TEMIC

MATRA MHS

C51 Family

Hardware Description of the 80C51, 80C52, 83C154 and 83C154D

1. 80C51, 80C52, 83C154 and 83C154D COMMON FEATURES DESCRIPTION

2. MORE FEATURES FOR MHS-154 PARTS

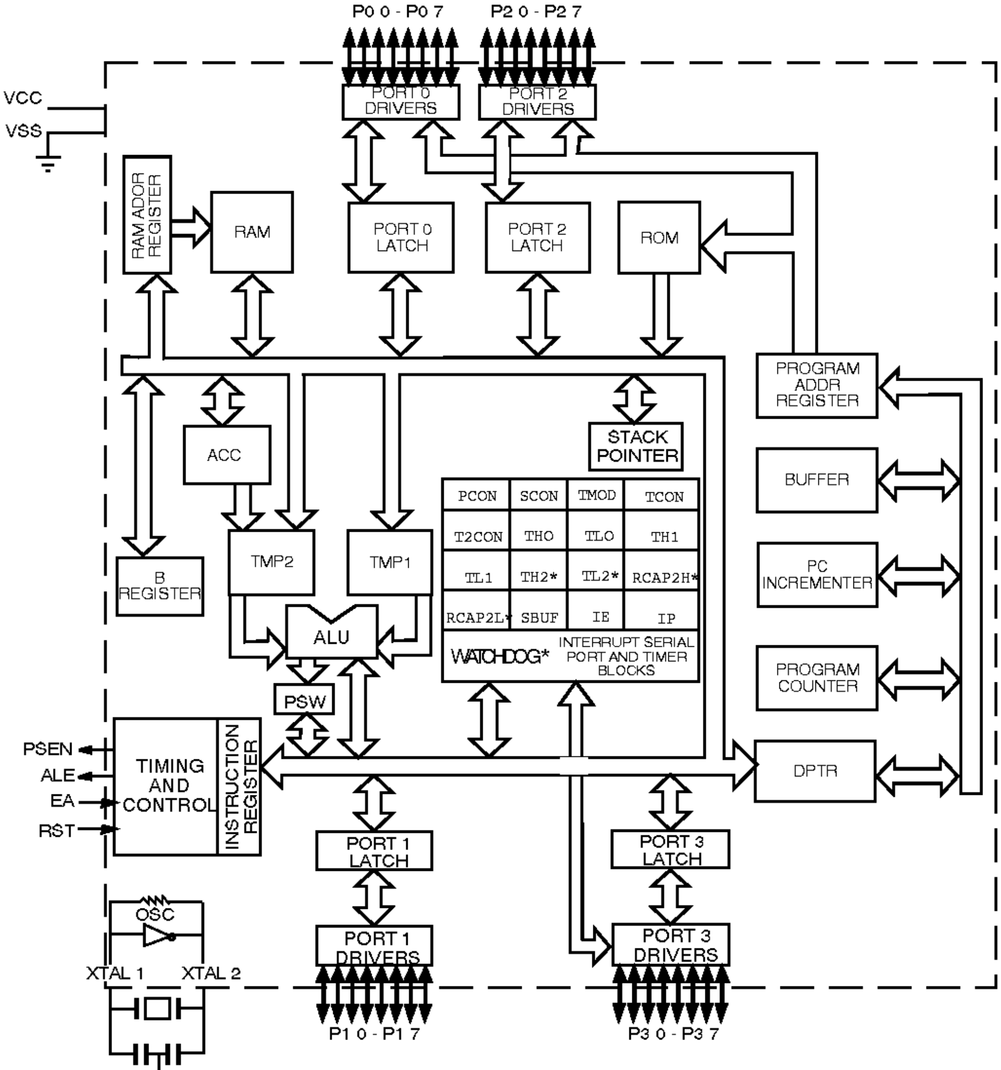
1. 80C51, 80C52, 83C154 And 83C154D Common Features Description

1.1 Introduction

This chapter presents a comprehensive description of the on-chip hardware features of the MHS C51 microcontrollers. Included in this description are :

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations
- The Timer/Counters
- The serial Interface
- The Interrupt System
- Reset
- The reduced Power Modes

Figure 1.1. C51 Architecture Block Diagram.



* 83C154 and 83C154D only.

Table 1 : The MHS C51 Family of Microcontrollers.

DEVICE NAME	ROMLESS VERSION	ROM BYTES	RAM BYTES	16–BIT TIMERS	PROCESS TYPE
80C51	80C31	4K	128	2	CMOS
80C52	80C32	8K	256	3	CMOS
83C154	80C154	16K	256	3*	CMOS
83C154D		32K	256	3*	CMOS

* included watch dog and Timer 32 bits.

The devices under consideration are listed in Table 1. As it becomes unwieldy to be constantly referring to each of these devices by their individual names, we will adopt a convention of refering to them generically as 80C51s, 80C52s and 83C154s, unless a specific member of the group is being refered to, in which case it will be specifically named. The 80C51s include the 80C51 and 80C31. The 80C52s are the 80C52 and 80C32. The 83C154s are the 83C154, the 80C154 and the 83C154D.

Figure 1.1 shows a functionnal block diagram of the 80C51s, 80C52s and 83C154s.

Special Function Registers

A map of the on-chip memory area called SFR (Special Function Register) space is shown in Figure 1.2. SFRs marked by parentheses are resident in the 80C52s and 83C154s but not in the 80C51s. IOCON marked by a star is only resident in the 83C154s.

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

Figure 1.2. SFR Map. (...) Indicates Resident in 80C52s and 83C154s, not in 80C51s.

8 Bytes

F8	*IOCON								FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)		(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

* 83C154s only.

User software should not write 1s to theses unimplemented locations, since they may be used in future MHS C51 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are described as below.

Accumulator

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

B Register

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

Program status word

The PSW register contains program status information as detailed in Figure 1-3.

Stack pointer

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data pointer

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 to 3

P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

Serial data buffer

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

Timer registers

Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit counting registers for Timer/Counters 0, 1, and 2, respectively.

Capture registers

The register pair (RCAP2H, RCAP2L) are the capture register for the Timer 2 “capture mode.” In this mode, in response to a transition at the 80C52’s T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer 2 also has a 16-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode. More about Timer 2’s features in Section 1.6.

Control registers

Special Function Registers IP, IE, TMOD, TCON, T2CON, SCON, and PCON contain control and status bits for the interrupt system, the timer/counters, and the serial port. They are described in later sections.

1.2 Oscillator and Clock Circuit

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter, which can be configured with off-chip components as a Pierce oscillator, as shown in Figure 1.4. The on-chip circuitry, and selection of off-chip components to configure the oscillator are discussed in Section 1.12.

Figure 1.4. Crystal/Ceramic Resonator Oscillator.

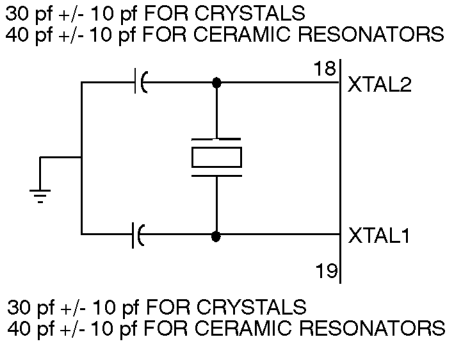

30 pf +/- 10 pf FOR CRYSTALS
40 pf +/- 10 pf FOR CERAMIC RESONATORS
18
XTAL2
XTAL1
19
30 pf +/- 10 pf FOR CRYSTALS
40 pf +/- 10 pf FOR CERAMIC RESONATORS

Figure 1.3. PSW : Program Status Work Register.

(MSB)							(LSB)
CY	AC	F0	RS1	RS0	OV	–	P

Symbol	Position	Name and Significance
CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary Carry flag. (For BCD operations.)
F0	PSW.5	Flag 0 (Available to the user for general purposes.)
RS1	PSW.4	Register bank Select control bits 1 & 0. Set/cleared by software to determine working register bank (see Note).
RS0	PSW.3	

Symbol	Position	Name and Significance
OV	PSW.2	Overflow flag.
–	PSW.1	(reserved)
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate and odd/even number of "one" bits in the accumulator, i.e., even parity.

Note : the contents of (RS1, RS0) enable the working register banks as follows

(0.0)–Bank 0	(00H-07H)
(0.1)–Bank 1	(08H-0FH)
(1.0)–Bank 2	(10H-17H)
(1.1)–Bank 3	(18H-1FH)

The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. The internal clocking signals are at half the oscillator frequency, and define the internal phases, states, and machine cycles, which are described in the next section.

1.3. CPU Timing

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a Phase 1 half, during which the Phase 1 clock is active, and a Phase 2 half, during which the Phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (State 1, Phase 1), through S6P2 (State 6, Phase 2). Each phase lasts for one oscillator period. Each state lasts for two oscillator periods. Typically, arithmetic and logical operations take place during Phase 1 and internal register-to-register transfers take place during Phase 2.

The diagrams in Figure 1-5 show the fetch/execute timing referenced to the internal states and phases. Since these internal clock signals are not user accessible, the XTAL2 oscillator signal and the ALE (Address Latch Enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle : once during S1P2 and S2P1, and again during S4P2 and S5P1.

Execution of one-cycle instruction begins at S1P2, when the opcode is latched into the Instruction Register. If it is a two-byte instruction, the second byte is read during S4 of the same machine cycle. If it is one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next opcode), is ignored, and the Program Counter is not incremented. In any case, execution is complete at the end of S6P2. Figures 1-5A and 1-5B show the timing for a 1-byte, 1-cycle instruction and for a 2-byte, 1-cycle instruction.

Most 80C51 instructions execute in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete. They take four cycles.

