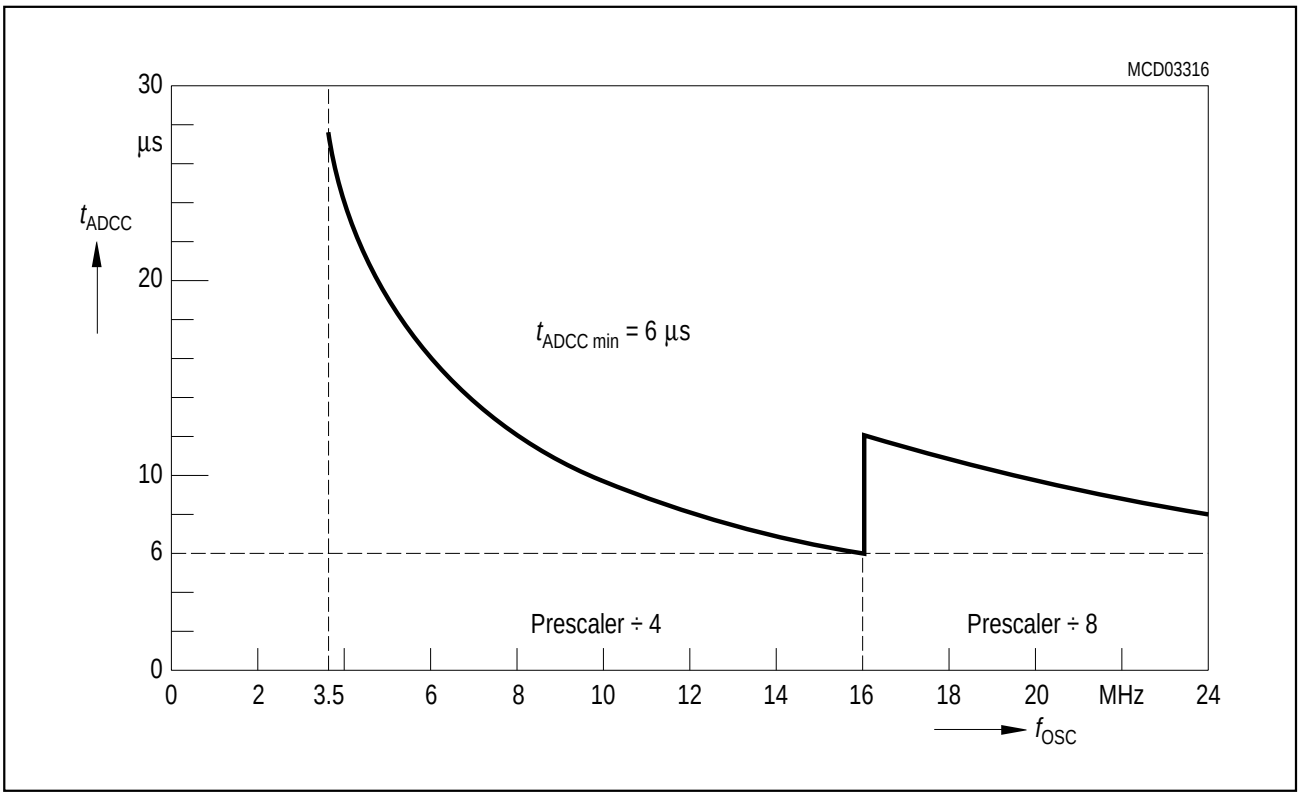


Figure 6-35
Minimum A/D Conversion Time in Relation to System Clock

6.4.5 A/D Converter Calibration

The C515A A/D converter includes hidden internal calibration mechanisms which assure a save 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 on the calibration mechanism. The calibration itself executes two basic functions :

- Offset calibration : compensation of the offset error of the internal comparator and capacitor network
- Linearity calibration : correction of the binary weighted capacitor network

The A/D converter calibration operates in two phases : calibration after a reset operation and calibration at each A/D conversion. The calibration phases are controlled by a state machine in the A/D converter. This state machine once executes a reset calibration phase after each reset operation of the C515A and stores the result values of the reset calibration phase after its end in an internal RAM. Further, these values are updated after each A/D conversion.

After a reset operation the A/D calibration is automatically started. This reset calibration phase which takes 3328 f_{ADC} clocks, alternating offset and linearity calibration is executed. Therefore, at 12 MHz oscillator frequency and with the default prescaler value of 4, a reset calibration time of approx. 2.2 ms is reached. The reset calibration phase is defined as follows (t_{OSC} = 1 / f_{OSC}) :

- Prescaler 4 selected : Reset calibration phase = 3328 x f_{ADC} = 13312 x t_{IN} = 26624 x t_{OSC}
- Prescaler 8 selected : Reset calibration phase = 3328 x f_{ADC} = 26624 x t_{IN} = 53248 x t_{OSC}

For achieving a proper reset calibration, the f_{ADC} prescaler value must satisfy the condition $f_{ADC\ max} \leq$ 2 MHz. For oscillator frequencies above 16 MHz this condition is not met with the default prescaler value (÷4) after reset. Therefore, the prescaler of the A/D converter must be adjusted by software immediately after reset by setting bit ADCL in SFR ADCON1. When setting bit ADCL directly after reset as required for oscillator clocks greater or equal 16 MHz, the clock prescaler ratio ÷8 is selected and therefore the absolute value for the reset calibration phase will be extended by factor 2.

After a reset operation of the C515A, this means when a reset calibration phase is started, the total unadjusted error TUE of the A/D converter is ± 6 LSB. After the reset calibration phase the A/D converter is calibrated according to its DC characteristics (TUE = ± 2 LSB). Nevertheless, during the reset calibration phase single or continuous A/D 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) has to be valid for an A/D conversion, it is recommended to start the first A/D conversions after reset when the reset calibration phase is finished.

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, stored in the calibration RAM, are always checked when an A/D conversion is executed and corrected if required.

7 Interrupt System

The C515A provides 12 interrupt sources with four priority levels. Five interrupts can be generated by the on-chip peripherals (timer 0, timer 1, timer 2, A/D converter, and serial interface) and seven interrupts may be triggered externally (P3.2/$\overline{\text{INT0}}$, P3.3/$\overline{\text{INT1}}$, P1.4/$\overline{\text{INT2}}$, P1.0/$\overline{\text{INT3}}$, P1.1/INT4, P1.2/INT5, P1.3/INT6). The wake-up from software power down mode interrupt has a special functionality which allows to exit from the software power-down mode by a short low pulse at pin P3.2/$\overline{\text{INT0}}$.

This chapter shows the interrupt structure, the interrupt vectors and the interrupt related special function registers. **Figure 7-1** and **7-2** give a general overview of the interrupt sources and illustrate the request and the control flags which are described in the next sections.

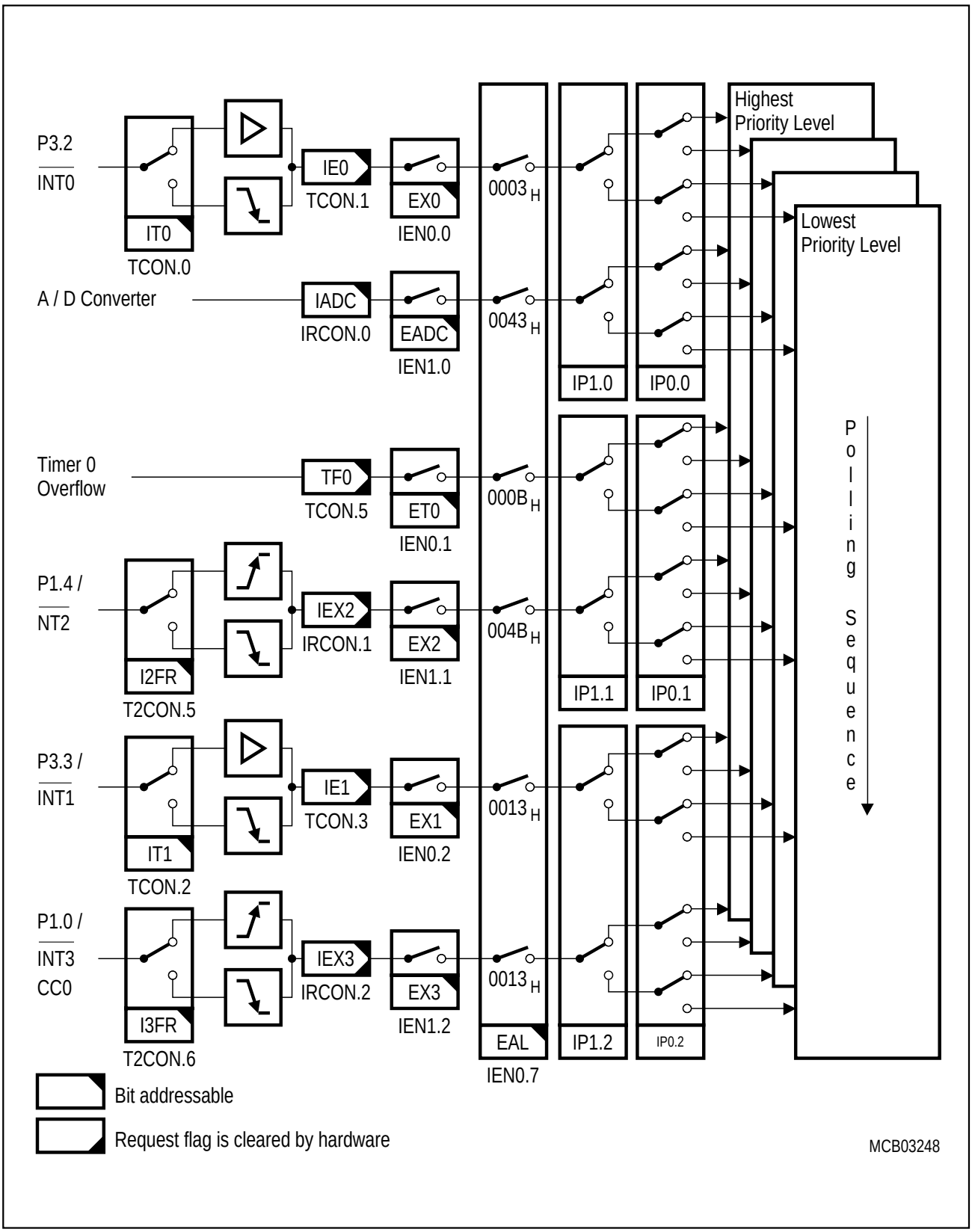


Figure 7-1
Interrupt Structure, Overview Part 1

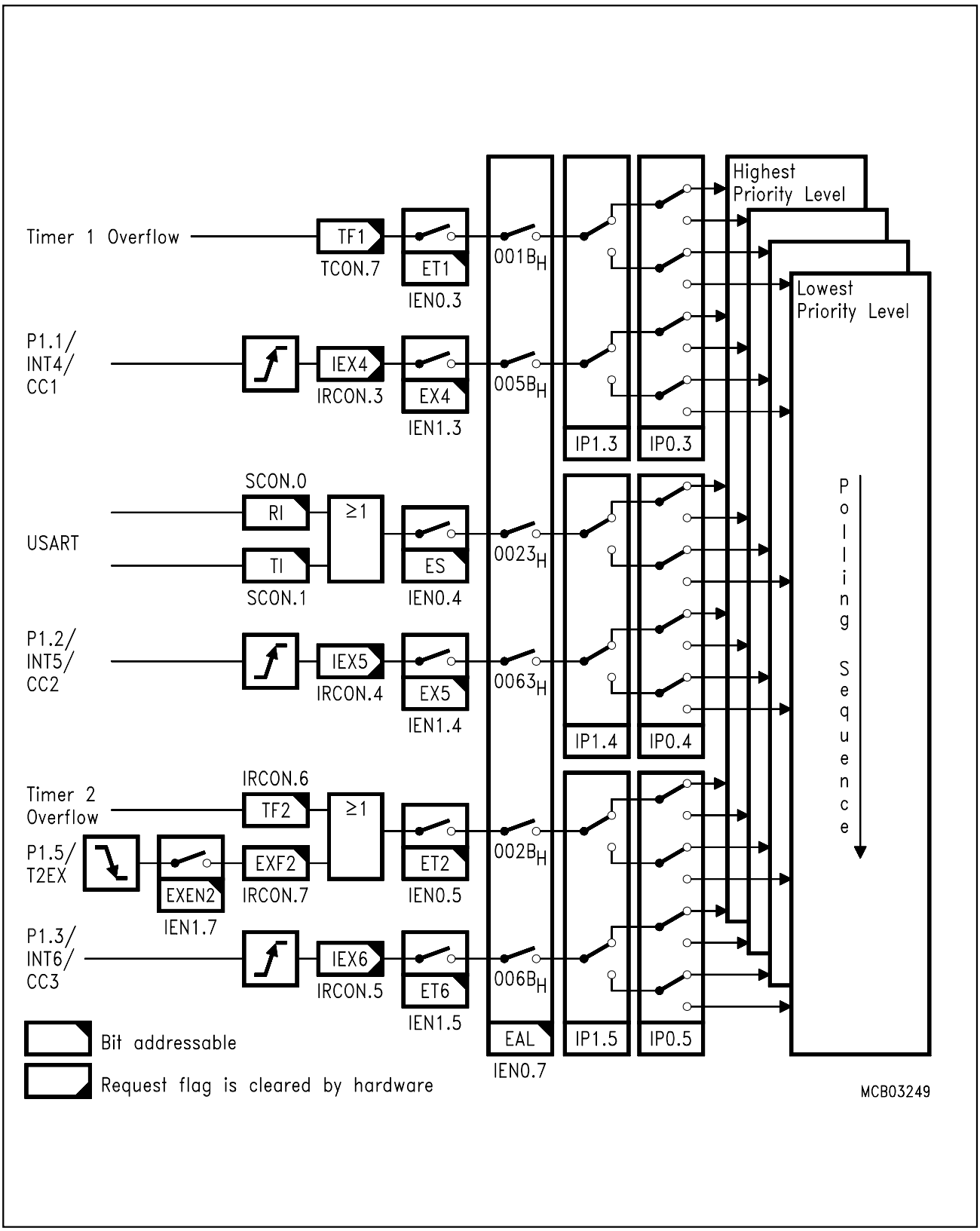


Figure 7-2
Interrupt Structure, Overview Part 2

7.1 Interrupt Registers

7.1.1 Interrupt Enable Registers

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

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

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

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

The shaded bit is not used for interrupt control.

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

The IEN1 register contains enable/disable flags of the timer 2 external timer reload interrupt, the external interrupts 2 to 6, and the A/D converter interrupt.

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

	MSB							LSB	
Bit No.	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

The shaded bit is not used for interrupt control.