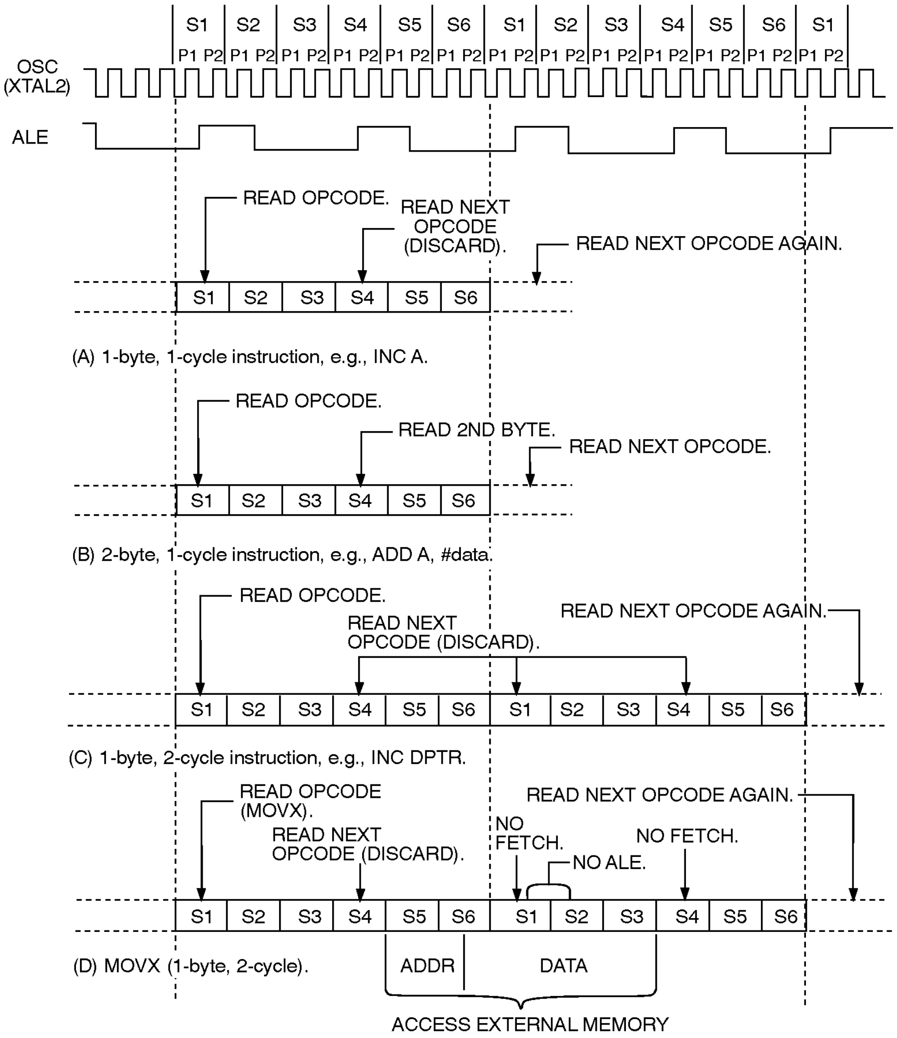
Normally, two codes bytes are fetched from Program Memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a 1-byte 2-cycle instruction that accesses external Data Memory. During a MOVX, two fetches are skipped while the external Data Memory is being addressed and strobed. Figure 1-5C and 1-5D show the timing for a normal 1-byte, 2-cycle instruction and for a MOVX instruction.

1.4 Port Structures and Operation

All four ports in the 80C51 are bidirectional. Each consists of a latch (Special Function Register P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

Figure 1.5. 80C51 fetch/Execute Sequences.



(A) 1-byte, 1-cycle instruction, e.g., INC A.

(B) 2-byte, 1-cycle instruction, e.g., ADD A, #data.

(C) 1-byte, 2-cycle instruction, e.g., INC DPTR.

(D) MOVX (1-byte, 2-cycle).

All the Port 3 pins, and (in the 80C52) two Port 1 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed below :

Port Pin	Alternate Function
*P1.0	T2 (Timer/Counter 2 external input)
*P1.1	T2EX (Timer/Counter 2 capture/reload trigger)
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt)
P3.3	$\overline{\text{INT1}}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{\text{WR}}$ (external Data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external Data memory read strobe)

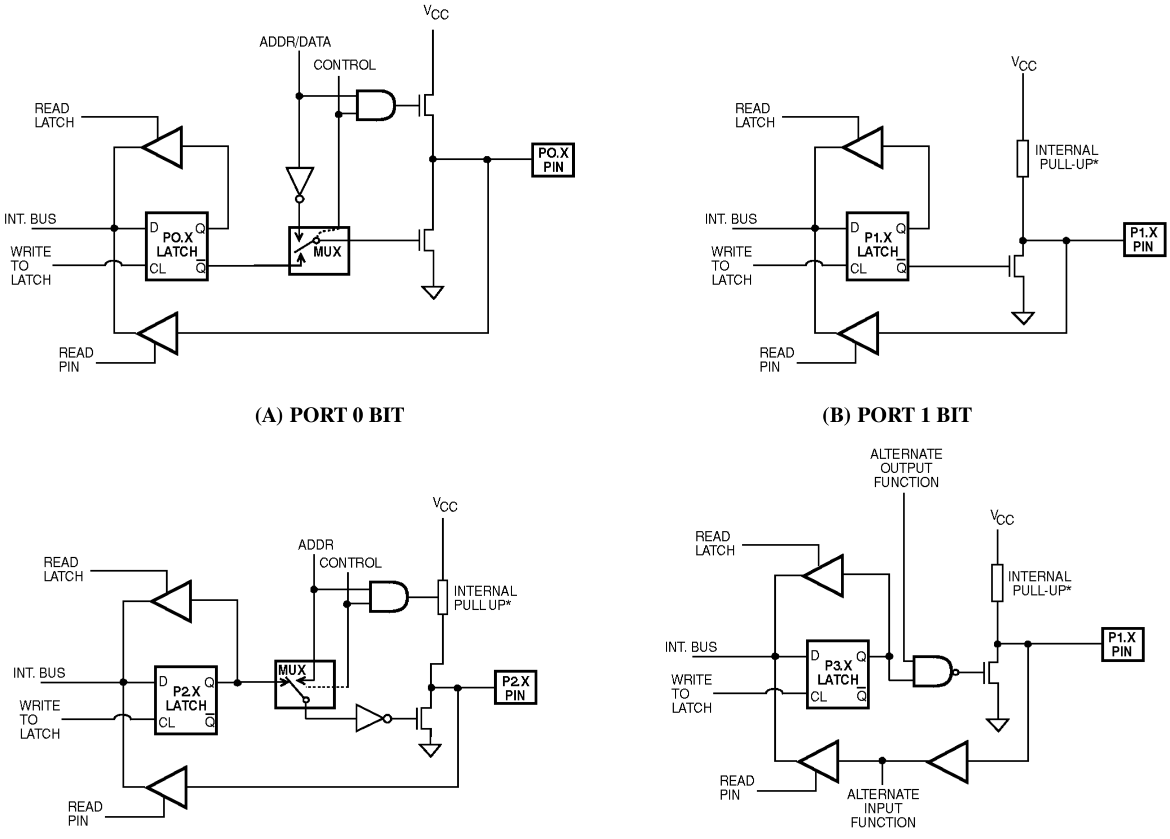
* P1.0 and P1.1 serve these alternate functions only on the 80C52, 83C154 and 83C154D.

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0.

I/O Configurations

Figure 1-6 shows a functional diagram of a typical bit latch and I/O buffer in each of the four 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 a port activate the "read latch" signal, and others activate the "read latch" signal, and others activate the "read pin" signal.

Figure 1.6. 80C51 Port Bit Latches and I/O Buffers.



(A) PORT 0 BIT

(B) PORT 1 BIT

(C) PORT 2 BIT

(D) PORT 3 BIT

* See Figure 1.7 for details of the internal pullup.

As shown in Figure 1-6, the output drivers of Ports 0 and 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Also shown in Figure 1-6, is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled “alternate output function.” The actual P3.X pin level is always available to the pin’s alternate input function, if any.

Ports 1, 2, and 3 have internal pull-ups. Ports 0 has open-drain outputs. Each I/O line can be independently used as an input or an output. (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pull-up, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 1-6A) is used only when the Port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that conditions it can be used as a high-impedance input.

Because Ports 1, 2, and 3 have fixed internal pullups they are sometimes called “quasi-bidirectional” ports. When configured as inputs they pull high and will source current (IIL, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered “true” bidirectional, because when configured as an input it floats.

All the port latches in the 80C51 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

Writing to a Port

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won’t actually appear at the output pin until the next Phase 1, which be at S1P1 of the next machine cycle.

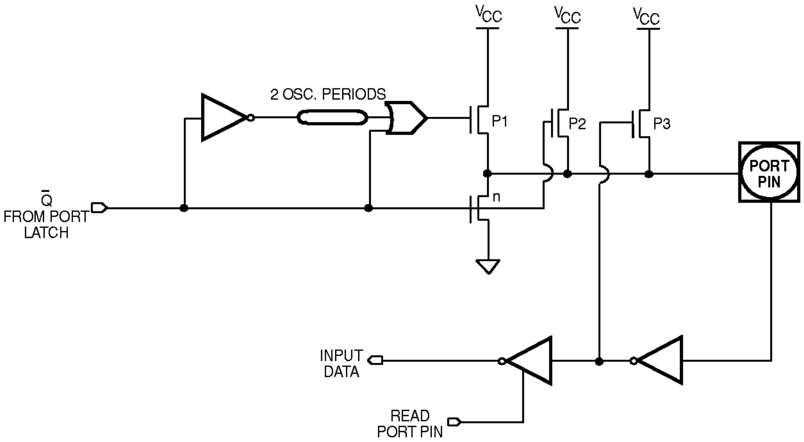
If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pull-up is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pull-up can source about 100 times the current that the normal pull-up can. It should be noted that the internal pull-ups are field-effect transistors, not linear resistors. The pull-up arrangements are shown in Figure 1-7.