Bit	Function
EXEN2	Timer 2 external reload interrupt enable If EXEN2 = 0, the timer 2 external reload interrupt is disabled. If EXEN2 = 1, the timer 2 external reload interrupt is enabled. The external reload function is not affected by EXEN2.
EX6	External interrupt 6 / capture/compare interrupt 3 enable If EX6 = 0, external interrupt 6 is disabled. If EX6 = 1, external interrupt 6 is enabled.
EX5	External interrupt 5 / capture/compare interrupt 2 enable If EX5 = 0, external interrupt 5 is disabled. If EX5 = 1, external interrupt 5 is enabled.
EX4	External interrupt 4 / capture/compare interrupt 1 enable If EX4 = 0, external interrupt 4 is disabled. If EX4 = 1, external interrupt 4 is enabled.
EX3	External interrupt 3 / capture/compare interrupt 0 enable If EX3 = 0, external interrupt 3 is disabled. If EX3 = 1, external interrupt 3 is enabled.
EX2	External interrupt 2 / capture/compare interrupt 4 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.

7.1.2 Interrupt Request / Control Flags

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.

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

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

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

	MSB							LSB	
Bit No.	CF_H	CE_H	CD_H	CC_H	CB_H	CA_H	$C9_H$	$C8_H$	
$C8_H$	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON

The shaded bits are not used for interrupt control.

Bit	Function
I3FR	External interrupt 3 rising/falling edge control flag If I3FR = 0, the external interrupt 3 is activated by a falling edge at P1.0/$\overline{INT3}$/CC0. If I3FR = 1, the external interrupt 3 is activated by a rising edge at P1.0/$\overline{INT3}$/CC0.
I2FR	External interrupt 2 rising/falling edge control flag If I2FR = 0, the external interrupt 2 is activated by a falling edge at P1.4/$\overline{INT2}$. If I2FR = 1, the external interrupt 2 is activated by a rising edge at P1.4/$\overline{INT2}$.

The **external interrupt 2** ($\overline{INT2}$) 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. lf an interrupt 2 is generated, flag IEX2 is cleared by hardware when the service routine is vectored to.

The **external interrupt 3** (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 P1.0/$\overline{INT3}$/CC0, 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.

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

	MSB							LSB	
Bit No.	$C7_H$	$C6_H$	$C5_H$	$C4_H$	$C3_H$	$C2_H$	$C1_H$	$C0_H$	
$C0_H$	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON

Bit	Function
EXF2	Timer 2 external reload flag EXF2 is set when a reload is caused by a falling edge on pin T2EX while EXEN2 = 1. If ET2 in IEN0 is set (timer 2 interrupt enabled), EXF2 = 1 will cause an interrupt. EXF2 can be used as an additional external interrupt when the reload function is not used. EXF2 must be cleared by software.
TF2	Timer 2 overflow flag Set by a timer 2 overflow and must be cleared by software. If the timer 2 interrupt is enabled, TF2 = 1 will cause an interrupt.
EX6	External interrupt 6 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.3/INT6/CC3. Cleared by hardware when processor vectors to interrupt routine.
IEX5	External interrupt 5 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.2/INT5/CC2. Cleared by hardware when processor vectors to interrupt routine.
IEX4	External interrupt 4 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.1/INT4/CC1. Cleared by hardware when processor vectors to interrupt routine.
IEX3	External interrupt 3 edge flag Set by hardware when external interrupt edge was detected or when a compare event occurred at pin P1.0/$\overline{\text{INT3}}$/CC0. Cleared by hardware when processor vectors to interrupt routine.
IEX2	External interrupt 2 edge flag Set by hardware when external interrupt edge was detected at pin P1.4/$\overline{\text{INT2}}$. Cleared by hardware when processor vectors to interrupt routine.
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 the logical OR of bit TF2 in register T2CON and bit EXF2 in register IRCON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared by software.

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

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

All of these interrupt request bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The only exceptions are the request flags IE0 and lE1. lf the external interrupts 0 and 1 are programmed to be level-activated, IE0 and lE1 are controlled by the external source via pin $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, respectively. Thus, writing a one to these bits will not set the request flag IE0 and/or lE1. In this mode, interrupts 0 and 1 can only be generated by software and by writing a 0 to the corresponding pins $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3), provided that this will not affect any peripheral circuit connected to the pins.

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

7.1.3 Interrupt Priority Registers

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

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

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

The shaded bits are not used for interrupt control.

Bit	Function
IP1.x IP0.x	Interrupt group priority level bits (x=1-6, see **table 7-1**)

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

7.2 Interrupt Priority Level Structure

The following table shows the interrupt grouping of the C515A interrupt sources.

Table 7-1
Interrupt Source Structure

Interrupt Group	Associated Interrupts High Priority → Low Priority		Priority
1	External interrupt 0	A/D converter interrupt	High
2	Timer 0 overflow	External interrupt 2	↓
3	External interrupt 1	External interrupt 3	
4	Timer 1 overflow	External interrupt 4	
5	Serial channel interrupt	External interrupt 5	
6	Timer 2 interrupt	External interrupt 6	Low

Each pair of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the 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 leveis are received simultaneously, the request of the highest priority is serviced first. lf 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, as follows.

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

7.3 How Interrupts are Handled

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

1. An interrupt of equal or higher priority is already in progress.
2. The current (polling) cycle is not in the final cycle of the instruction in progress.
3. The instruction in progress is RETI or any write access to registers IEN0 and 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 7-3**.

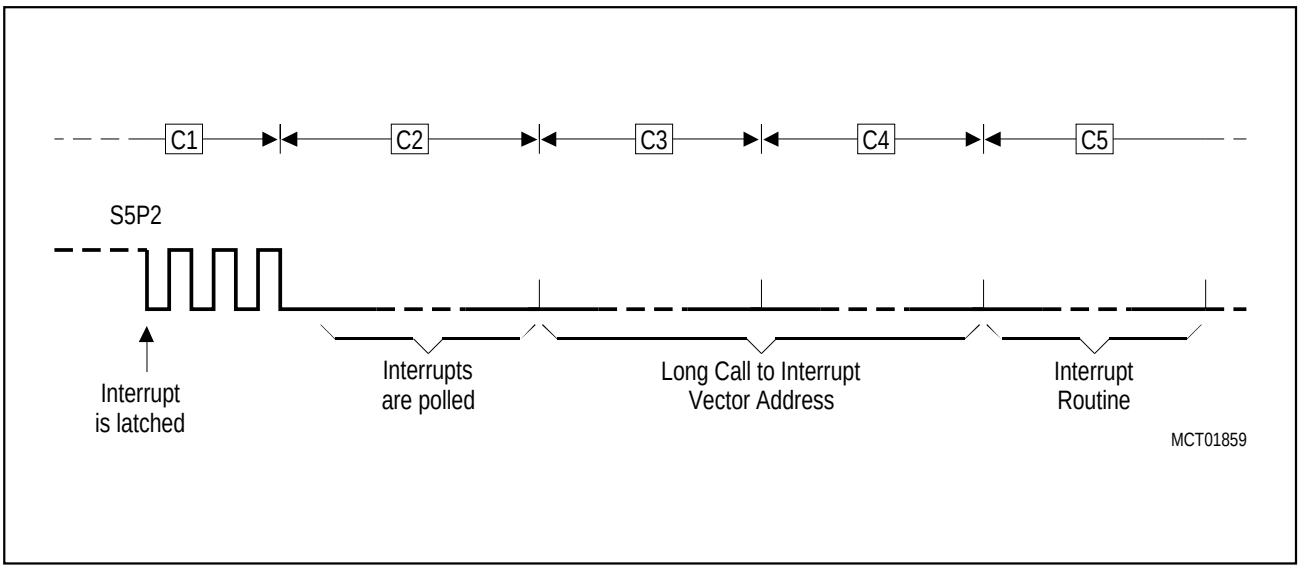


Figure 7-3
Interrupt Response Timing Diagram

Note that if an interrupt of a higher priority level goes active prior to S5P2 in the machine cycle labeled C3 in **figure 7-3** then, in accordance with the above rules, it will be vectored to during C5 and C6 without any instruction for the lower priority routine to be executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not; then this has to be done by the user's software. The hardware clears the external interrupt flags IE0 and IE1 only if they were transition-activated. The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with an address that depends on the source of the interrupt being vectored to, as shown in the following **table 7-2**.

Table 7-2
Interrupt Source and Vectors

Interrupt Source	Interrupt Vector Address	Interrupt Request Flags
External Interrupt 0	0003_H	IE0
Timer 0 Overflow	$000B_H$	TF0
External Interrupt 1	0013_H	IE1
Timer 1 Overflow	$001B_H$	TF1
Serial Channel	0023_H	RI / TI
Timer 2 Overflow / Ext. Reload	$002B_H$	TF2 / EXF2
A/D Converter	0043_H	IADC
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
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.

7.4 External Interrupts

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

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

The external interrupts 2 and 3 can be programmed to be negative or positive transition-activated by setting or clearing bit I2FR or I3FR in register T2CON. lf IxFR = 0 (x = 2 or 3), the external interrupt x is negative transition-activated. lf 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. The external timer 2 reload trigger interrupt request flag EXF2 will be activated by a negative transition at pin Pl.5/T2EX but only if bit EXEN2 is set.

Since the external interrupt pins ($\overline{INT2}$ to INT6) are sampled once in each machine cycle, an input high or low should be held for at least 6 oscillator periods to ensure sampling. lf the external interrupt is transition-activated, the external source has to hold the request pin low (high for $\overline{INT2}$ and $\overline{INT3}$, if it is programmed to be negative transition-active) for at least one cycle, and then hold it high (low) for at least one cycle to ensure that the transition is recognized so that the corresponding interrupt request flag will be set (see **figure 7-4**). The external interrupt request flags will automatically be cleared by the CPU when the service routine is called.

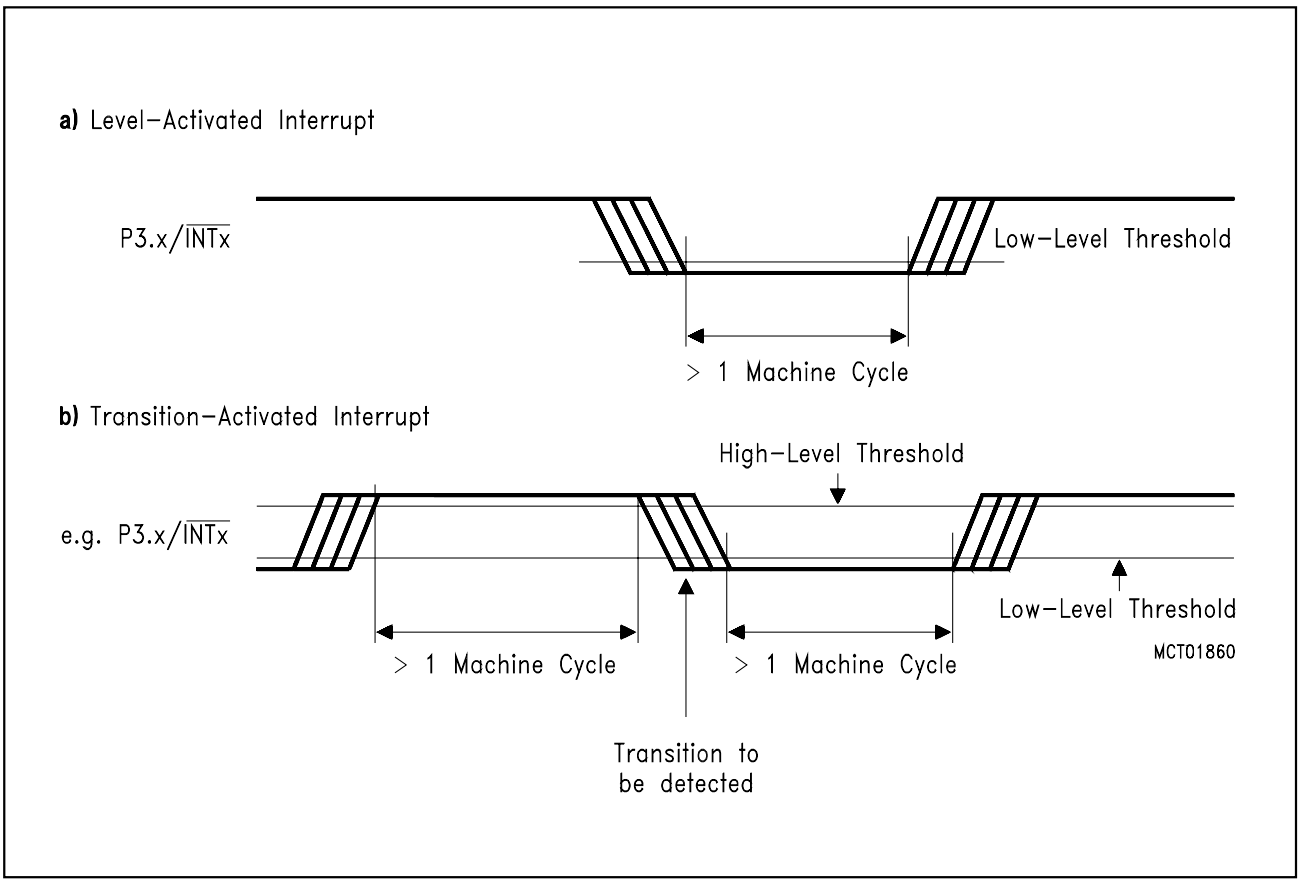


Figure 7-4
External Interrupt Detection

7.5 Interrupt Response Time

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

A longer response time would be obtained if the request was blocked by one of the three previously listed conditions. If an interrupt of equal or higer priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles since the longest instructions (MUL and DIV) are only 4 cycles long; and, if the instruction in progress is RETI or a write access to registers IEN0, IEN1 or IP0, IP1 the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction, if the instruction is MUL or DIV).

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

8 Fail Safe Mechanisms

The C515A offers enhanced fail safe mechanisms, which allow an automatic recovery from software upset or hardware failure :

- a programmable watchdog timer (WDT), with variable time-out period from 512 µs up to approx. 1.1 s at 12 MHz.
- an oscillator watchdog (OWD) which monitors the on-chip oscillator and forces the microcontroller into reset state in case the on-chip oscillator fails; it also provides the clock for a fast internal reset after power-on.

8.1 Programmable Watchdog Timer

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

The watchdog timer in the C515A is a 15-bit timer, which is incremented by a count rate of $f_{OSC}/24$ up to $f_{OSC}/384$. The system clock of the C515A is divided by two prescalers, a divide-by-two and a divide-by-16 prescaler. For programming of the watchdog timer overflow rate, the upper 7 bit of the watchdog timer can be written. **Figure 8-1** shows the block diagram of the watchdog timer unit.