In the CMOS versions, the pull-up consists of three pFETs. It should be noted that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite : it is on when its gate sees a 0, and off when its gate sees a 1.

pFET 1 in Figure 1-7 is the transistor that is turned on 2 oscillator periods after a 0-to-1 transition in the port latch. While it’s on, it turns on pFET 3 (a weak pull-up), through the inverter. This inverter and pFET form a latch which hold the 1.

Note that if the pin is emitting a 1, a negative glitch on the pin from some external source can turn off pFET 3, causing the pin to go into a float state, pFET 2 is a very weak pull-up which is on whenever the nFET is off, in traditional CMOS style. It’s only about 1/10 the strenght of pFET3. Its function is to restore a 1 to the pin in the event the pin *had* a 1 and lost it to a glitch.

Figure 1.7. Ports 1 and 3 CMOS Internal Pull-up Configurations. Port 2 is similar except that it holds the strong pullup on while emitting 1s that are address bits. (See test, “Accessing External Memory”.)

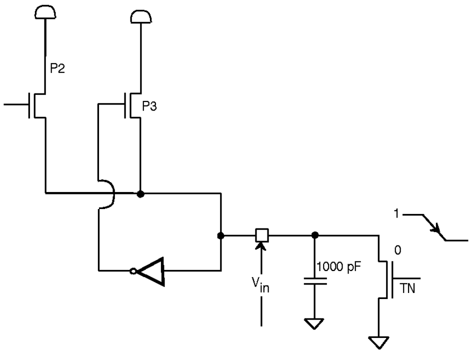


CMOS Configuration. pFET 1 is turned on for 2 osc. periods after Q makes a 1-to-0 transition. During this time, pFET 1 also turns on pFET 3 through the inverter to form a latch which holds the 1. pFET 2 is also on.

Port loading and interfacing

The output buffer of Ports 1, 2 and 3 can each drive 3LS TTL inputs. The pins can be driven by open-collector and open-drain outputs, but note that 0-to-1 transition will not be fast. In the CMOS device, an input 0 turns off pullup P3, leaving only the weak pullup P2 to drive the transistor. The figure 1-8 shows an example where the port is driven by an open drain transistor t_N. The parasitic capacitance is equal to 100pF.

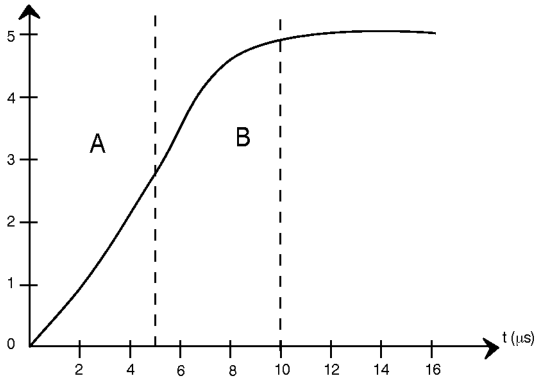
Figure 1.8. Port Interfacing.



The figure 1-9 shows the behaviour of the port during 0-to-1 transition.

In the area A only pullup P2 sinks the capacitor and takes 5 µs to switch from 0 volt to 2 volts. In the area B, pullup P2 and P3 feed the capacitor and the time to charge the capacitor is divide roughly by ten. So this figure shows it takes some machine cycles before having a true high level during a 0-to-1 transition.

Figure 1.9. Port Behaviour During 0-to-1 Transition.



Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which ? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called “read-modify-write” instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin :

ANL	(logical AND, e.G., ANL P1,A)
ORL	(logical OR, e.g., ORL P2,A)
XRL	(logical EX-OR, e.g., XRL P3,A)
JBC	(jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)
CPL	(complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(decrement, e.g., DEC P2)
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)

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that 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 transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

Further details are given in the next chapter concerning the powerful functions of the 83C154 I/O PORTS.

1.5 Accessing External Memory

Accesses to external memory are of two types : accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{\text{PSEN}}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (alternate function of P3.7 and P3.6) to strobe the memory.

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).

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. Note that the Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This is during the execution of a MOVX @DPTR instruction. During this time the Port 2 latch (the Special Function register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being 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.

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signal drives both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs, and do not require external pull-ups. Signal ALE (address latch enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transitions of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{\text{WR}}$ is activated, and remains there until after $\overline{\text{WR}}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is desactivated.

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

External program Memory is accessed under two conditions :

1) Whenever signal $\overline{\text{EA}}$ is active ; or
2) Whenever the program counter (PC) contains a number that is larger than 0FFFH (1FFFH for the 80C52, 3FFFH for the 83C154 and 7FFFH for the 83C154D.

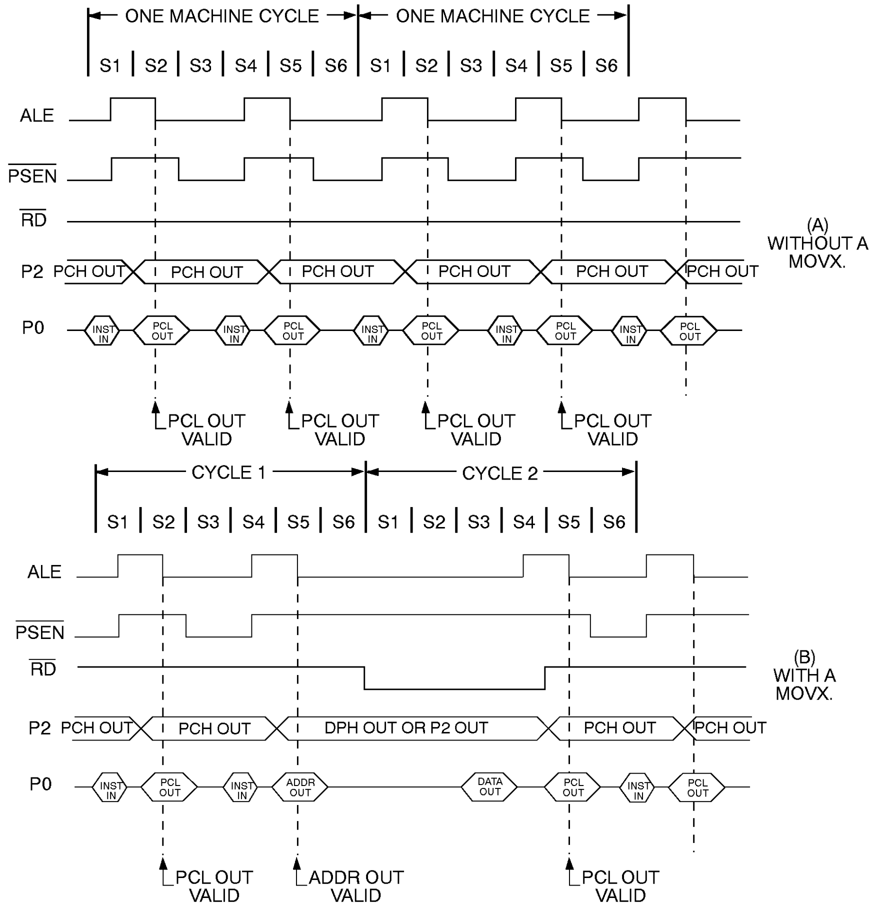
This requires that the ROMless versions have $\overline{\text{EA}}$ wired low to enable the lower 4K (8K for the 80C32, 16K for the 80C154 and 32K for the 80C154D) program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

$\overline{\text{PSEN}}$

The read strobe for external fetches is $\overline{\text{PSEN}}$. $\overline{\text{PSEN}}$ is not activated for internal fetches. When the CPU is accessing external Program Memory, $\overline{\text{PSEN}}$ is activated twice every cycle (except during a MOVX instruction) whether or not the byte fetched is actually needed for the current instruction. When $\overline{\text{PSEN}}$ is activated its timing is not the same as $\overline{\text{RD}}$. A complete $\overline{\text{RD}}$ cycle, including activation and deactivation of ALE and $\overline{\text{RD}}$, takes 12 oscillator periods. A complete $\overline{\text{PSEN}}$ cycle, including activation and deactivation of ALE and $\overline{\text{PSEN}}$, takes 6 oscillator periods. The execution sequence for these two types of read cycles are shown in Figure 1-10 for comparison.

Figure 1.10. External Program Memory Execution.



ALE

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 to an external latch during fetches from external Program Memory. For that purpose ALE is activated twice every machine cycle. This activation takes place even when the cycle involves no external fetch. The only time an ALE pulse doesn't come out is during an access to external Data Memory. The first ALE of the second cycle of a MOVX instructions is missing (see Figure 1-10). Consequently, in any system that does not use external Data Memory, ALE is activated at a constant rate of 1/6 the oscillator frequency, and can be used for external clocking or timing purposes.

Overlapping External Program and Data Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is being used to store data. In the 80C51, the external Program and Data Memory spaces can be combined by ANDing $\overline{PSEN}$ and $\overline{RD}$. A positive-logic AND of these two signals produces an active-low read strobe that can be used for the combined physical memory. Since the $\overline{PSEN}$ cycle is faster than the $\overline{RD}$ cycle, the external memory needs to be fast enough to accomodate the $\overline{PSEN}$ cycle.

1.6 Timer/Counters

The 80C51 has two 16-bit timer/counter registers : Timer 0 and Time 1. The 80C52, 83C154 and 83C154D have these two plus one more : Timer 2. All three can be configured to operate either as timers or event counters.

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

In the "counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0, T1 or (in the 80C52, 83C154 and 83C154D) T2. 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 2 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 should be held for at least one full machine cycle.

In addition to the "timer" or "counter" selection, Timer 0 and Timer 1 have four operating modes from which to select. Timer 2, in the 80C52, 83C154 and 83C154D has three modes of operation : "capture," "auto-reload" and "baud rate generator."

Figure 1.11. TMOD : Timer/Counter Mode Control Register.

(MSB)							(LSB)
GATE	C/$\overline{T}$	M1	M0	GATE	C/T	M1	M0
TIMER 1				TIMER 0			

GATE Gating control When set. Timer/counter "x" is enabled only while "$\overline{INTx}$" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.

C/T Timer or Counter Selector Cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from "Tx" input pin).

M1	M0	Operatong Mode	
0	0	MCS-48 Timer "TLx" serves as five-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 timer it overflows.	
1	1	(Timer 0)	TL0 is an eight bit timer counter-controlled by the standard Timer 0 control bits TH0 is an eight-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1)	Timer-counter 1 stopped.

Timer 0 and Timer 1

These timer/counter are present in both the 80C51, the 80C52, the 83C154 and the 83C154D. The “timer” or “counter” function is selected by control bits $C/\overline{T}$ in the Special Function Register TMOD (Figure 1-11). These two timer/counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both timer/counters. Modes 3 is different. The four operating modes are described below.

Mode 0

Putting either Timer into mode 0 makes it look like an 8048 Timer, which is an 8-bit counter with a divide-by-32 prescaler. Figure 1-12 shows the mode 0 operation as it applies to Timer 1.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{\text{INT1}}$ = 1. (Setting GATE =1 allows the Timer to be controlled by external input $\overline{\text{INT1}}$, to facilitate pulse width measurements). TR1 is a control bit in the Special Function register TCON (Figure 1-10). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ingored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and $\overline{\text{INT0}}$ for the corresponding Timer 1 signals in Figure 1-12. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL1) with automatic reload, as shown in Figure 1-14. Overflow from TL1 not only sets TF1, which is preset by sofware. The reload leaves TH1 unchanged.

Mode 2 operation is the same for Timer/Counter 0.

Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Figure 1.12. Timer/Counter 1 Mode 0 : 13-bit Counter.

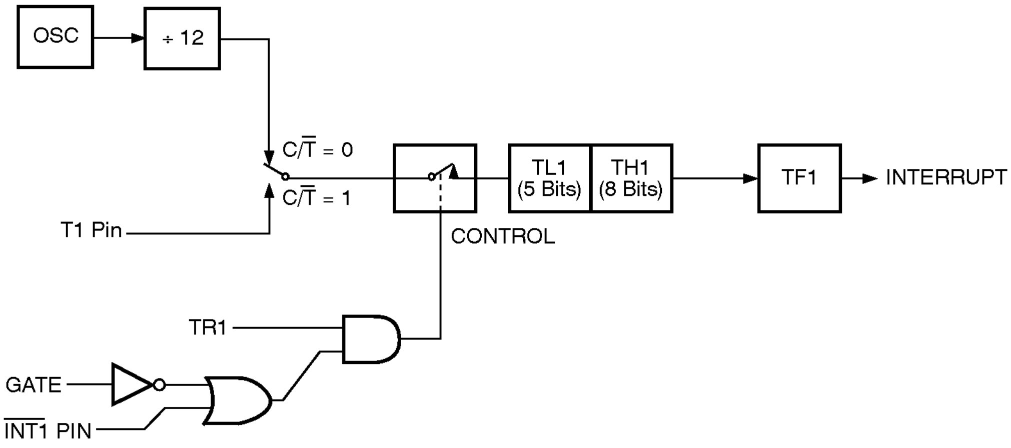


Figure 1.13. TCON : Timer/Counter Control Register.

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.
IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 1-15. 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. With Timer 0 in Mode 3, an 80C51 can look like it has three timer/counters, and an 80C52, an 83C154 and 83C154D, like it has four. 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 port as a baud rate generator, or in fact, in any application not requiring an interrupt.

Timer 2

Timer 2 is a 16-bit timer/counter which is present only in the 80C52, 83C154 and 83C154D. Like Timers 0 and 1, it can operate either as a timer or as an event counter.

Figure 1.14. Timer/Counter 1 Mode 2 : 8-bit Auto-reload.

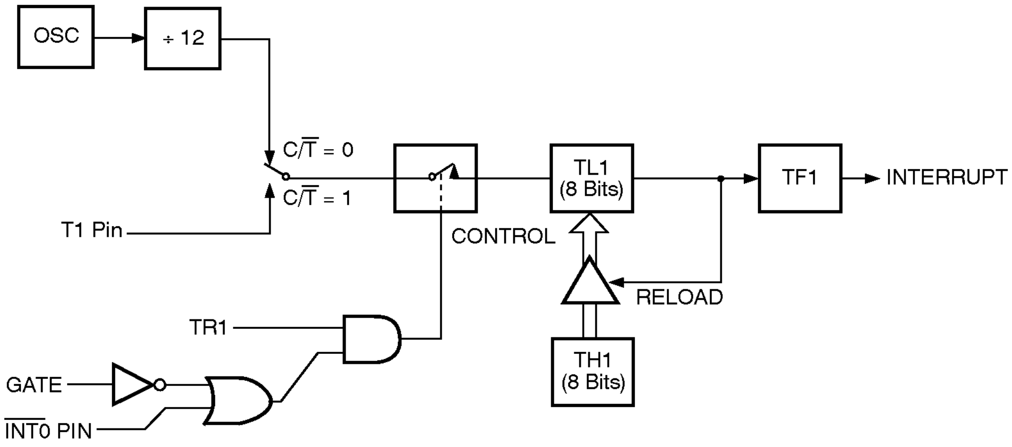


Figure 1.15. Timer/Counter 0 Mode 3 : Two 8-bit Counters.

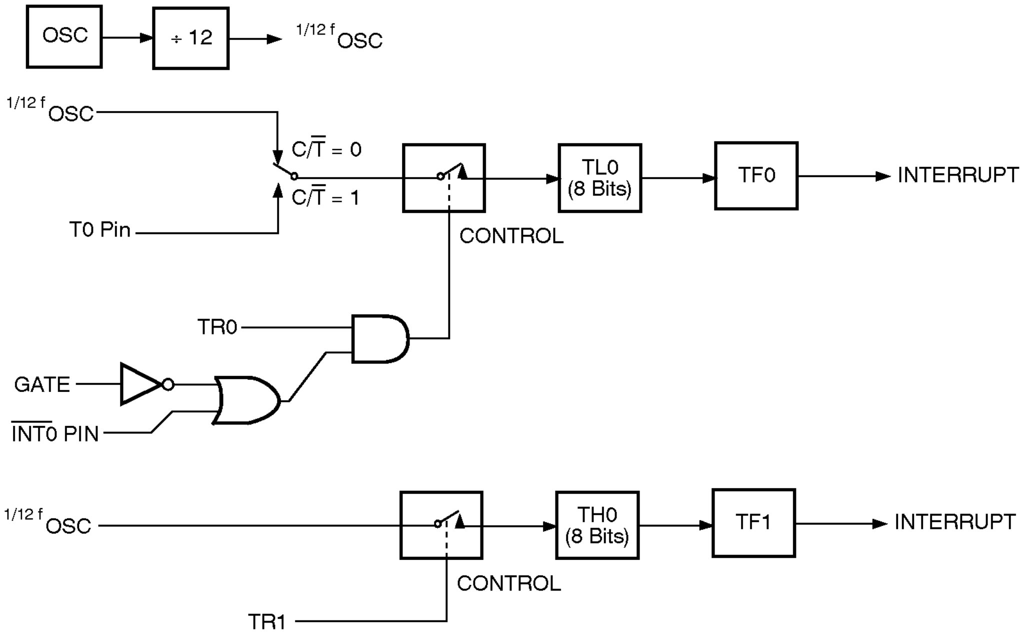


Figure 1.16. T2CON : Timer/Counter 2 Control Register.

(MSB) (LSB)

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{\text{T2}}$	CP/$\overline{\text{RL2}}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{\text{T2}}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{\text{RL2}}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto–reload on Timer 2 overflow.

This is selected by bit C/$\overline{\text{T2}}$ in the Special Function Register T2CON (Figure 1-16). It has three operating modes : “capture,” “autoload” and “baud rate generator,” which are selected by bits in T2CON as shown in Table 2.

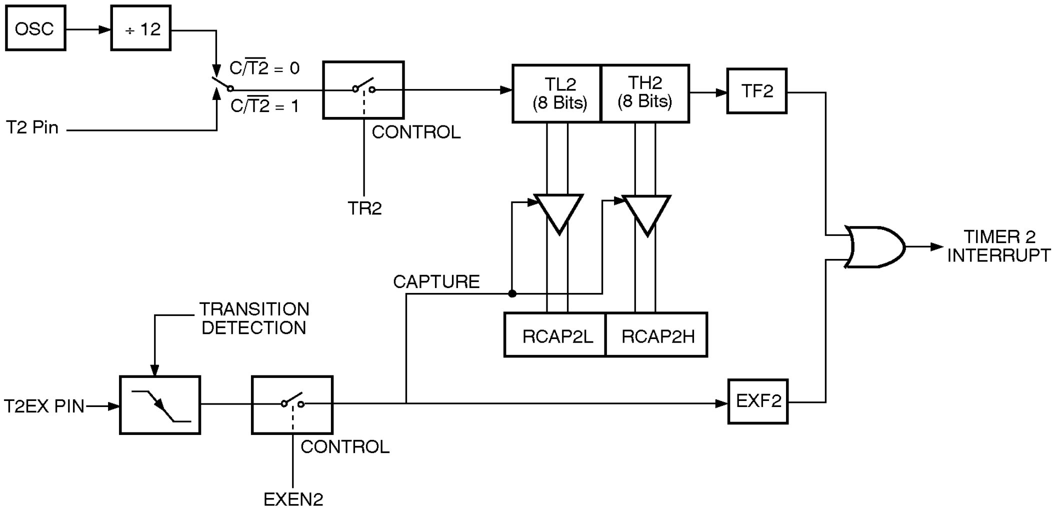
Table 2 : Timer 2 Operating Modes.

RCLK + TCLK	CP/$\overline{\text{RL2}}$	TR2	MODE
0	0	1	16–bit auto–reload
0	1	1	16–bit capture
1	X	1	baud rate generator
X	X	0	(off)

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the 80C52, 83C154 and 83C154D. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.

The capture mode is illustrated in Figure 1-17.

Figure 1.17. Timer 2 in Capture Mode.