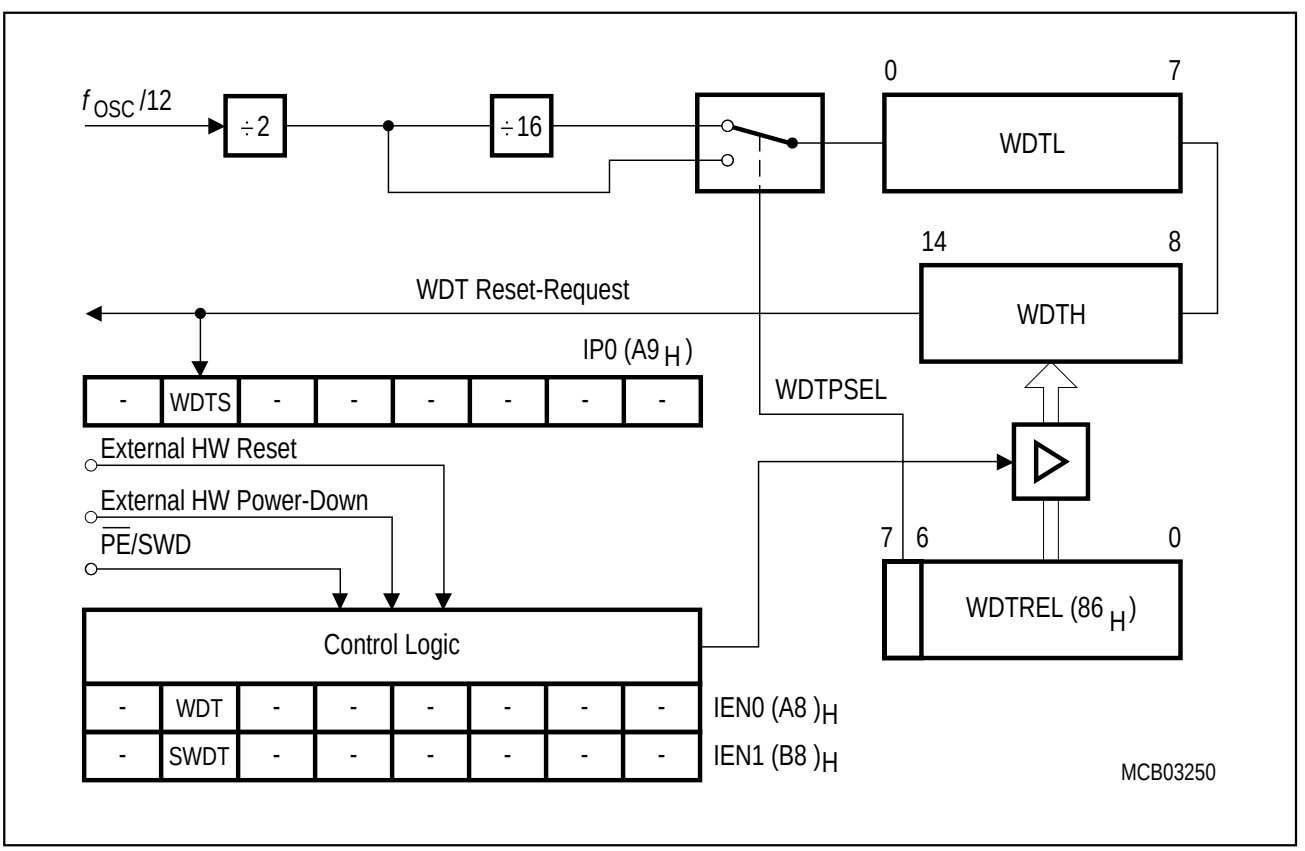
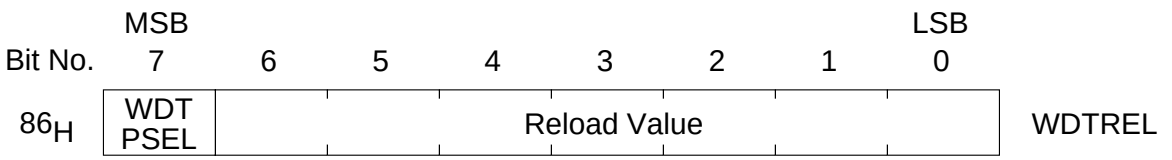


Figure 8-1
Block Diagram of the Programmable Watchdog Timer

8.1.1 Input Clock Selection

The input clock rate of the watchdog timer is derived from the system clock of the C515A. 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. **Tabel 8-1** shows resulting timeout periods at f_{OSC} = 12 and 24 MHz.

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



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

Table 8-1
Watchdog Timer Time-Out Periods (WDTPSEL = 0)

WDTREL	Time-Out Period		Comments
	f_{OSC} = 12 MHz	f_{OSC} = 24 MHz	
00_H	65.535 ms	32.768 ms	This is the default value
80_H	1.1 s	0.55 s	Maximum time period
$7F_H$	512 µs	256 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 flags (located in SFR IP0).

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

	MSB							LSB	
	AF_H	AE_H	AD_H	AC_H	AB_H	AA_H	$A9_H$	$A8_H$	
$A8_H$	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	IEN0
	BF_H	BE_H	BD_H	BC_H	BB_H	BA_H	$B9_H$	$B8_H$	
$B8_H$	EXEN2	SWDT	EX6	EX5	EX4	EX3	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 directly before SWDT is set to prevent an unintentional refresh of the watchdog timer.
SWDT	Watchdog timer start flag. Set to activate the Watchdog Timer. When directly set after setting WDT, a watchdog timer refresh is performed.
WDTS	Watchdog timer status flag. Set by hardware when a watchdog Timer reset occured. Can be cleared and set by software.

8.1.3 Starting the Watchdog Timer

Immediately after start (see next section for the start procedure), the watchdog timer is initialized to the reload value programmed to WDTREL.0 - WDTREL.6. After an external HW or $\overline{\text{HWPD}}$ reset, an oscillator power on reset, or a watchdog timer reset, register WDTREL is cleared to 00_H. WDTREL can be loaded by software at any time.

There are two ways to start the watchdog timer depending on the level applied to pin $\overline{\text{PE}}$/SWD. This pin serves two functions, because it is also used for blocking the power saving modes. (see also **chapter 9**).

8.1.3.1 The First Possibility of Starting the Watchdog Timer

The automatic start of the watchdog timer directly while an external HW reset is a hardware start initialized by strapping pin $\overline{\text{PE}}$/SWD to V_{CC}. In this case the power saving modes (power down mode, idle mode and slow down mode) are also disabled and cannot be started by software. If pin $\overline{\text{PE}}$/SWD is left unconnected, a weak pull-up transistor ensures the automatic start of the watchdog timer.

The self-start of the watchdog timer by a pin option has been implemented to provide high system security in electrically very noisy environments.

Note: The automatic start of the watchdog timer is only performed if $\overline{\text{PE}}$/SWD (power-save enable/ start watchdog timer) is held at high level while $\overline{\text{RESET}}$ or $\overline{\text{HWPD}}$ is active. A positive transition at these pins during normal program execution will not start the watchdog timer. Furthermore, when using the hardware start, the watchdog timer starts running with its default time-out period. The value in the reload register WDTREL, however, can be overwritten at any time to set any time-out period desired.

8.1.3.2 The Second Possibility of Starting the Watchdog Timer

The watchdog timer can also be started by software. Setting of bit SWDT in SFR IEN1 starts the watchdog timer. Using the software start, the timeout period can be programmed before the watchdog timer starts running.

Note that once the watchdog timer has been started it can only be stopped if one of the following conditions are met :

- active external hardware reset through pin $\overline{\text{RESET}}$ with a low level at pin $\overline{\text{PE}}$/SWD
- active hardware power down signal $\overline{\text{HWPD}}$, independently of the level at $\overline{\text{PE}}$/SWD
- entering idle mode or power down mode by software

See **chapter 9** for entering the power saving modes by software.

8.1.4 Refreshing the Watchdog Timer

At the same time the watchdog timer is started, the 7-bit register WDTH is preset by the contents of WDTREL.0 to WDTREL.6. Once started the watchdog cannot be stopped by software but can only be refreshed to the reload value by first setting bit WDT (IEN0.6) and by the next instruction setting SWDT (IEN1.6). Bit WDT will automatically be cleared during the second machine cycle after having been set. For this reason, setting SWDT bit has to be a one cycle instruction (e.g. SETB SWDT). This double-instruction refresh of the watchdog timer is implemented to minimize the chance of an unintentional reset of the watchdog.

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

8.1.5 Watchdog Reset and Watchdog Status Flag

If the software fails to clear the watchdog in time, an internally generated watchdog reset is entered at the counter state $7FFC_H$. The duration of the reset signal then depends on the prescaler selection (either 8 cycles or 128 cycles). This internal reset differs from an external one only in so far as the watchdog timer is not disabled and bit WDTS (watchdog timer status, bit 6 in SFR IP0) is set. **Figure 8-2** shows a block diagram of all reset requests in the C515A 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.

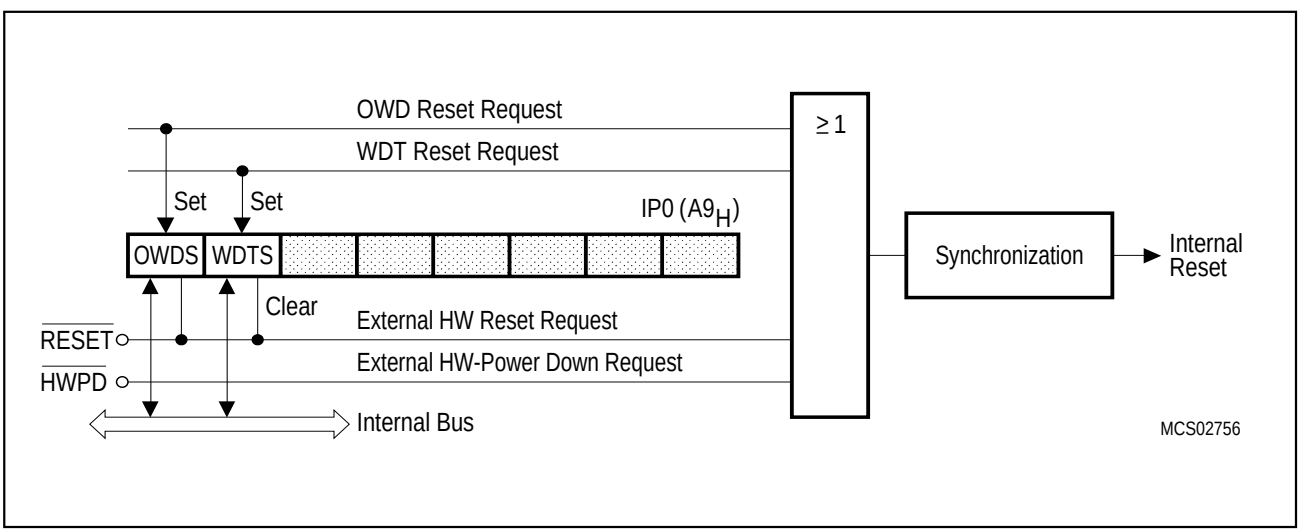


Figure 8-2
Watchdog Timer Status Flags and Reset Requests

8.2 Oscillator Watchdog Unit

The oscillator watchdog unit serves for four functions:

- **Monitoring of the on-chip oscillator's function**
  The watchdog supervises the on-chip oscillator's frequency; if it is lower than the frequency of the auxiliary RC oscillator in the watchdog unit, the internal clock is supplied by the RC oscillator and the device is brought into reset; if the failure condition disappears (i.e. the on-chip oscillator has a higher frequency than the RC oscillator), the part executes a final reset phase of typ. 1 ms in order to allow the oscillator to stabilize; then the oscillator watchdog reset is released and the part starts program execution again.
- **Fast internal reset after power-on**
  The oscillator watchdog unit provides a clock supply for the reset before the on-chip oscillator has started. The oscillator watchdog unit also works identically to the monitoring function.
- **Restart from the hardware power down mode.**
  If the hardware power down mode is terminated the oscillator watchdog has to control the correct start-up of the on-chip oscillator and to restart the program. The oscillator watchdog function is only part of the complete hardware power down sequence; however, the watchdog works identically to the monitoring function.
- **Control of external wake-up from software power-down mode**
  When the power-down mode is left by a low level at the P3.2/$\overline{\text{INT0}}$ pin, the oscillator watchdog unit assures that the microcontroller resumes operation (execution of the power-down wake-up interrupt) with the nominal clock rate. In the power-down mode the RC oscillator and the on-chip oscillator are stopped. Both oscillators are started again when power-down mode is released. When the on-chip oscillator has a higher frequency than the RC oscillator, the microcontroller starts operation after a final delay of typ. 1 ms in order to allow the on-chip oscillator to stabilize.

Note: The oscillator watchdog unit is always enabled. If $\overline{\text{PE}}$/SWD is low and V_{AREF} is low the oscillator watchdog is disabled (testmode) !

8.2.1 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 the comparison with the frequency of the on-chip oscillator. It also shows the modifications which have been made for integration of the wake-up from power down mode capability.

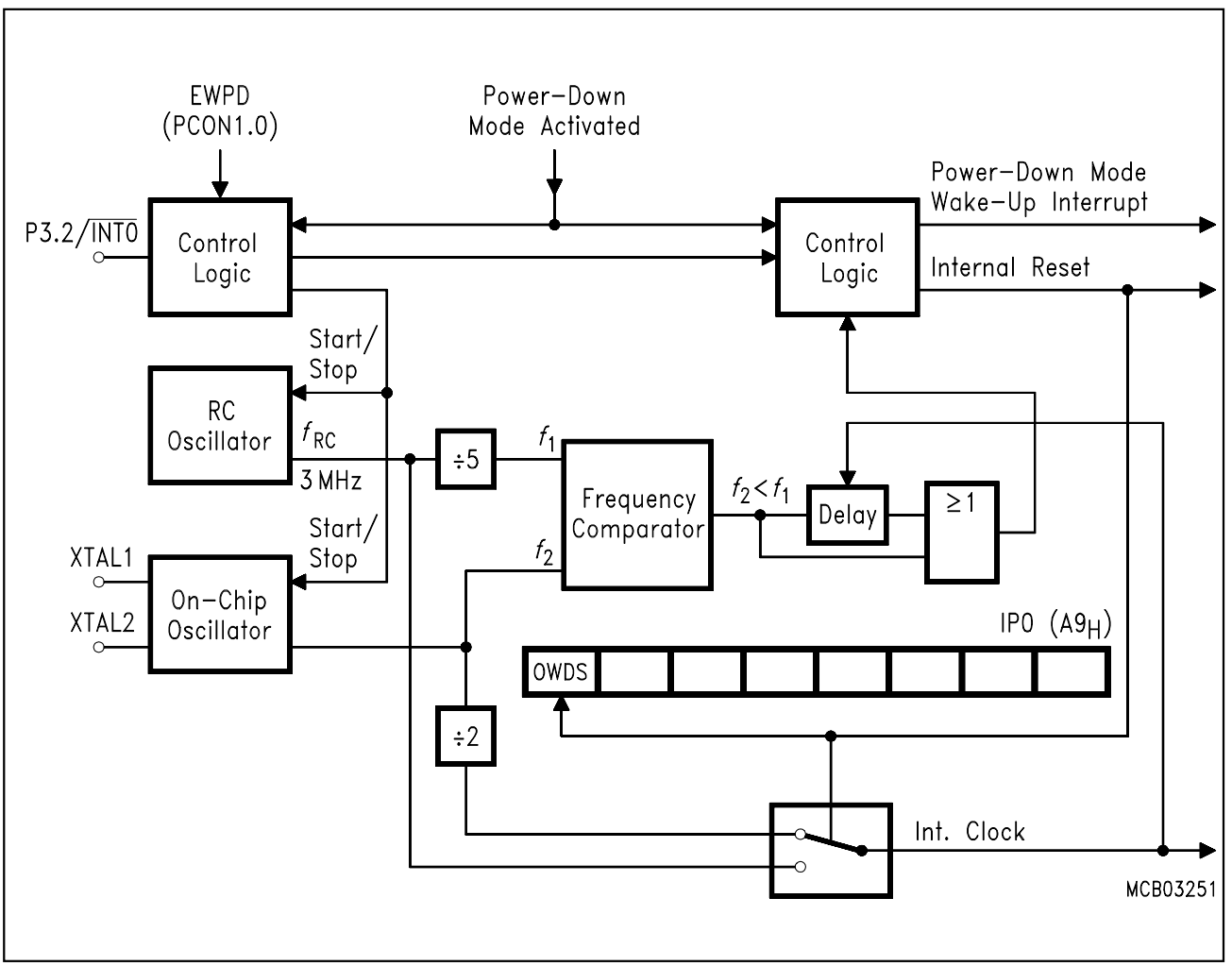


Figure 8-3
Functional Block Diagram of the Oscillator Watchdog

The frequency coming from the RC oscillator is divided by 5 and compared to the on-chip oscillator's frequency. If the frequency coming from the on-chip oscillator is found lower than the frequency derived from the RC oscillator the watchdog detects a failure condition (the oscillation at the on-chip oscillator could stop because of crystal damage etc.). In this case it switches the input of the internal clock system to the output of the RC oscillator. This means that the part is being clocked even if the on-chip oscillator has stopped or has not yet started. At the same time the watchdog activates the internal reset in order to bring the part in its defined reset state. The reset is performed because clock is available from the RC oscillator. This internal watchdog reset has the same effects as an externally applied reset signal with the following exceptions: The Watchdog Timer Status flag WDTS is not reset (the Watchdog Timer however is stopped); and bit OWDS is set. This allows the

software to examine error conditions detected by the Watchdog Timer even if meanwhile an oscillator failure occured.