In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

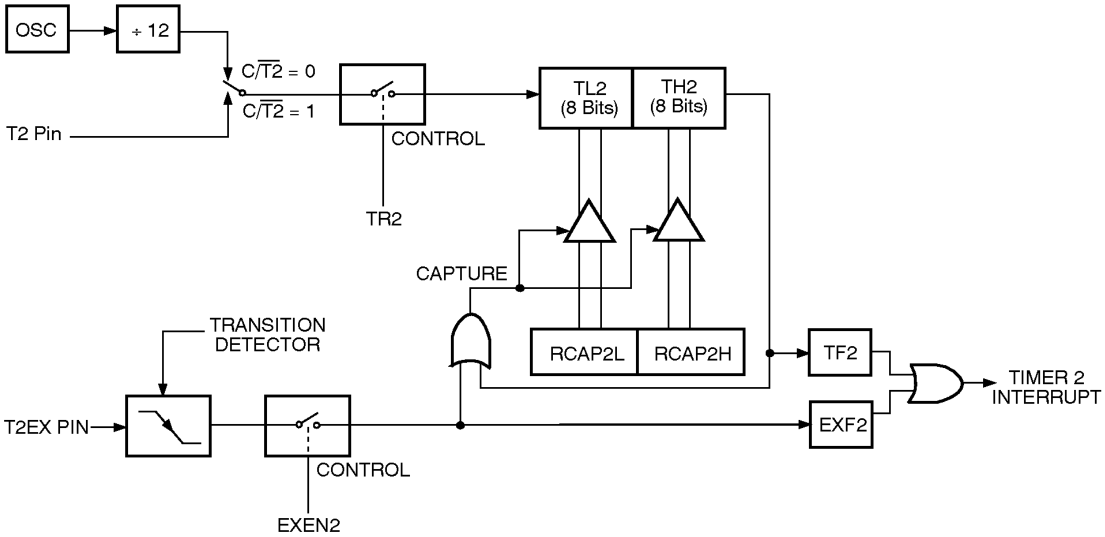
The auto-reload mode is illustrated in Figure 1-18.

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

1.7 Serial Interface (80C51 and 80C52 only)

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previoulsy 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.

Figure 1.18. Timer 2 in Auto-Reload Mode.



The serial port can operate in 4 modes :

Mode 0 : Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received : 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1 : 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.

Mode 2 : 11 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3 : 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.

In all four modes, transmission is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in Mde 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

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

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

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

Serial port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 1-19. 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 interrupts bits (TI and RI).

Figure 1.19. SCON : Serial Port Control Register.

(MSB) (LSB)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

where SM0, SM1 specify the serial port mode, as follows :

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	f_{OSC}./12
0	1	1	8 bit UART	variable
1	0	2	9 bit UART	f_{OSC}./64 or f_{OSC}./32
1	1	3	9 bit UART	variable

• SM2 enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be received. In mode 0, SM2 should be 0.

• REN enables serial reception. Set by software to enable reception. Clear by software to disable reception.

• TB8 is the 9th data bit that will be transmitted in modes 2 and 3. Set or clear by software as desired.

• RB8 in modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.

• TI is transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.

• RI is receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Baud Rates

The baud rate in Mode 0 is fixed :

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is its value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

$$\text{Mode 2 Baud Rate} = \frac{2^{\text{SMOD}}}{64} \times (\text{Oscillator Frequency})$$

In the 80C51, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate. In the 80C52, 83C154 and 83C154D, these baud rates can be determined by Timer 1, or by Timer 2, or by both (one for transmit and the other for receive).

Using Timer 1 to Generate Baud Rates

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

$$\begin{matrix}\text{Modes 1, 3}\\ \text{Baud rate}\end{matrix} = \frac{2^{\text{SMOD}}}{32} \times (\text{Timer 1 Overflow rate})$$

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

$$\begin{matrix}\text{Modes 1, 3}\\ \text{Baud rate}\end{matrix} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (\text{TH1})]}$$

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

Figure 1-20 lists various commonly used baud rates and how they can be obtained from Timer 1.

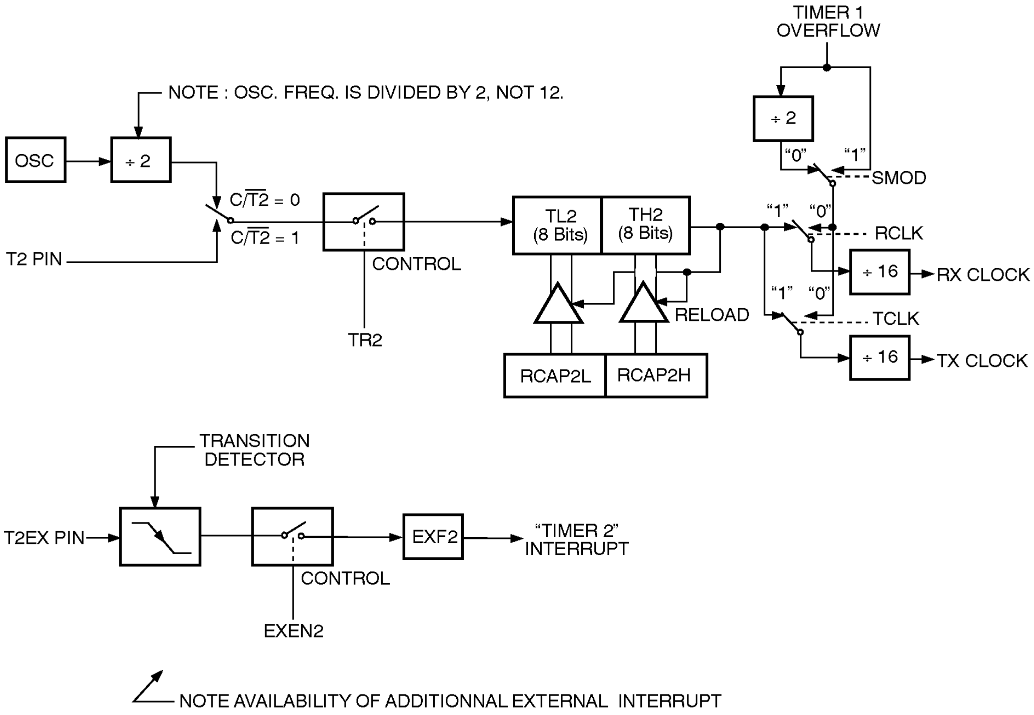
Figure 1.20. Timer 1 Generated Commonly Used Baud Rates.

Baud Rate	f_{OSC}	SMOD	TIMER 1		
			C/$\overline{T}$	Mode	Reload Value
MODE 0 MAX : 1MHZ	12 MHZ	x	x	x	x
MODE 2 MAX : 375K	12 MHZ	1	x	x	x
MODES 1,3 : 62.5K	12 MHZ	1	0	2	FFH
19.2K	11.059 MHZ	1	0	2	FDH
9.6K	11.059 MHZ	0	0	2	FDH
4.8K	11.059 MHZ	0	0	2	FAH
2.4K	11.059 MHZ	0	0	2	F4H
1.2K	11.059 MHZ	0	0	2	E8H
137.5	11.986 MHZ	0	0	2	1DH
110	6 MHZ	0	0	2	72H
110	12 MHZ	0	0	1	FEEBH

Using Timer 2 to Generate Baud Rates

In the 80C52, 83C154 and 83C154D, Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Figure 1-16). Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 1-21.

Figure 1.21. Timer 2 in Baud Rate Generator Mode.



The baud rate generator mode is similar to the auto-reload mode, in that a rollower in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registes RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows :

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow rate}}{16}$$

The Timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation ($C/\overline{T2}$ = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increment every state time (thus at 1/2 the oscillator frequency). In that case the baud rate is given by the formula

$$\begin{array}{c}\text{Modes 1, 3}\\ \text{Baud rate}\end{array} = \frac{\text{Oscillator Frequency}}{32 \times [65536\text{–(RCAP2H, RCAP2L)}]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 1-21. This Figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

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

TR2) before accessing the Timer 2 or RCAP registers, in this case.

More About Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received : 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Figure 1-22 shows a simplified functional diagram of the serial port in mode 0, and associated timing.

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 bit position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between “write to SBUF”, and activation of SEND.

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

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. 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 11111110 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 cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift 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.

More 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. In the 80C51 the baud rate is determined by the Timer 1 overflow rate. In the 80C52, 83C154 and 83C154D it is determined either by the Timer 1 overflow rate, or the Timer 2 overflow rate, or both (one for transmit and the other for receive).

Figure 1-23 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit and receive.

Transmission is initiated by any 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 actually commences at S1P1 of the machine cycle following the next rollower 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 fisrt shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros 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 rollower 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 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

Figure 1.22. Serial Port 0.