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

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

If software power-down mode is activated the RC oscillator and the on-chip oscillator is stopped. Both oscillators are again started in power-down mode when a low level is detected at the P3.2/$\overline{\text{INT0}}$ input pin and when bit EWPD in SFR PCON1 is set (wake-up from power-down mode enabled). After the start-up phase of the watchdog circuitry in power-down mode, a power-down mode wake-up interrupt is generated (instead of an internal reset). Detailed description of the wake-up from software power-down mode is given in section 9.4.2.

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 Timer Status Flag. Set by hardware when an oscillator watchdog reset occured. Can be set and cleared by software.

8.2.2 Fast Internal Reset after Power-On

The C515A can use the oscillator watchdog unit for a fast internal reset procedure after power-on.

Normally the members of the 8051 family (e. g. SAB 80C52) enter their default reset state not before the on-chip oscillator starts. The reason is that the external reset signal must be internally synchronized and processed in order to bring the device into the correct reset state. Especially if a crystal is used the start up time of the oscillator is relatively long (typ. 1 ms). During this time period the pins have an undefined state which could have severe effects e.g. to actuators connected to port pins.

In the C515A the oscillator watchdog unit avoids this situation. After power-on the oscillator watchdog's RC oscillator starts working within a very short start-up time (typ. less than 2 microseconds). In the following the watchdog circuitry detects a failure condition for the on-chip oscillator because this has not yet started (a failure is always recognized if the watchdog's RC oscillator runs faster than the on-chip oscillator). As long as this condition is valid the watchdog uses the RC oscillator output as clock source for the chip. This allows correct resetting of the part and brings all ports to the defined state (see also **chapter 5** of this manual). The delay time between power-on and correct reset state is max 34 µs (more details see **chapter 5.2**).

9 Power Saving Modes

The C515A 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 be also 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 which are located in the special function registers PCON und PCON1. The SFR PCON is located at SFR address 87_H. PCON1 is located in the mapped SFR area (RMAP=1) at SFR address 88_H.. Bit RMAP, which controls the access to the mapped SFR area, is located in SFR SYSCON ($B1_H$).

The bits PDE, PDS and IDLE, IDLS located in SFR PCON select the power down mode or the idle mode, respectively. If the power down mode and the idle mode are set at the same time, power down takes precedence.

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

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

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

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

Note: The PDS bit, which controls the software power down mode is forced to logic low whenever the external $\overline{PE}$/SWD pin is held at logic high level. Changing the logic level of the $\overline{PE}$/SWD pin from high to low will irregularly terminate the software power down mode an is not permitted.

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

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

Symbol	Function
–	Reserved bits for future use.
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 via the pin P3.2/$\overline{INT0}$ (more details see **section 9.2**).

9.2 Idle Mode

In the idle mode the oscillator of the C515A continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D converter,and all timers with the exception of the watchdog timer are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

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

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

Normally, the port pins hold the logical state they had at the time when the idle mode was activated. If some pins are programmed to serve as alternate functions they still continue to output during idle mode if the assigned function is on. This especially applies to the system clock output signal at pin P1.6/CLKOUT and to the serial interfaces in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ are hold 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 an unintentional entering of the idle mode which would leave the watchdog timer's task of system protection without effect.

Note:

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

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

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

There are two ways to terminate the idle mode:

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

9.3 Slow Down Mode Operation

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

In the slow down mode all signal frequencies that are derived from the oscillator clock are divided by 8. This also includes the clock output signal at pin P1.6/CLKOUT. Further, if the slow down mode is used pin $\overline{PE}$/SWD must be held low.

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 1/8 of the nominal system clock rate. The controller actually enters the slow down mode after a short synchronization period (max. two machine cycles). The slow down mode is disabled by clearing bit SD.

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

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

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

- The idle mode can be terminated by activation of any enabled interrupt. The CPU operation is resumed, the interrupt will be serviced and the next instruction to be executed after the RETI instruction will be the one following the instruction that sets the bits IDLS and SD. Nevertheless the slow down mode keeps enabled and if required has to be disabled by clearing the bit SD in the corresponding interrupt service routine or after the instruction that sets the bits IDLS and SD.
- 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 has to 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 osciillator and the on-chip oscillator which operates with the XTAL pins is stopped. Therefore, all functions of the microcontroller are stopped and only the contents of the on-chip RAM, XRAM and the SFR's are maintained. The port pins, which are controlled by their port latches, output the values that are held by their SFR's. The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the software power down mode. ALE and $\overline{\text{PSEN}}$ hold at logic low level (see **table 9-1**).

In the software power down mode of operation, V_{CC} can be reduced to minimize power consumption. It must be ensured, however, that is V_{CC} not reduced before the software power down mode is invoked, and that V_{CC} is restored to its normal operating level before the software 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 the P3.2/$\overline{\text{INT0}}$ pin. Using reset to leave software power down mode puts the microcontroller with its SFRs into the reset state. Using the P3.2/$\overline{\text{INT0}}$ pin for software power down mode exit starts the RC oscillator and the on-chip oscillator and maintains the state of the SFRs, which has been frozen when software power down mode is entered. Leaving software power down mode should not be done before V_{CC} 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 has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6), the following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, does not initiate the software 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 software 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 software 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 software power down mode. When the double instruction sequence shown above is used, the software power down mode can only be left by a reset operation. If the external wake-up from power down capability has also to 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 external wake-up from software power down by
                             ;setting EWPD
ANL     SYSCON,#11101111B   ;reset RMAP (for future SFR accesses)
```

Setting EWPD automatically disables all interrupts still maintaining all actual values of the interrupt enable bits.

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 software power down mode is exit via a hardware reset, the microcontroller with its SFRs is put into the hardware reset state and the content of RAM and XRAM are not changed. The reset signal that terminates the software power down mode also restarts the RC oscillator and the on-chip oscillatror. The reset operation should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (similar to power-on reset).

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

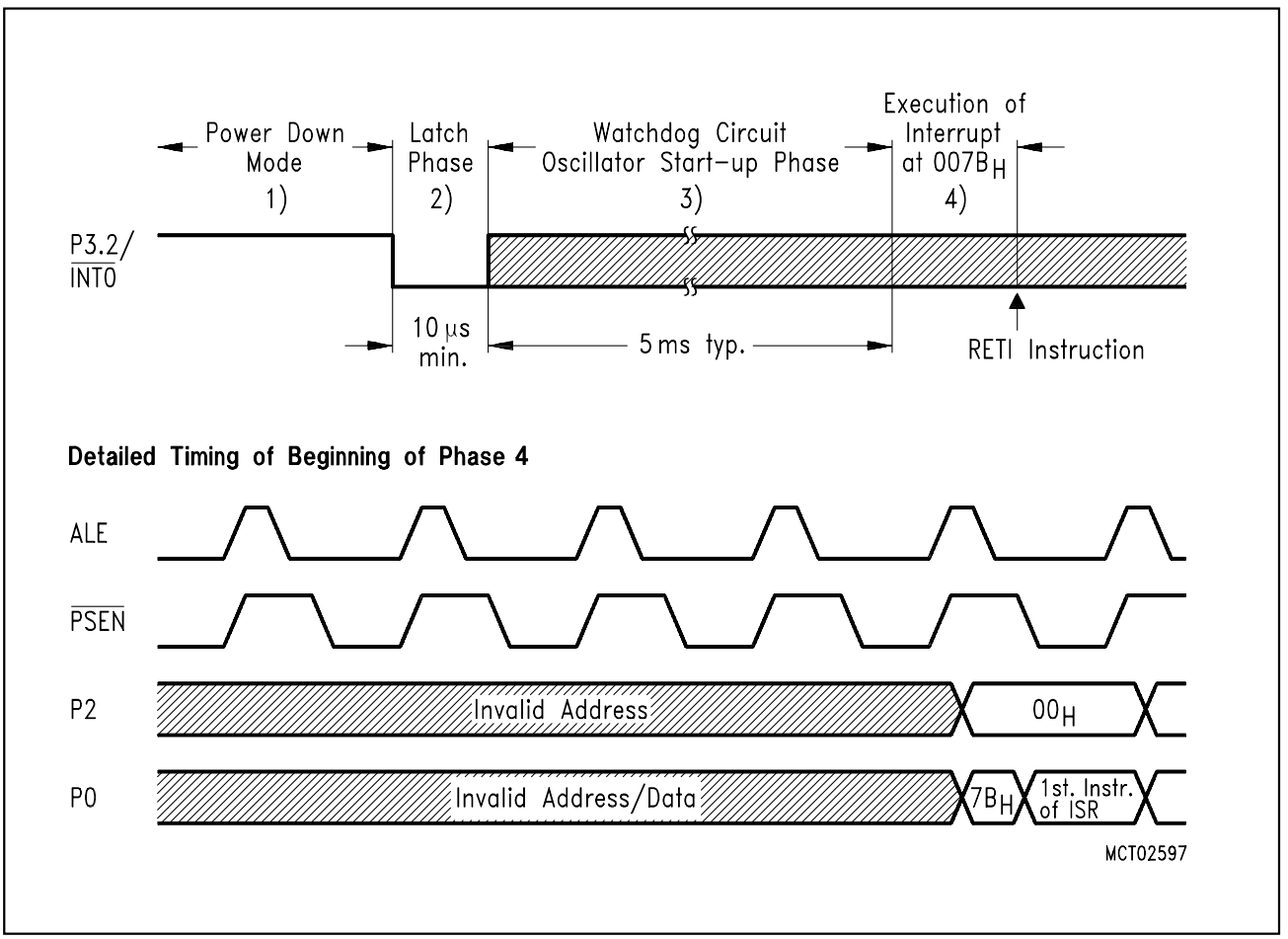


Figure 9-1
Wake-up from Power Down Mode Procedure

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

1. In software power down mode pin P3.2/$\overline{\text{INT0}}$ must be held at high level.
2. software power down mode is left when P3.2/$\overline{\text{INT0}}$ goes low for at least 10 µs (latch phase). After this delay the internal RC oscillator and the on-chip oscillator are started, the state of pin P3.2/$\overline{\text{INT0}}$ is internally latched, and P3.2/$\overline{\text{INT0}}$ can be set again to high level if required. Thereafter, the oscillator watchdog unit controls the wake-up procedure in its start-up phase.

3. The oscillator watchdog unit starts operation. When the on-chip oscillator clock is detected for stable nominal frequency, the microcontroller starts again with its operation initiating the power down wake-up interrupt. The interrupt address of the first instruction to be executed after wake-up is $007B_H$.
4. After the RETI instruction of the power down wake-up interrupt routine has been executed, the instruction which follows the initiating software power down mode double instruction sequence will be executed. The peripheral units timer 0/1/2 , SSC, CAN controller, and WDT are frozen until end of phase 4.

All interrupts of the C515A are disabled from phase 2) until the end of phase 4). Other Interrupts can be first handled after the RETI instruction of the wake-up interrupt routine.

9.5 State of Pins in Software Initiated Power Saving Modes

In the idle mode and in the software power down mode the port pins of the C515A have a well defined status which is listed in the following **table 9-1**. This state of some pins also depends on the location of the code memory (internal or external).

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

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
PORT2	Data	Data	Address	Data
PORT1, 3, 5	Data / alternate outputs	Data / last output	Data / alternate outputs	Data / last output
PORT 5	Data	Data	Data	Data
P7.0	Data	Data	Data	Data

9.6 Hardware Power Down Mode

The power down mode of the C515A can also be initiated by an external signal at the pin $\overline{\text{HWPD}}$. Because this power down mode is activated by an external hardware signal it mode is referred to as hardware power down mode in opposite to the program controlled software power down mode.

Pin $\overline{\text{PE}}$/SWD has no control function for the hardware power down mode; it enables and disables only the use of all software controlled power saving modes (idle mode, software power down mode).

The function of the hardware power down mode is as follows:

– The pin $\overline{\text{HWPD}}$ controls this mode. If it is on logic high level (inactive) the part is running in the normal operating modes. If pin $\overline{\text{HWPD}}$ gets active (low level) the part enters the hardware power down mode; as mentioned above this is independent of the state of pin $\overline{\text{PE}}$/SWD.

$\overline{\text{HWPD}}$ is sampled once per machine cycle. If it is found active, the device starts a complete internal reset sequence. This takes two machine cycles; all pins have their default reset states during this time. This reset has exactly the same effects as a hardware reset; i.e.especially the watchdog timer is stopped and its status flag WDTS is cleared. In this phase the power consumption is not yet reduced. After completion of the internal reset both oscillators of the chip are disabled, the on-chip oscillator as well as the oscillator watchdog's RC oscillator. At the same time the port pins and several control lines enter a floating state as shown in **table 9-2**. In this state the power consumption is reduced to the power down current I_{PD}. Also the supply voltage can be reduced.

Table 9-2 also lists the voltages which may be applied at the pins during hardware power down mode without affecting the low power consumption.

Table 9-2
Status of all Pins During Hardware Power Down Mode

Pins	Status	Voltage Range at Pin During HW-Power Down
P0, P1, P2, P3, P4, P5	Floating outputs/ Disabled input function	$V_{SS} \le V_{IN} \le V_{CC}$
$\overline{\text{EA}}$	Active input	$V_{IN} = V_{CC}$ or $V_{IN} = V_{SS}$
$\overline{\text{PE}}$/SWD	Active input, Pull-up resistor Disabled during HW power down	$V_{IN} = V_{CC}$ or $V_{IN} = V_{SS}$
XTAL 1	Active output	pin may not be driven