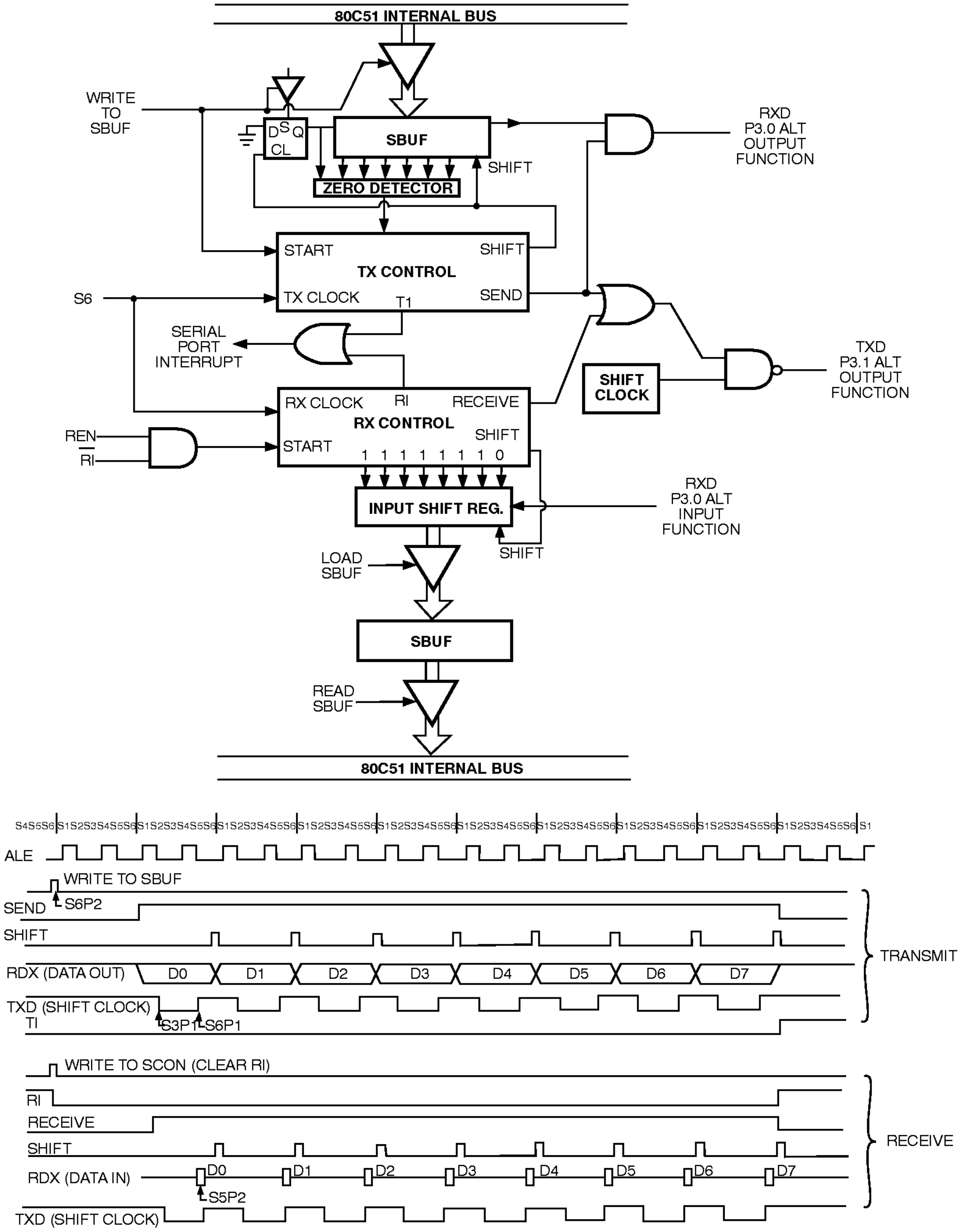
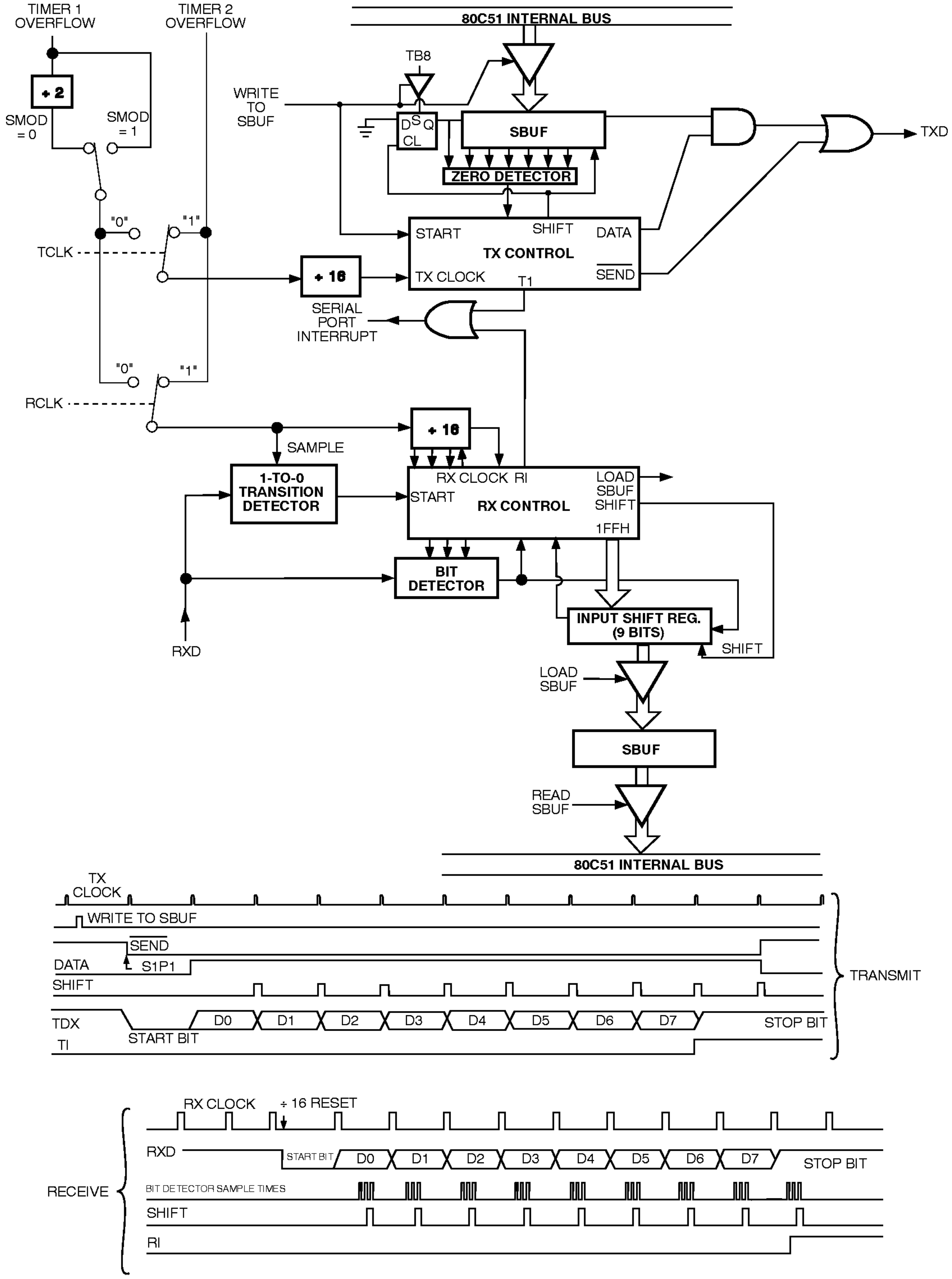


Figure 1.23. Serial Port Mode 1. TCLK, RCLK, and Timer 2 are present in the 80C32/80C52, 80C154/83C154 and 83C154D.



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

As data bits come in from the right, 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 SBUFF and RB8, and set RI. Will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

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

If either of these two confitions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go 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.

More 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 goes into RB8 is SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in mode 2. Mode 3 may have a variable baud rate generated from either Timer 1 or 2 depending on the state of TCLK and RCLK.

Figures 1-24 A and B show 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.

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 commences at S1P1 of the machine cycle following 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 $\overline{\text{SEND}}$ 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 is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

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

As data bits come 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 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 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 bits go 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 1.24A. Serial Port Mode 2.