XTAL 2	Disabled input function	$V_{SS} \le V_{IN} \le V_{CC}$
$\overline{\text{PSEN}}$, ALE	Floating outputs/ Disabled input function (for test modes only)	$V_{SS} \le V_{IN} \le V_{CC}$
$\overline{\text{RESET}}$	Active input; must be at high level if $\overline{\text{HWPD}}$ is used	$V_{IN} = V_{CC}$
V_{AREF}	ADC reference supply input	$V_{SS} \le V_{IN} \le V_{CC}$

The hardware power down mode is maintained while pin $\overline{\text{HWPD}}$ is held active. If $\overline{\text{HWPD}}$ goes to high level (inactive state) an automatic start up procedure is performed:

- First the pins leave their floating condition and enter their default reset state as they had immediately before going to float state.
- Both oscillators are enabled. While the on-chip oscillator (with pins XTAL1 and XTAL2) usually needs a longer time for start-up, if not externally driven (with crystal approx. 1 ms), the oscillator watchdog's RC oscillator has a very short start-up time (typ. less than 2 µs).
- Because the oscillator watchdog is active it detects a failure condition if the on-chip oscillator hasn't yet started. Hence, the watchdog keeps the part in reset and supplies the internal clock from the RC oscillator.
- Finally, when the on-chip oscillator has started, the oscillator watchdog releases the part from reset after it performed a final internal reset sequence and switches the clock supply to the on-chip oscillator. This is exactly the same procedure as when the oscillator watchdog detects first a failure and then a recovering of the oscillator during normal operation. Therefore, also the oscillator watchdog status flag is set after restart from hardware power down mode.
  When automatic start of the watchdog was enabled ($\overline{\text{PE}}$/SWD connected to V_{CC}), the watchdog timer will start, too (with its default reload value for time-out period).

The SWD-Function of the $\overline{\text{PE}}$/SWD Pin is sampled only by a hardware reset. Therefore at least one power-on reset has to be performed.

9.7 Hardware Power Down Reset Timing

The following figures show the timing diagrams for entering (**figure 9-2**) and leaving (**figure 9-3**) the hardware power down mode. If there is only a short signal at pin $\overline{\text{HWPD}}$ (i.e. $\overline{\text{HWPD}}$ is sampled active only once), then a complete internal reset is executed. Afterwards, the normal program execution starts again (**figure 9-4**).

Note: Delay time caused by internal logic is not included.

The $\overline{\text{RESET}}$ pin overrides the hardware power down function, i.e. if reset gets active during hardware power down it is terminated and the device performs the normal reset function. Thus, pin $\overline{\text{RESET}}$ has to be inactive during hardware power down mode.

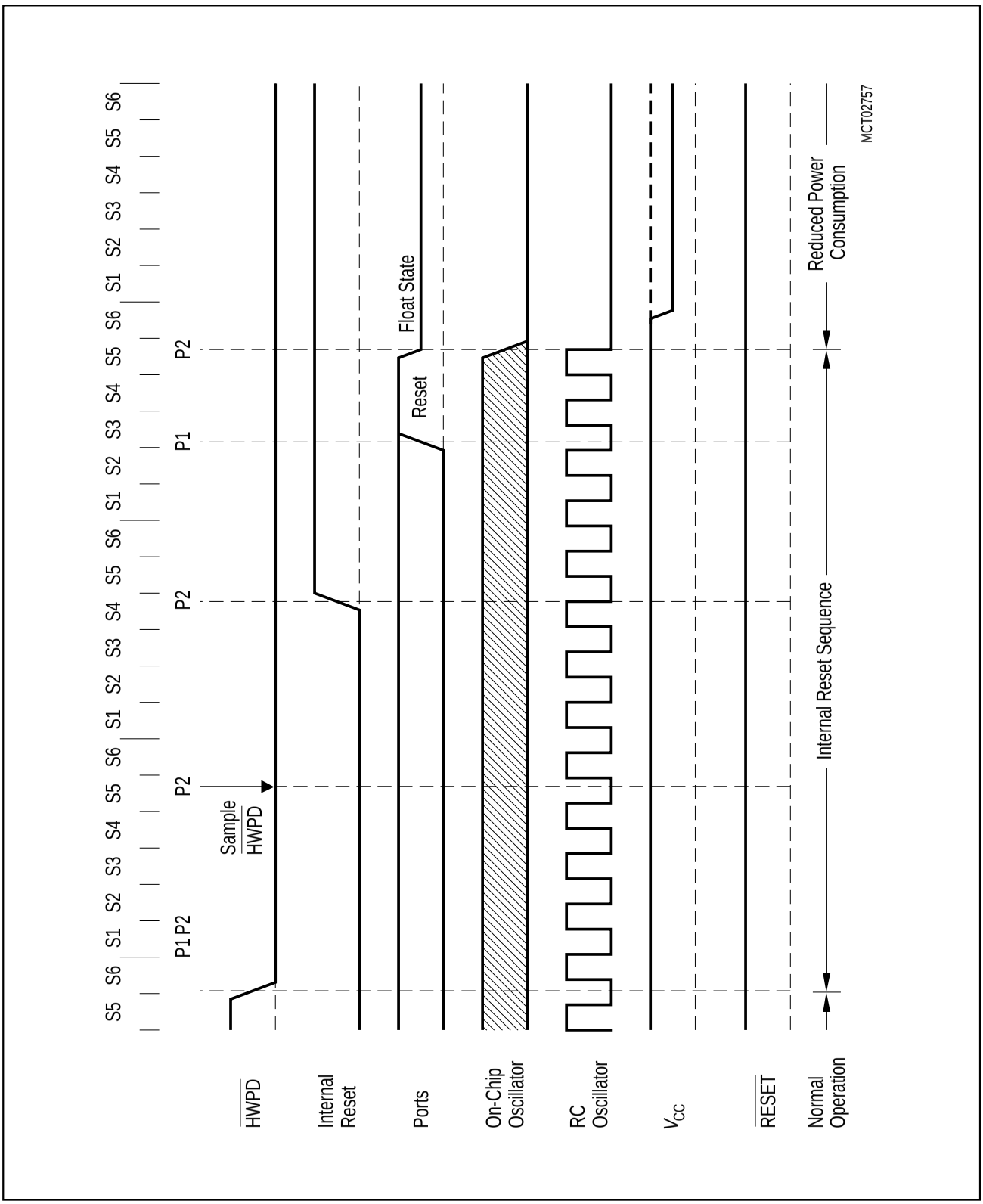


Figure 9-2
Timing Diagram of Entering Hardware Power Down Mode

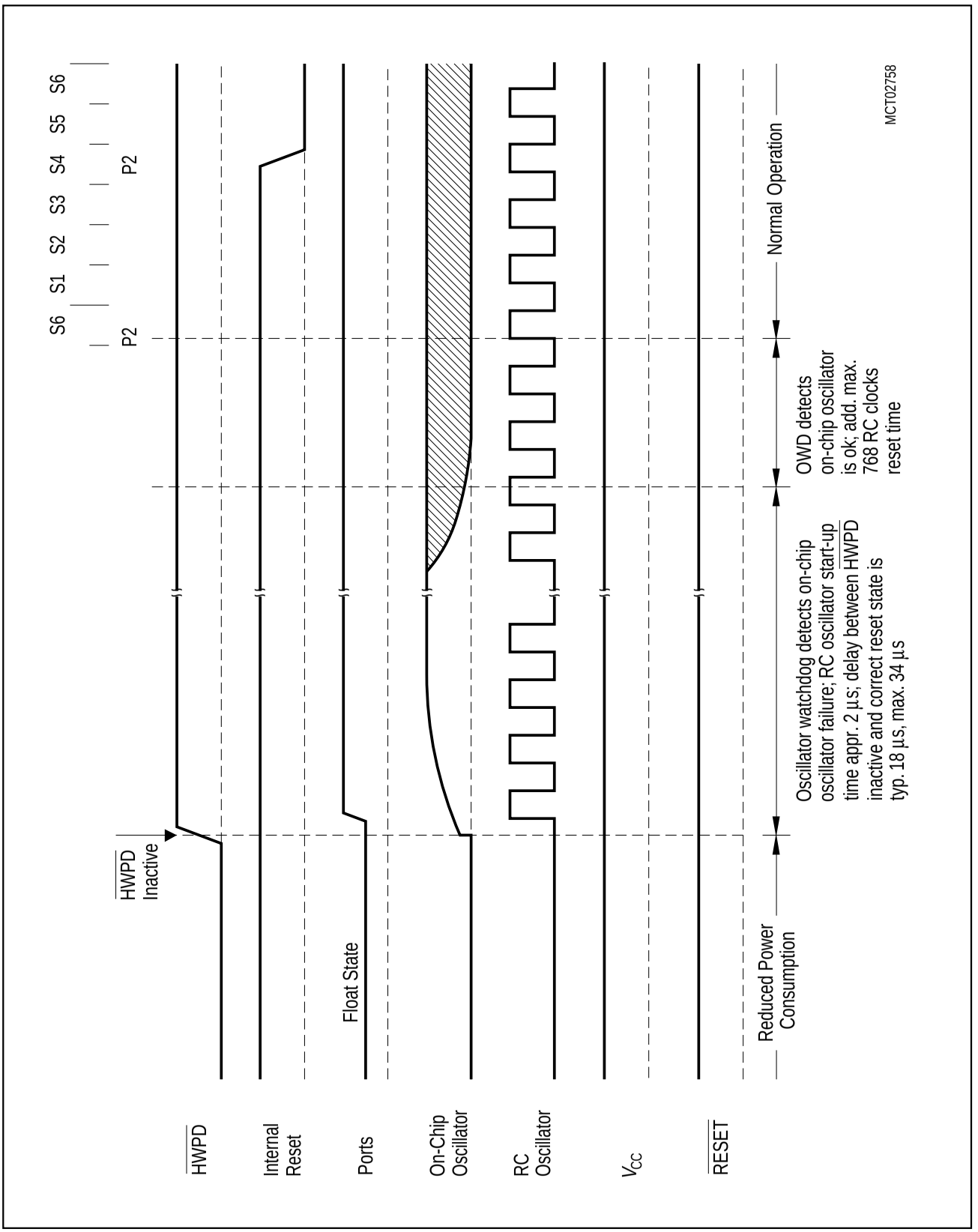


Figure 9-3
Timing Diagram of Leaving Hardware Power Down Mode

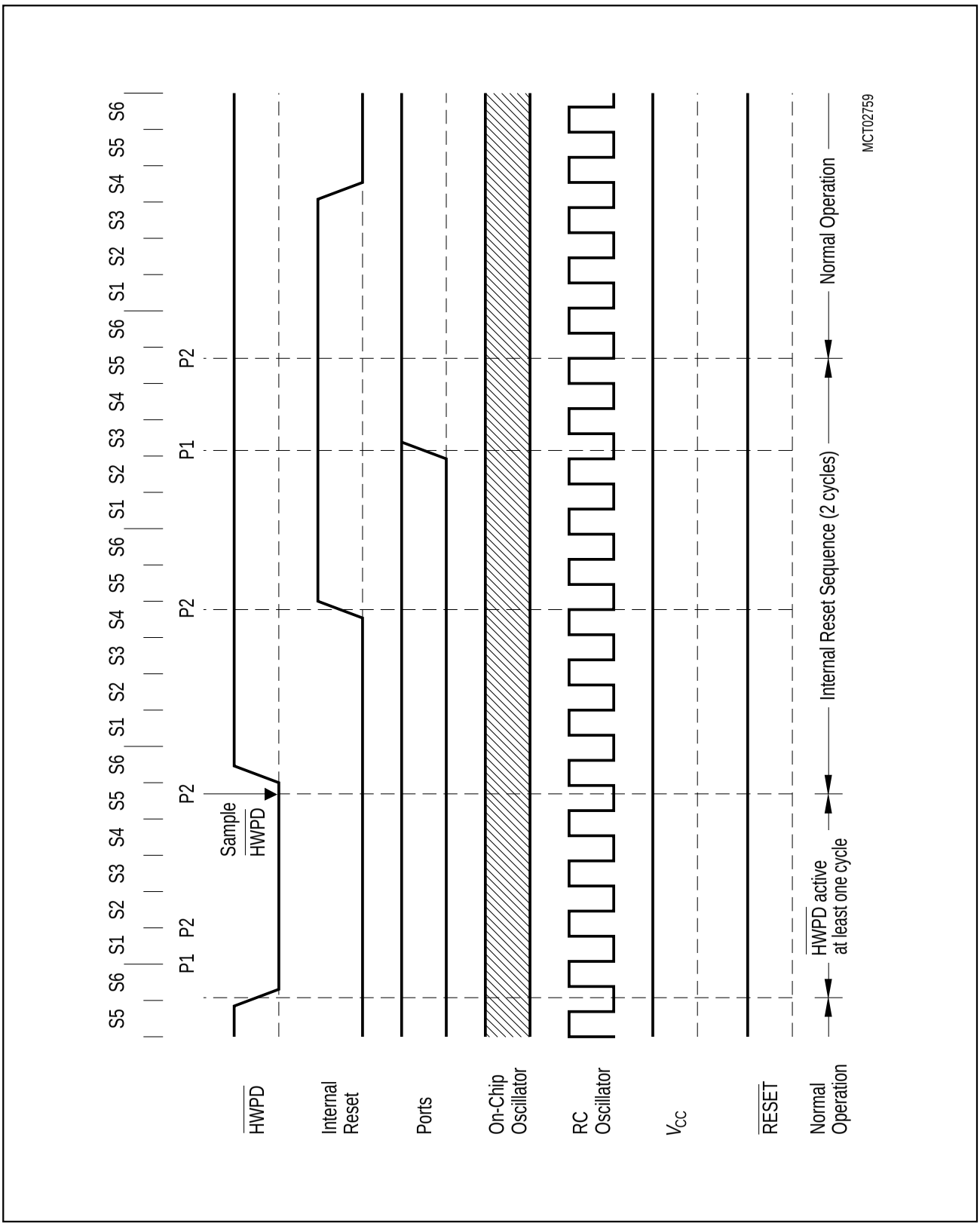


Figure 9-4
Timing Diagram of Hardware Power Down Mode, $\overline{\text{HWPD}}$-Pin is active for only one cycle

10 Device Specifications

10.1 Absolute Maximum Ratings

Ambient temperature under bias (T_A) .....– 40 ˚C to + 125 ˚C
Storage temperature (T_{ST}) .....– 65 ˚C to + 150 ˚C
Voltage on V_{CC} pins with respect to ground (V_{SS}) .....– 0.5 V to 6.5 V
Voltage on any pin with respect to ground (V_{SS}) .....– 0.5 V to V_{CC} + 0.5 V
Input current on any pin during overload condition .....– 10 mA to + 10 mA
Absolute sum of all input currents during overload condition .....| 100 mA |
Power dissipation of package ..... TBD

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

10.2 DC Characteristics

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

T_A = 0 to 70 °C for the SAB-C515A
T_A = – 40 to 85 °C for the SAF-C515A
T_A = – 40 to 110 °C for the SAH-C515A
T_A = – 40 to 125 °C for the SAK-C515A

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage					
Pins except $\overline{EA}$,$\overline{RESET}$,$\overline{HWPD}$	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
$\overline{EA}$ pin	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
$\overline{HWPD}$ and $\overline{RESET}$ pins	V_{IL2}	– 0.5	0.2 V_{CC} + 0.1	V	–
Input high voltage					
pins except $\overline{RESET}$, XTAL2 and $\overline{HWPD}$	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
XTAL2 pin	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	–
$\overline{RESET}$ and $\overline{HWPD}$ pin	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage					
Ports 1, 2, 3, 4, 5	V_{OL}	–	0.45	V	I_{OL} = 1.6 mA [1]
Port 0, ALE, $\overline{PSEN}$	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA [1]
Output high voltage					
Ports 1, 2, 3, 4, 5	V_{OH}	2.4	–	V	I_{OH} = – 80 µA
		0.9 V_{CC}	–	V	I_{OH} = – 10 µA
Port 0 in external bus mode,	V_{OH1}	2.4	–	V	I_{OH} = – 800 µA
ALE, $\overline{PSEN}$		0.9 V_{CC}	–	V	I_{OH} = – 80 µA [2]
Logic 0 input current					
Ports 1, 2, 3, 4, 5	I_{LI}	– 10	– 70	µA	V_{IN} = 0.45 V
Logical 0-to-1 transition current,					
Ports 1, 2, 3, 4, 5	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current					
Port 0 and 6, $\overline{EA}$, $\overline{HWPD}$	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Input low current					
to $\overline{RESET}$ for reset	I_{IL2}	– 10	– 100	µA	V_{IN} = 0.45 V
XTAL2	I_{IL3}	–	– 15	µA	V_{IN} = 0.45 V
$\overline{PE}$/SWD	I_{IL4}	–	– 20	µA	V_{IN} = 0.45 V
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25˚C
Overload current	I_{OV}	–	± 5	mA	[8] [9]

Notes see next page

Power Supply Current

Parameter		Symbol	Limit Values		Unit	Test Condition
			typ. [10]	max. [11]		
Active mode	18 MHz	I_{CC}	16.9	23.1	mA	[4]
	24 MHz	I_{CC}	21.7	29.4	mA	
Idle mode	18 MHz	I_{CC}	8.5	12.1	mA	[5]
	24 MHz	I_{CC}	11.0	15.0	mA	
Active mode with slow-down enabled	18 MHz	I_{CC}	5.6	8.0	mA	[6]
	24 MHz	I_{CC}	6.6	9.6	mA	
Active mode with slow-down enabled	18 MHz	I_{CC}	3.0	4.1	mA	[7]
	24 MHz	I_{CC}	3.3	4.7	mA	
Power-down mode		I_{PD}	10	50	µA	V_{CC} = 2…5.5 V [3]

Notes :

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

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

3) I_{PD} (software power-down mode) is measured under following conditions:
EA = $\overline{\text{RESET}}$ = Port 0 = Port 6 = V_{CC} ; XTAL1 = N.C.; XTAL2 = V_{SS} ; $\overline{\text{PE}}$/SWD = V_{SS} ; $\overline{\text{HWPD}}$ = V_{CC} ; V_{AGND} = V_{SS} ; V_{AREF} = V_{CC} ; all other pins are disconnected.
I_{PD} (hardware power-down mode): independent from any particular pinn connection.

4) I_{CC} (active mode) is measured with:
XTAL2 driven with t_{CLCH} , t_{CHCL} = 5 ns , V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.;
EA = $\overline{\text{PE}}$/SWD = Port 0 = Port 6 = V_{CC} ; $\overline{\text{HWPD}}$ = V_{CC} ; $\overline{\text{RESET}}$ = V_{SS} ;
all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (appr. 1 mA).

5) I_{CC} (idle mode) is measured with all output pins disconnected and with all peripherals disabled;
XTAL2 driven with t_{CLCH} , t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.;
$\overline{\text{RESET}}$ = V_{CC} ; $\overline{\text{HWPD}}$ = Port 0 = Port 6 = V_{CC} ; EA = $\overline{\text{PE}}$/SWD = V_{SS} ; all other pins are disconnected;

6) I_{CC} (active mode with slow-down mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with t_{CLCH} , t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.;
$\overline{\text{RESET}}$ = V_{CC} ; $\overline{\text{HWPD}}$ = Port 6 = V_{CC} ; EA = $\overline{\text{PE}}$/SWD = V_{SS} ; all other pins are disconnected; the microcontroller is put into slow-down mode by software;

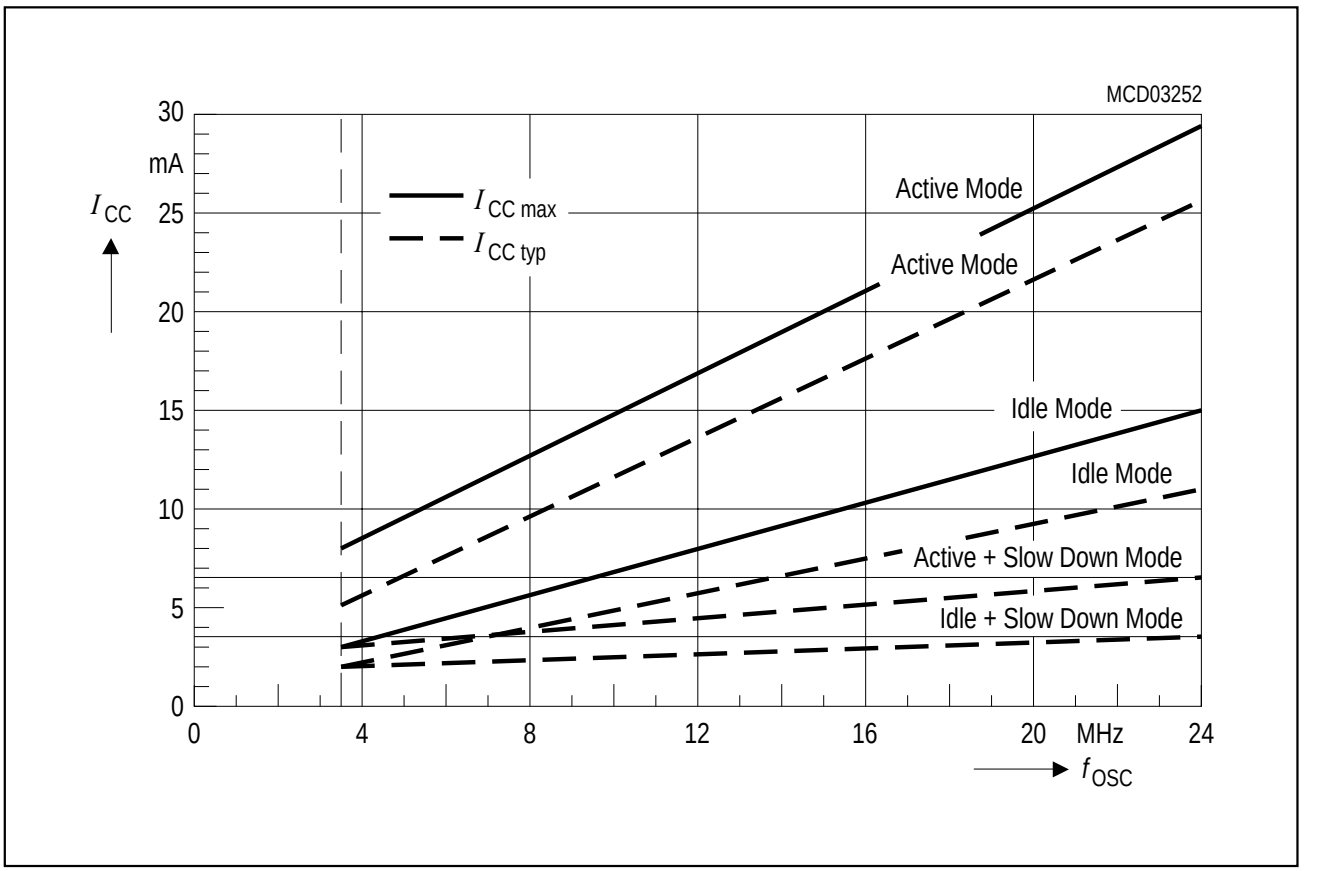
7) I_{CC} (idle mode with slow-down mode) is measured with all output pins disconnected and with all peripherals disabled; XTAL2 driven with t_{CLCH} , t_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5 V, V_{IH} = V_{CC} – 0.5 V; XTAL1 = N.C.;
$\overline{\text{RESET}}$ = V_{CC} ; $\overline{\text{HWPD}}$ = Port 6 = V_{CC} ; EA = $\overline{\text{PE}}$/SWD = V_{SS} ; all other pins are disconnected;
the microcontroller is put into idle mode with slow-down mode enabled by software;

8) Overload conditions occur if the standard operating conditions are exceeded, ie. the voltage on any pin exceeds the specified range (i.e. V_{OV} > V_{CC} + 0.5 V or V_{OV} < V_{SS} - 0.5 V). The supply voltage V_{CC} and V_{SS} must remain within the specified limits. The absolute sum of input currents on all port pins may not exceed 50 mA.

9) Not 100% tested, guaranteed by design characterization

10) The typical I_{CC} values are periodically measured at T_A = +25 °C and V_{CC} = 5 V but not 100% tested.

11) The maximum I_{CC} values are measured under worst case conditions (T_A = 0 °C or -40 °C and V_{CC} = 5.5 V)



ICC Diagram

Power Supply Current Calculation Formulas

Parameter	Symbol	Formula
Active mode	$I_{CC\ typ}$ $I_{CC\ max}$	0.79 * f_{OSC} + 2.7 1.04 * f_{OSC} + 4.4
Idle mode	$I_{CC\ typ}$ $I_{CC\ max}$	0.43 * f_{OSC} + 0.7 0.48 * f_{OSC} + 3.5
Active mode with slow-down enabled	$I_{CC\ typ}$ $I_{CC\ max}$	0.17 * f_{OSC} + 2.5 0.28 * f_{OSC} + 2.9
Idle mode with slow-down enabled	$I_{CC\ typ}$ $I_{CC\ max}$	0.06 * f_{OSC} + 1.9 0.09 * f_{OSC} + 2.5

Note : f_{osc} is the oscillator frequency in MHz. I_{CC} values are given in mA.

10.3 A/D Converter Characteristics

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

T_A = 0 to 70 °C for the SAB-C515A
T_A = – 40 to 85 °C for the SAF-C515A
T_A = – 40 to 110 °C for the SAH-C515A
T_A = – 40 to 125 °C for the SAK-C515A

4 V ≤ V_{AREF} ≤ V_{CC} + 0.1 V ; V_{SS} – 0.1 V ≤ V_{AGND} ≤ V_{SS} + 0.2 V

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Analog input voltage	V_{AIN}	V_{AGND}	V_{AREF}	V	1)
Sample time	t_S	–	$16 \times t_{IN}$ $8 \times t_{IN}$	ns	Prescaler ÷ 8 Prescaler ÷ 4 2)
Conversion cycle time	t_{ADCC}	–	$96 \times t_{IN}$ $48 \times t_{IN}$	ns	Prescaler ÷ 8 Prescaler ÷ 4 3)
Total unadjusted error	T_{UE}	–	± 2	LSB	V_{SS} + 0.5 V ≤ V_{IN} ≤ V_{CC} – 0.5 V 4)
Internal resistance of reference voltage source	R_{AREF}	–	t_{ADC} / 250 – 1	kΩ	t_{ADC} in [ns] 5) 6)
Internal resistance of analog source	R_{ASRC}	–	t_S / 500 – 0.8	kΩ	t_S in [ns] 2) 6)
ADC input capacitance	C_{AIN}	–	50	pF	6)

Notes see next page.

Clock calculation table :

Clock Prescaler Ratio	ADCL	t_{ADC}	t_S	t_{ADCC}
÷ 8	1	$8 \times t_{IN}$	$16 \times t_{IN}$	$96 \times t_{IN}$
÷ 4	0	$4 \times t_{IN}$	$8 \times t_{IN}$	$48 \times t_{IN}$

Further timing conditions : t_{ADC} min = 500 ns
t_{IN} = 2 / f_{OSC} = 2 t_{CLCL}

Notes:

1) V_{AIN} may exeed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases will be $X000_H$ or $X3FF_H$, respectively.

2) During the sample time the input capacitance C_{AIN} can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach their final voltage level within t_S. After the end of the sample time t_S, changes of the analog input voltage have no effect on the conversion result.

3) This parameter includes the sample time t_S, the time for determining the digital result and the time for the calibration. Values for the conversion clock t_{ADC} depend on programming and can be taken from the table on the previous page.

4) T_{UE} is tested at V_{AREF} = 5.0 V, V_{AGND} = 0 V, V_{CC} = 4.9 V. It is guaranteed by design characterization for all other voltages within the defined voltage range.
   If an overload condition occurs on maximum 2 not selected analog input pins and the absolute sum of input overload currents on all analog input pins does not exceed 10 mA, an additional conversion error of 1/2 LSB is permissible.

5) During the conversion the ADC's capacitance must be repeatedly charged or discharged. The internal resistance of the reference source must allow the capacitance to reach their final voltage level within the indicated time. The maximum internal resistance results from the programmed conversion timing.

6) Not 100% tested, but guaranteed by design characterization.

10.4 AC Characteristics (18 MHz)

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

T_A = 0 to 70 °C for the SAB-C515A
T_A = – 40 to 85 °C for the SAF-C515A
T_A = – 40 to 110 °C for the SAH-C515A
T_A = – 40 to 125 °C for the SAK-C515A

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

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 18 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	71	–	2 t_{CLCL} – 40	–	ns
Address setup to ALE	t_{AVLL}	26	–	t_{CLCL} – 30	–	ns
Address hold after ALE	t_{LLAX}	26	–	t_{CLCL} – 30	–	ns
ALE low to valid instruction in	t_{LLIV}	–	122	–	4 t_{CLCL} – 100	ns
ALE to $\overline{\text{PSEN}}$	t_{LLPL}	31	–	t_{CLCL} – 25	–	ns
$\overline{\text{PSEN}}$ pulse width	t_{PLPH}	132	–	3 t_{CLCL} – 35	–	ns
$\overline{\text{PSEN}}$ to valid instruction in	t_{PLIV}	–	92	–	3 t_{CLCL} – 75	ns
Input instruction hold after $\overline{\text{PSEN}}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{\text{PSEN}}$	t_{PXIZ}*)	–	46	–	t_{CLCL} – 10	ns
Address valid after $\overline{\text{PSEN}}$	t_{PXAV}*)	48	–	t_{CLCL} – 8	–	ns
Address to valid instr in	t_{AVIV}	–	180	–	5 t_{CLCL} – 98	ns
Address float to $\overline{\text{PSEN}}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the C515A to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