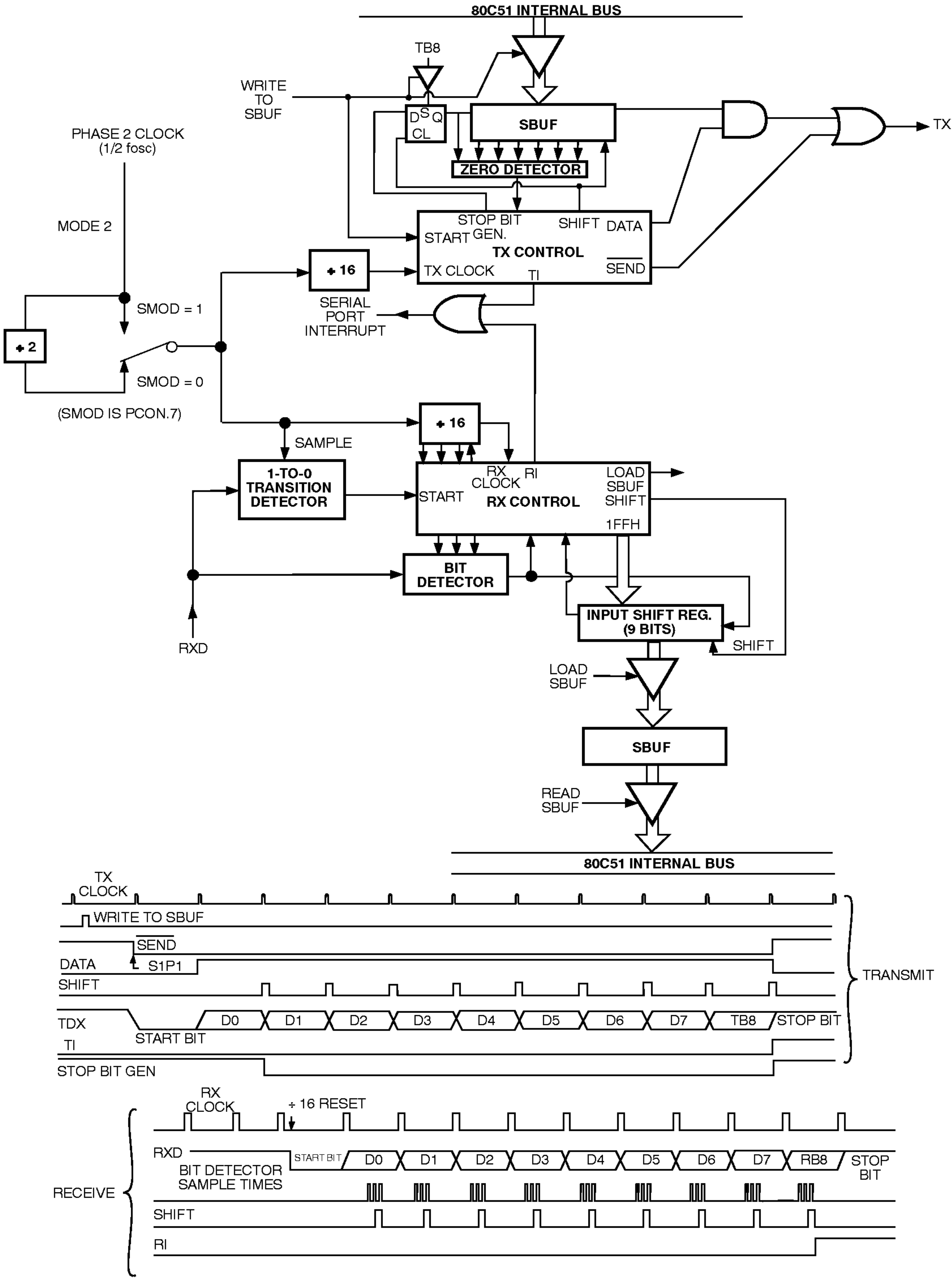
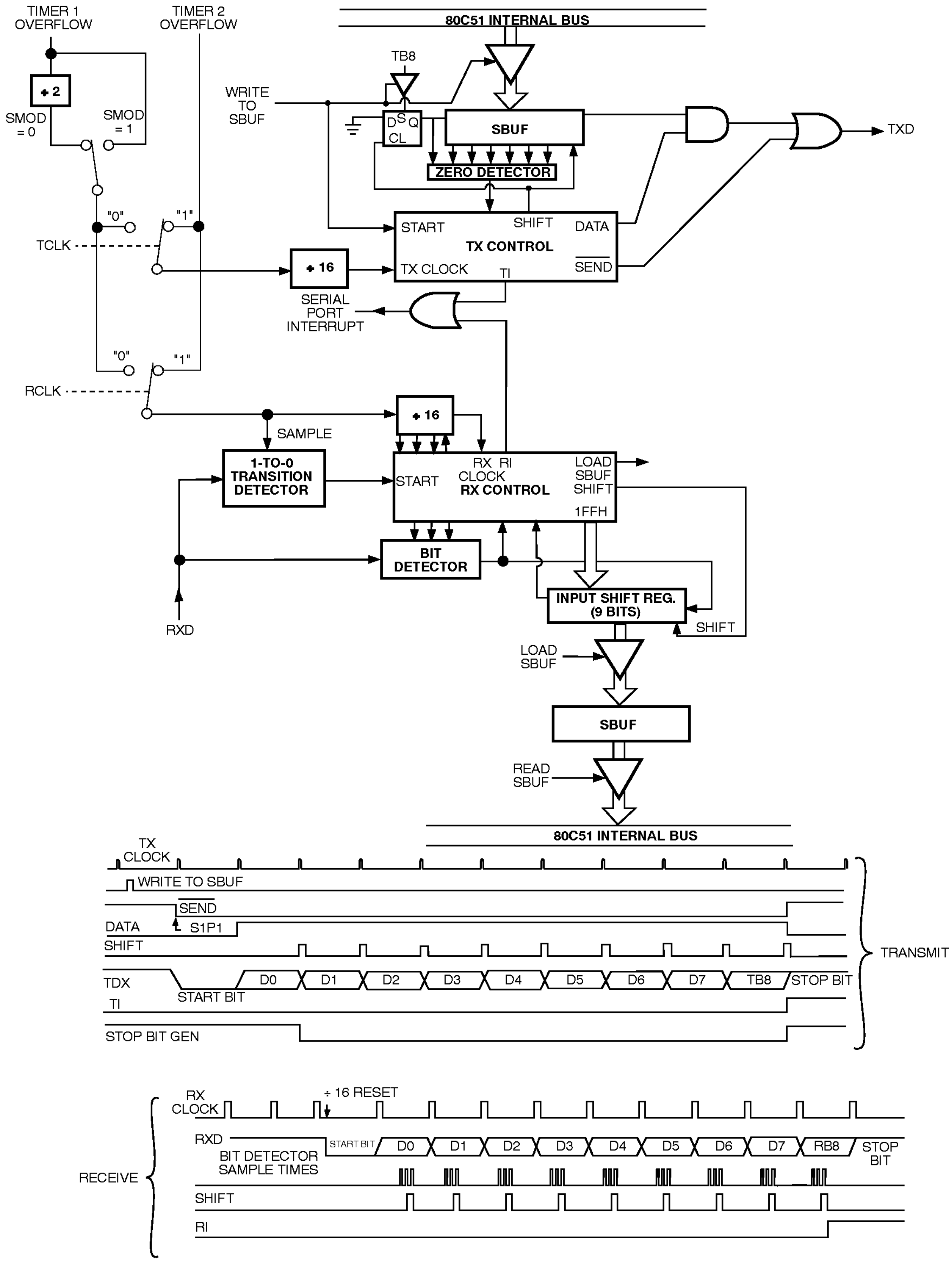


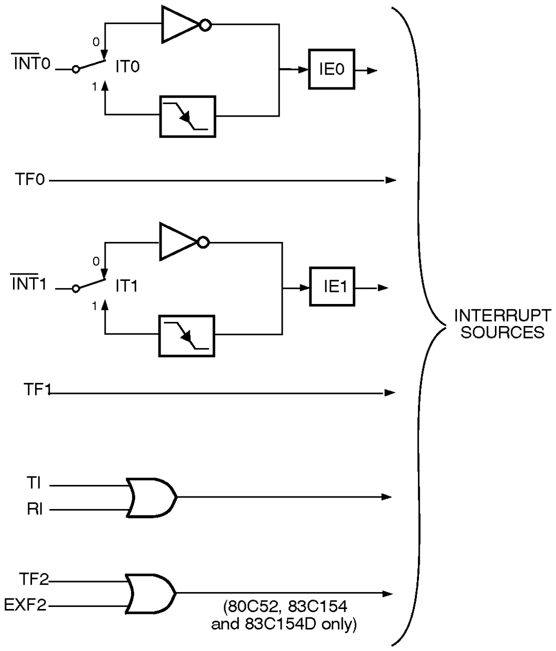
Figure 1.24B. Serial Port Mode 3. TCLK, RCLK and Timer 2 are present in the 80C32/80C52, 80C154/83C154 and in the 83C154D.



1.8 Interrupts

The 80C51 provides 5 interrupt sources. The 80C52 83C154 and 83C154D provide 6. Theses are shown in Figure 1-25.

Figure 1.25. MHS C51 Interrupt Sources.



The external interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or 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 it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective timer/counter registers (except see Section 1.6 for Timer 0 in mode 3). 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.

The Serial Port Interrupt is generated by the logical OR of RI and TI. 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 RI or TI that generated the interrupt, and the bit will have to be cleared in software.

In the 80C52, 83C154 and 83C154D, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. 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 in software.

All of the bits that generate interrupts can be set or cleared by sofware, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in sofware.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 1-26). Note that IE contains also a global disable bit, EA, which disables all interrupts at once.

Figure 1.26. IE : Interrupt Enable Register.

(MSB) (LSB)

EA	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enables bit.
–	IE.6	reserved.
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (Figure 1-27). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

Figure 1.27. IP : Interrupt Priority Register.

(MSB) (LSB)

X	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
–	IP.7	reserved
–	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it do the higher priority level.
PS	IP.4	defines the Serial Port Interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the external interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

If two requests of different priority levels are received simultaneoulsy, the request of higher priority level is serviced. If requests of the same priority level are received simultaneously, an internal polling sequence determine which request is serviced. Thus within each priority level is a second priority structure determined by the polling sequence, as follows :

	SOURCE	PRIORITY WITHIN LEVEL
1.	IE0	(highest)
2.	TF0	
3.	IE1	
4.	TF1	
5.	RI + TI	
6.	TF2 + EXF2	(lowest)

Note that the "priority within level" structure is only used to resolve *simultaneous requests of the same priority level*.

How Interrupts Are Handled

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

1. An interrupt of equal or higher priority level is already in progress.

2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.

3. The instruction in progress is RETI or any access to the IE or IP registers.

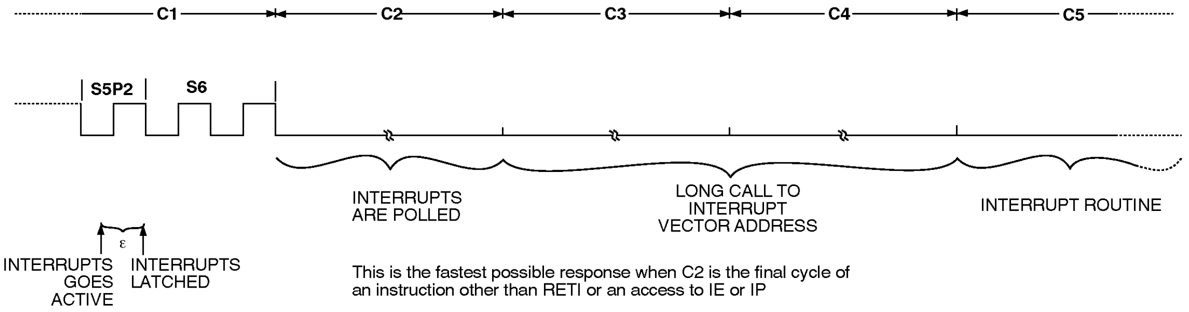
Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least *one more* instruction will be executed before any interrupt is vectored to.

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 then that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not *still* active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the facts that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 1-28.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 1-28, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Figure 1.28. Interrupt response Timing Diagram.



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP

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, and in other cases it doesn’t. It never clears the Serial Port or Timers 2 flags. This has to be done in the user’s software. It clears an external interrupt flag (IE0 or IE1) only if it was 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 PC with an address that depends on the source of the interrupt being vectored to, as shown below.

SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
TF2 + EXF2	002BH

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that 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.

External Interrupts

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

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt 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.

Response Time

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ levels are inverted and latched into IE0 and IE1 and S5P2 of every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the service routine. Figure 1-28 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level 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 an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 8 cycles.