CLKOUT Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 18 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	349	–	7 t_{CLCL} – 40	–	ns
CLKOUT high time	t_{SHSL}	71	–	2 t_{CLCL} – 40	–	ns
CLKOUT low time	t_{SLSH}	516	–	10 t_{CLCL} – 40	–	ns
CLKOUT low to ALE high	t_{SLLH}	16	96	t_{CLCL} – 40	t_{CLCL} + 40	ns

AC Characteristics (18 MHz, cont'd)

External Data Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		18 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 18 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	233	–	6 t_{CLCL} – 100	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	233	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	81	–	2 t_{CLCL} – 30	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	128	–	5 t_{CLCL} – 150	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	51	–	2 t_{CLCL} – 60	ns
ALE to valid data in	t_{LLDV}	–	294	–	8 t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	335	–	9 t_{CLCL} – 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	117	217	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	92	–	4 t_{CLCL} – 130	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	16	96	t_{CLCL} – 40	t_{CLCL} + 40	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	11	–	t_{CLCL} – 45	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	239	–	7 t_{CLCL} – 150	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	16	–	t_{CLCL} – 40	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive Characteristics

Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 18 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	55.6	285.7	ns
High time	t_{CHCX}	15	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	15	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	15	ns
Fall time	t_{CHCL}	–	15	ns

10.5 AC Characteristics (24 MHz)

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

T_A = 0 to 70 °C for the SAB-C515A
T_A = – 40 to 85 °C for the SAF-C515A
T_A = – 40 to 110 °C for the SAH-C515A

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

Program Memory Characteristics

Parameter	Symbol	Limit Values				Unit
		24 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 24 MHz		
		min.	max.	min.	max.	
ALE pulse width	t_{LHLL}	43	–	2 t_{CLCL} – 40	–	ns
Address setup to ALE	t_{AVLL}	17	–	t_{CLCL} – 25	–	ns
Address hold after ALE	t_{LLAX}	17	–	t_{CLCL} – 25	–	ns
ALE low to valid instruction in	t_{LLIV}	–	80	–	4 t_{CLCL} – 87	ns
ALE to $\overline{PSEN}$	t_{LLPL}	22	–	t_{CLCL} – 20	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	95	–	3t_{CLCL} – 30	–	ns
$\overline{PSEN}$ to valid instruction in	t_{PLIV}	–	60	–	3 t_{CLCL} – 65	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ} *)	–	32	–	t_{CLCL} – 10	ns
Address valid after $\overline{PSEN}$	t_{PXAV} *)	37	–	t_{CLCL} – 5	–	ns
Address to valid instr in	t_{AVIV}	–	148	–	5 t_{CLCL} – 60	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

*) Interfacing the C515A to devices with float times up to 37 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

CLKOUT Characteristics

Parameter	Symbol	Limit Values				Unit
		24 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 24 MHz		
		min.	max.	min.	max.	
ALE to CLKOUT	t_{LLSH}	252	–	7 t_{CLCL} – 40	–	ns
CLKOUT high time	t_{SHSL}	43	–	2 t_{CLCL} – 40	–	ns
CLKOUT low time	t_{SLSH}	377	–	10 t_{CLCL} – 40	–	ns
CLKOUT low to ALE high	t_{SLLH}	2	82	t_{CLCL} – 40	t_{CLCL} + 40	ns

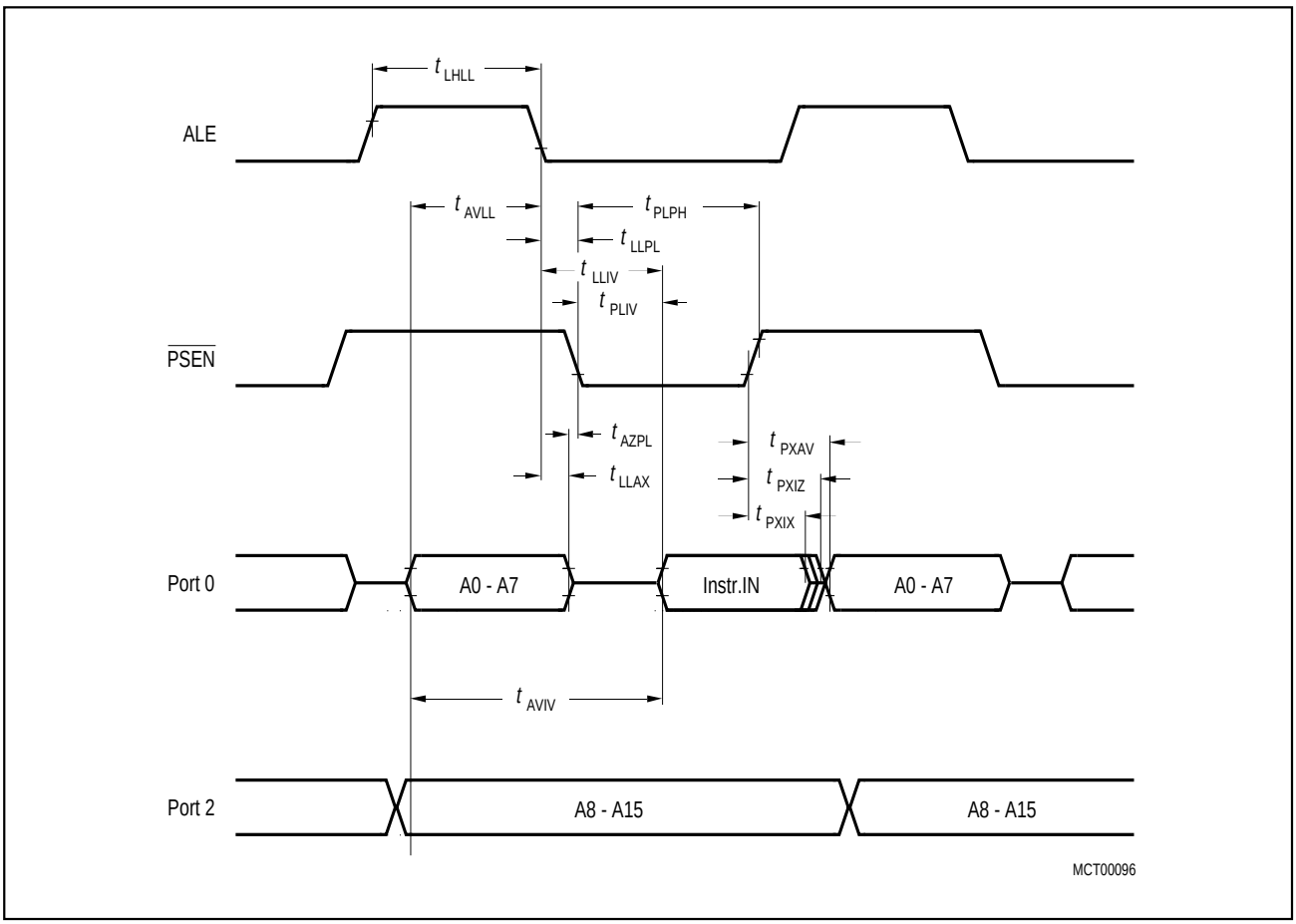
AC Characteristics (24 MHz, cont'd)

External Data Memory Characteristics

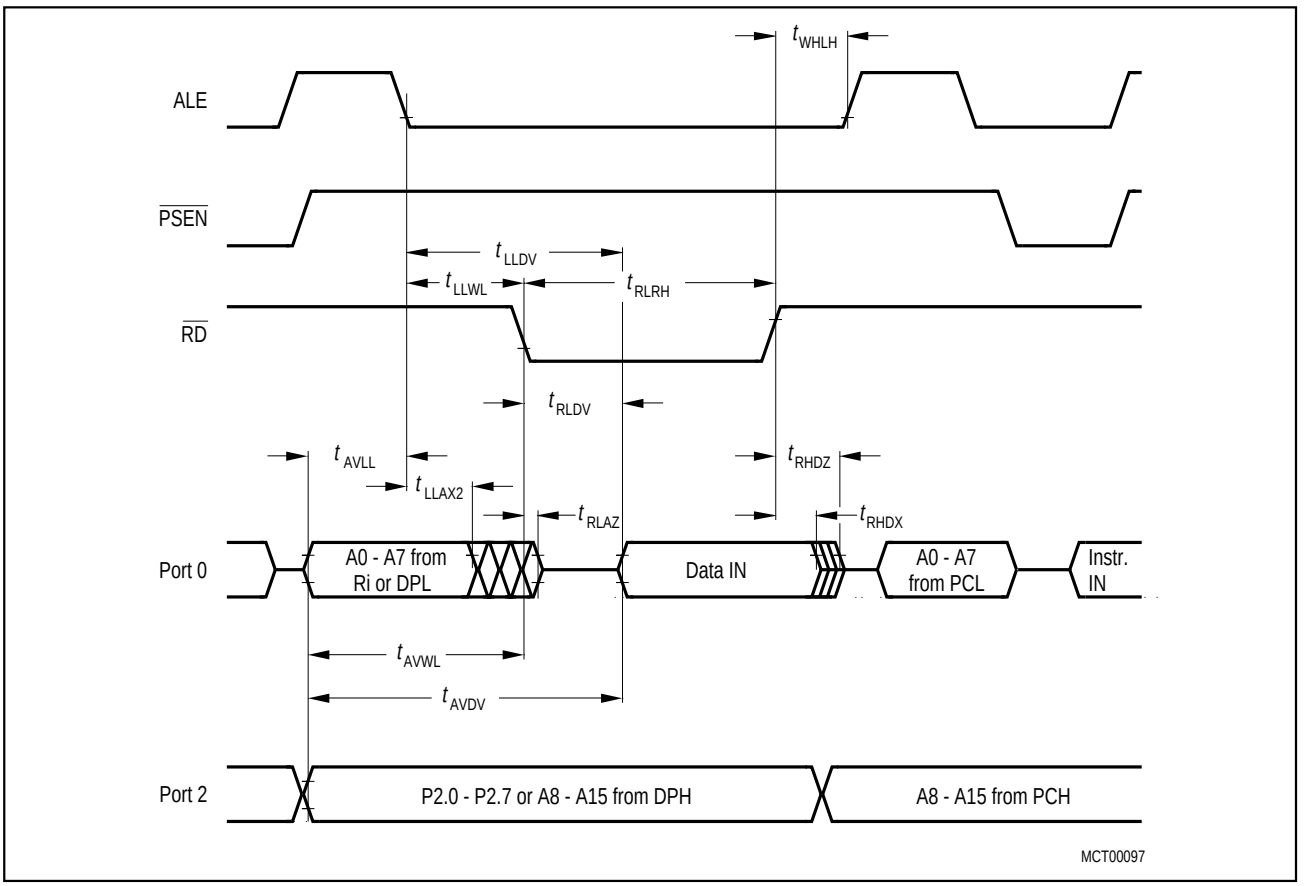
Parameter	Symbol	Limit Values				Unit
		24 MHz Clock		Variable Clock $1/t_{CLCL}$ = 3.5 MHz to 24 MHz		
		min.	max.	min.	max.	
$\overline{RD}$ pulse width	t_{RLRH}	180	–	6 t_{CLCL} – 70	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	180	–	6 t_{CLCL} – 70	–	ns
Address hold after ALE	t_{LLAX2}	56	–	2 t_{CLCL} – 27	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	118	–	5 t_{CLCL} – 90	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	63	–	2 t_{CLCL} – 20	ns
ALE to valid data in	t_{LLDV}	–	200	–	8 t_{CLCL} – 133	ns
Address to valid data in	t_{AVDV}	–	220	–	9 t_{CLCL} – 155	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	75	175	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	67	–	4 t_{CLCL} – 97	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	17	67	t_{CLCL} – 25	t_{CLCL} + 25	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	5	–	t_{CLCL} – 37	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	170	–	7 t_{CLCL} – 122	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	15	–	t_{CLCL} – 27	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

External Clock Drive Characteristics

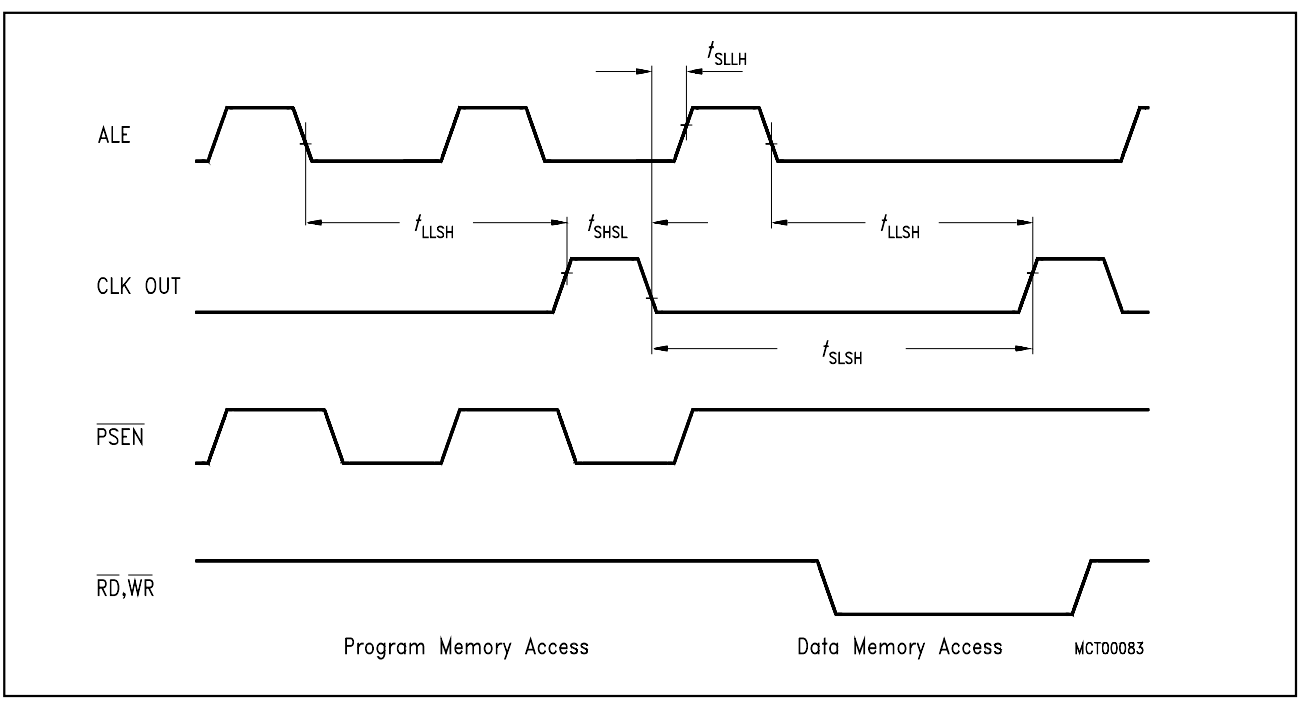
Parameter	Symbol	Limit Values		Unit
		Variable Clock Freq. = 3.5 MHz to 24 MHz		
		min.	max.	
Oscillator period	t_{CLCL}	41.7	285.7	ns
High time	t_{CHCX}	12	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	12	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	12	ns
Fall time	t_{CHCL}	–	12	ns



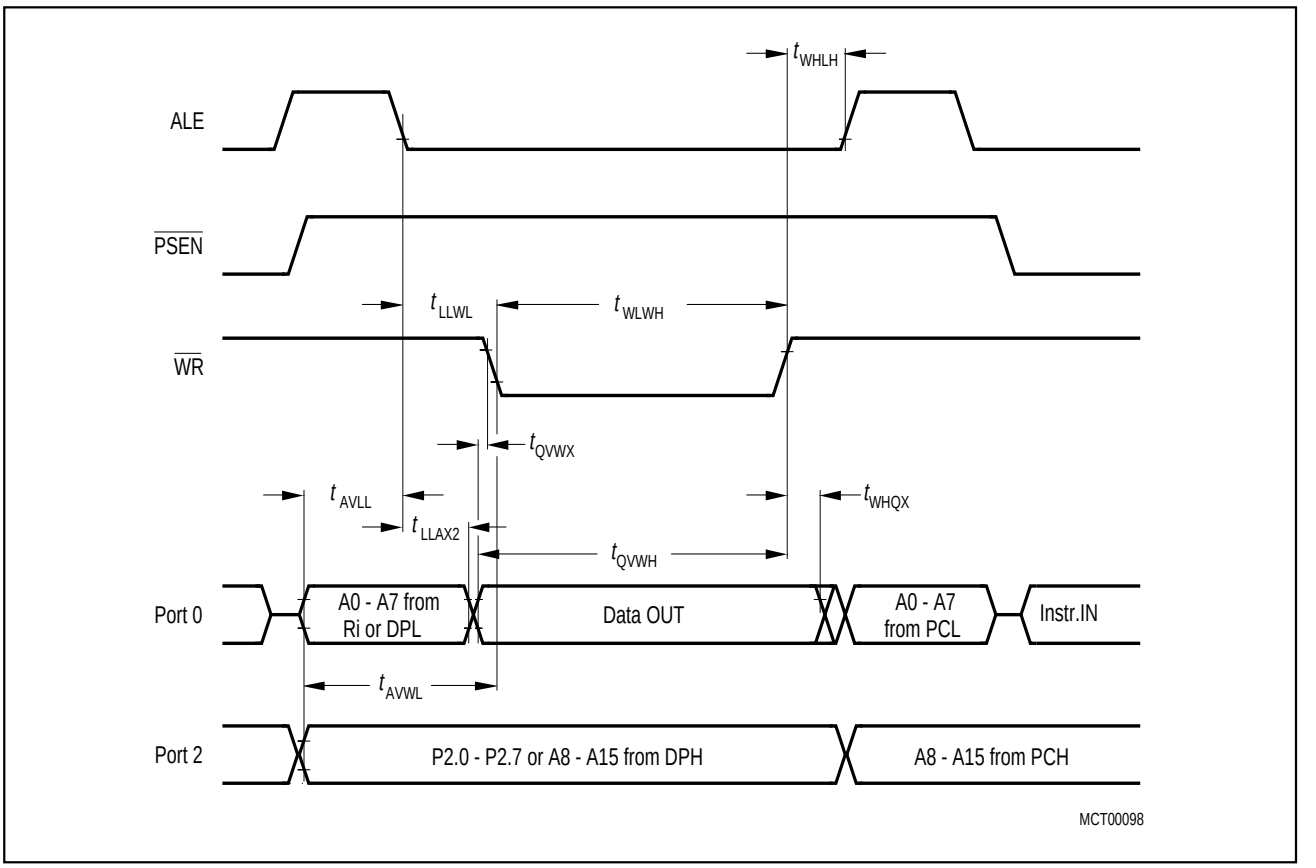
Program Memory Read Cycle



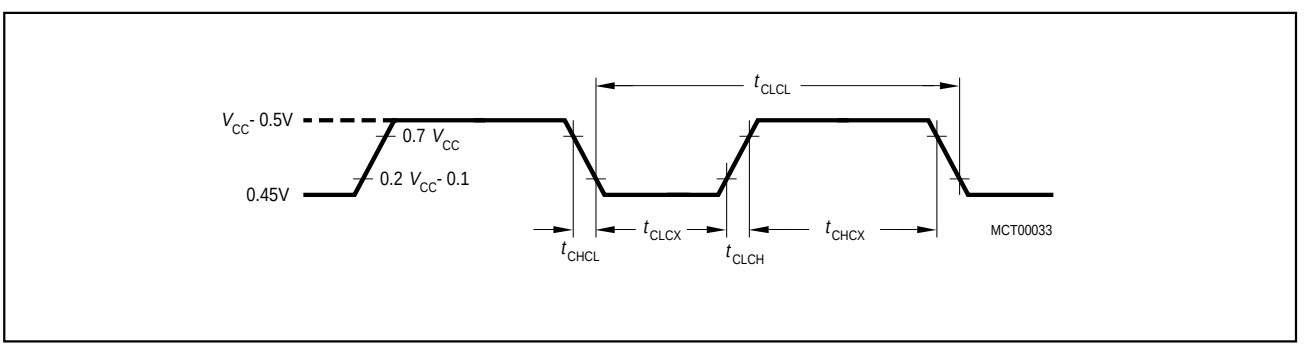
Data Memory Read Cycle



CLKOUT Timing



Data Memory Write Cycle

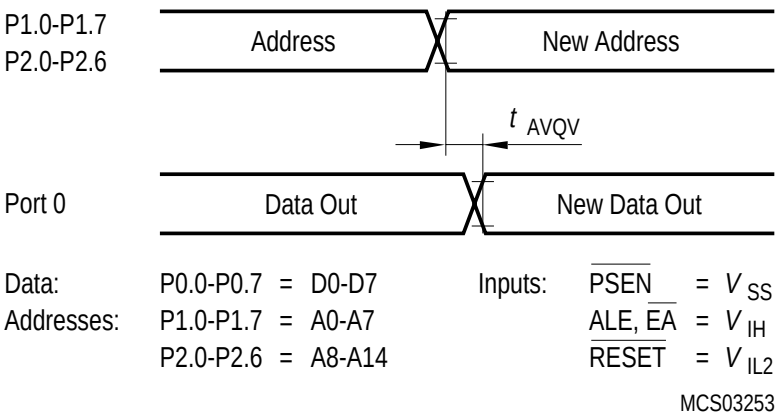


External Clock Drive on XTAL2

10.6 ROM Verification Characteristics for the C515A

ROM Verification Mode 1

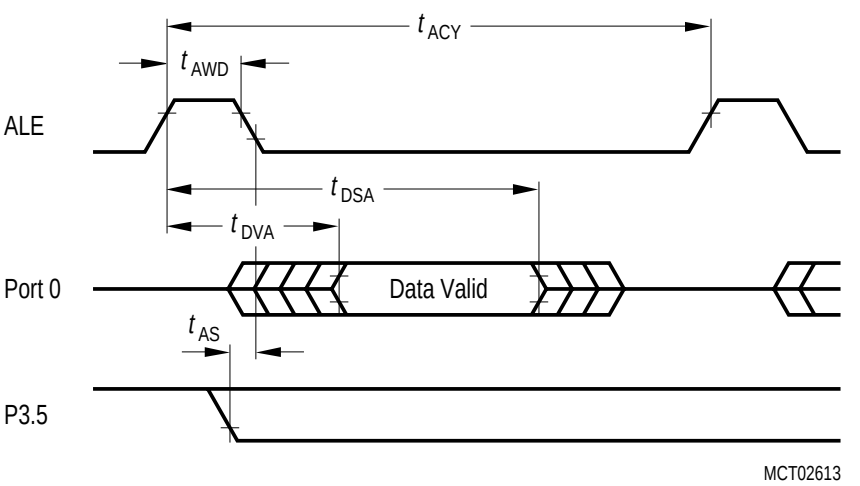
Parameter	Symbol	Limit Values		Unit
		min.	max.	
Address to valid data	t_{AVQV}	–	10 t_{CLCL}	ns



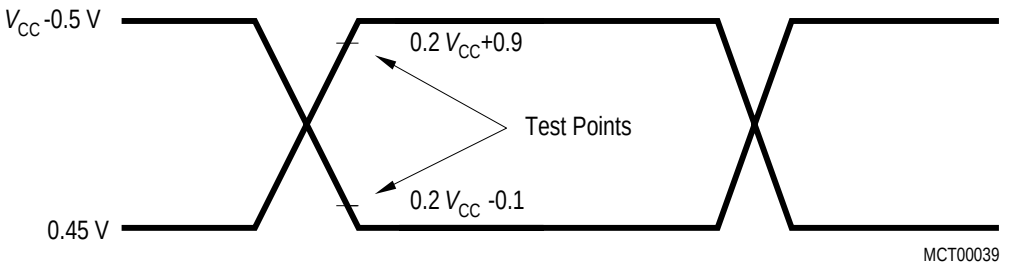
ROM Verification Mode 1

ROM Verification Mode 2

Parameter	Symbol	Limit Values			Unit
		min.	typ	max.	
ALE pulse width	t_{AWD}	–	2 t_{CLCL}	–	ns
ALE period	t_{ACY}	–	12 t_{CLCL}	–	ns
Data valid after ALE	t_{DVA}	–	–	4 t_{CLCL}	ns
Data stable after ALE	t_{DSA}	8 t_{CLCL}	–	–	ns
P3.5 setup to ALE low	t_{AS}	–	t_{CLCL}	–	ns
Oscillator frequency	1/ t_{CLCL}	3.5	–	24	MHz

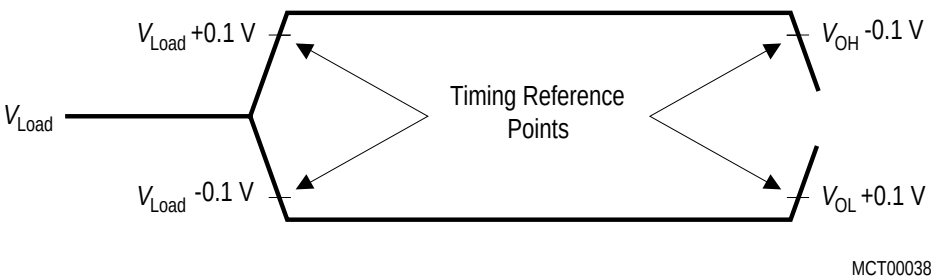


ROM Verification Mode 2



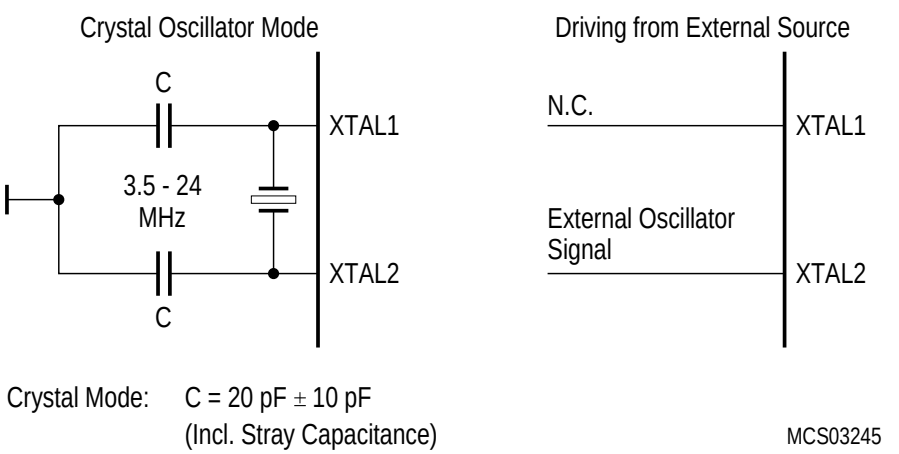
AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at V_{IHmin} for a logic '1' and V_{ILmax} for a logic '0'.

AC Testing: Input, Output Waveforms



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

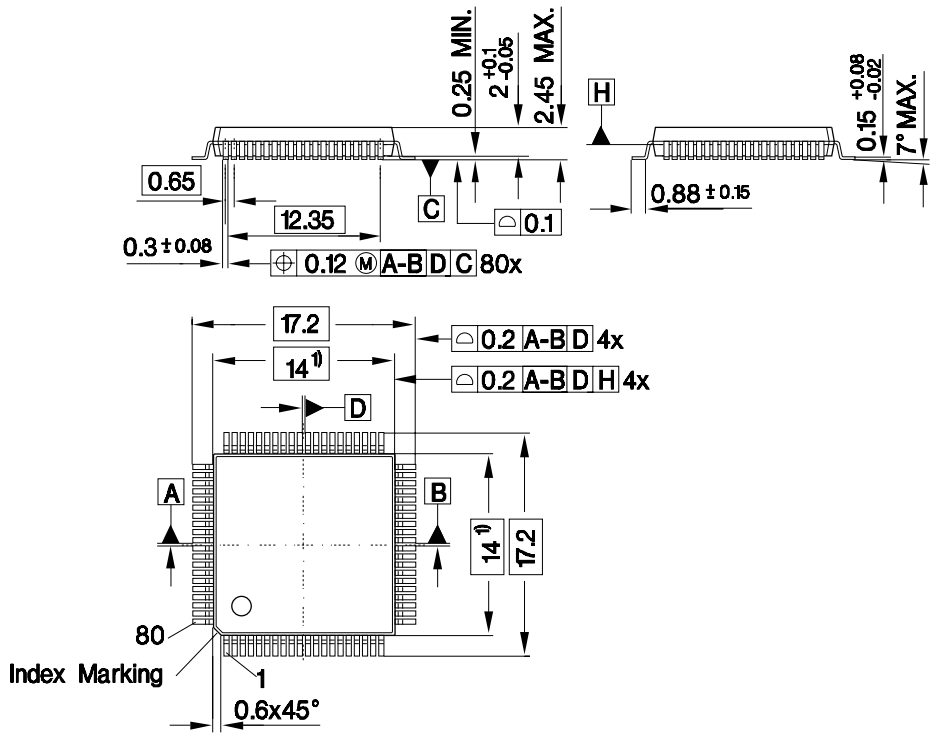
AC Testing : Float Waveforms



Recommended Oscillator Circuits for Crystal Oscillator

10.7 Package Information

Plastic Package, P-MQFP-80-1 (SMD)
(Plastic Metric Quad Flat Package)



P-MQFP-80-1 Package Outline

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

Dimensions in mm

11 Index

Note: Bold page numbers refer to the main definition part of SFRs or SFR bits.

R

S

T