1.9 Single Step Operation

The 80C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least once instruction of the interrupted program is executed. One way to use this feature for single-step operation is to program one of the external interrupts (say, $\overline{\mathrm{INT0}}$) to be level-activated. The service routine for the interrupt will terminate with the following code :

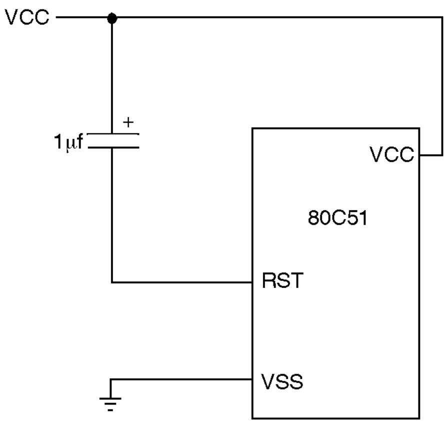
```
JNB      P3.2,$      WAIT HERE TILL INTO
                     GOES HIGH
JB    P3.2,$         NOW WAIT HERE TILL
                     IT GOES LOW
RETI                 GO BACK AND EXECUTE
                     ONE INSTRUCTION
```

Now, if the $\overline{\mathrm{INT0}}$ pin, which is also the P3.2 pin, is hold normally low, the CPU will go right into the External interrupt 0 routine and stay there until $\overline{\mathrm{INT0}}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

1.10 Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger.

Figure 1.29. Power on Reset Circuit.



A reset accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running*. The CPU responds by executing an internal reset. It also configures the ALE and $\overline{\mathrm{PSEN}}$ pins as inputs. (They are quasi-bidirectional). The internal reset is executed during the second cycle in which RST is high and is repeated every cycle until RST goes low. It leaves the internal registers as follows :

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP (80C51)	XXX00000B
IP (80C52, 83C154 and 83C154D)	XX000000B
IE (80C51)	0XX00000B
IE (80C52, 83C154 and 83C154D)	0X000000B
TMOD	00H
TCON	00H
T2CON (80C52, 83C154 and 83C154D)	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H (80C52 83C154 and 83C154D)	00H
RCAP2L (80C52, 83C154 and 83C154D)	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H
PCON (80C51 and 80C52)	0XXX0000B
PCON (83C154 and 83C154D)	000X0000B

The internal RAM is not affected by reset. When VCC is turned on, the RAM content is indeterminate unless the part is returning from a reduced power mode of operation.

Power-on reset

An automatic reset can be obtained when VCC is turned on by connecting the RST pin to VCC through a 1 µf capacitor providing the VCC risetime does not exceed a millisecond and the oscillator start-up time does not exceed 10 milli-seconds. This power-on reset circuit is shown in Figure 1-29. When power comes on, the current drawn by RST commences to charge the capacitor. The voltage at RST is the difference between VCC and the capacitor voltage, and decreases from VCC as the cap charges. The larger the capacitor, the more slowly VRST decreases. VRST must remain above the lower threshold of the Schmitt Trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

1.11 Power-Saving Modes of Operation

For applications where power consumption is a critical factor, the MHS C51 parts provide two power mode : power-down and idle mode. The first one reduces the consumption up to few microamperes and the second one divides the consumption roughly by 25 %. Both of the modes are controlled by software via the PCON register. In Power-Down mode (PD = 1, PCON = 87H => XXXX XX1X) the oscillator is frozen. In idle mode (IDL = 1, PCON = 87H => XXXX XX01). The oscillator continues to run and the interrupt, serial port, and timer blocks continue to be clocked but the clock signal is gated off the CPU. The activities of the CPU no longer exists unless waiting for an interrupt request. Both Power-Down and Idle mode are explained below. Further function concerning the MHS C154 parts will be explain in the next chapter.

Idle Mode

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirely : the Stack Pointer, Program Counter, Program Status Word, Accumulator and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{PSEN}$ hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into idle.

The flag bits GF0 and GF1 can be used to give and indication if an interrupt occured during normal operation or during and Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The over way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

Figure 1.30. PCON : Power Control Register.

(MSB) (LSB)

SMOD	–	–	–	GF1	GF0	PD	IDL

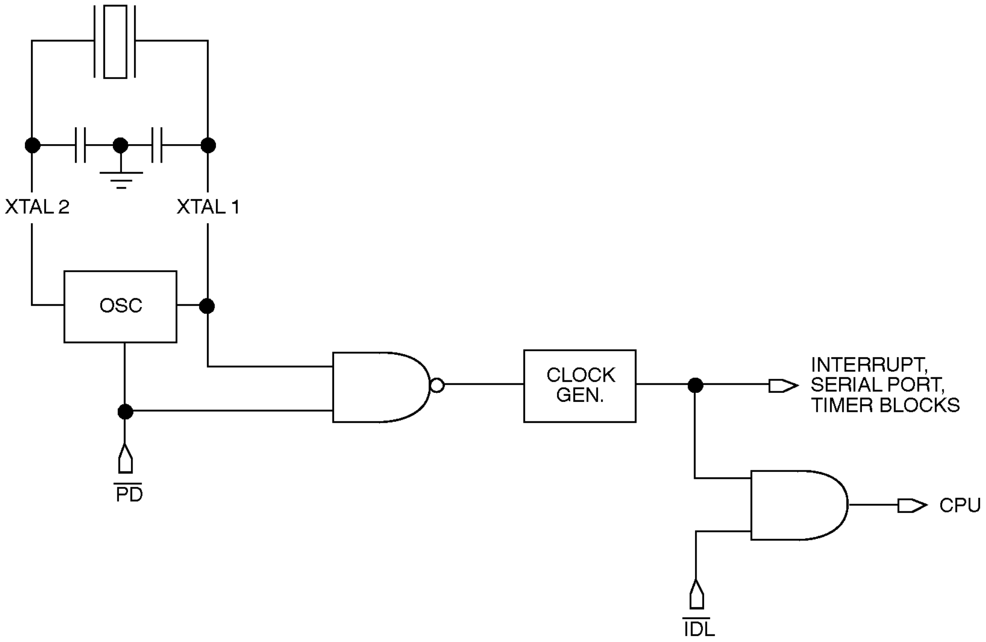
Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
–	PCON.6	(Reserved)
–	PCON.5	(Reserved)
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).

Power Down Mode

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Registers are held. The port pins output the values held by their respective SFRs. ALE and $\overline{PSEN}$ output lows.

Figure 1.31. Idle and Power Down Hardware.



The only exit from Power Down is a hardware reset. Reset redefines all the SFRs, but does not change the on-chip RAM.

In the Power down mode of operation, VCC can be reduced to minimize power consumption. Care must be taken, however, to ensure that VCC is not reduced before the Power Down mode is invoken, and that VCC is restored to its normal operating level, before the Power Down mode is terminated. The reset that terminates Power Down also frees the oscillator. The reset should not be activated before VCC is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10 msec).

This table shows the state of ports during idle and power-down modes.

MODE	PROGRAM MEMORY	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

1.12 More about the On-chip Oscillator

The on-chip oscillator circuitry for 80C51's family, shown in Figure 1-32, consists of a single stage linear inventer for use as a crystal-controlled, positive reactance oscillator.

The on-chip oscillator is able to run with a crystal or with ceramic resonnator. The fig. 1.33 shows the schematic to work which a crystal and a ceramic resonnator working on a fundamental mode.

Figure 1.32. On-chip oscillator Circuit.

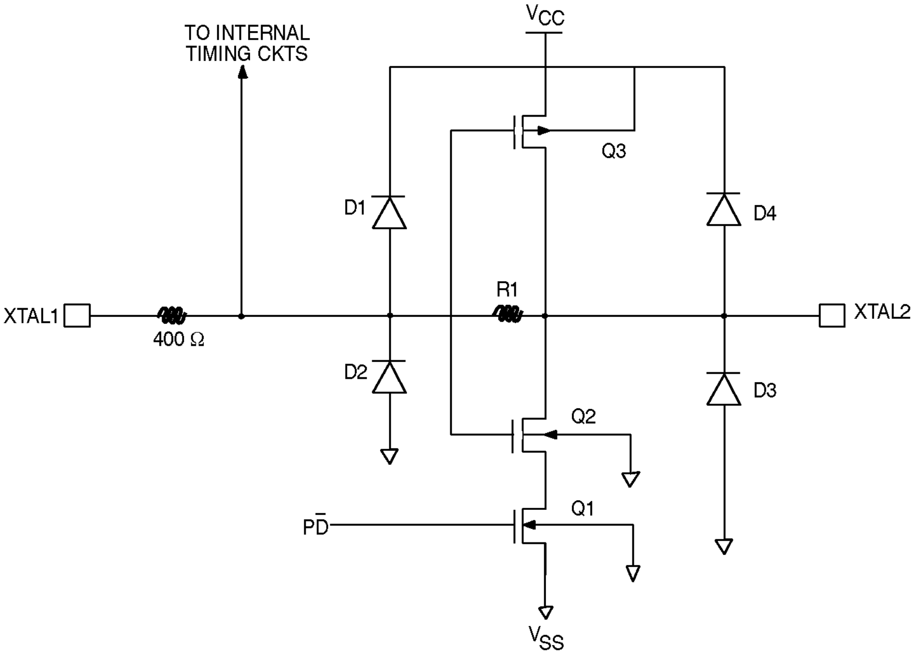


Figure 1.33. Fundamental Resonance.

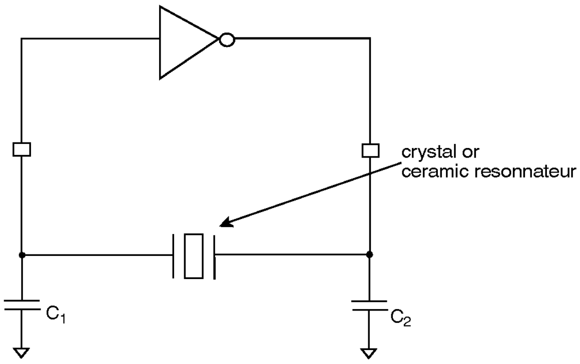
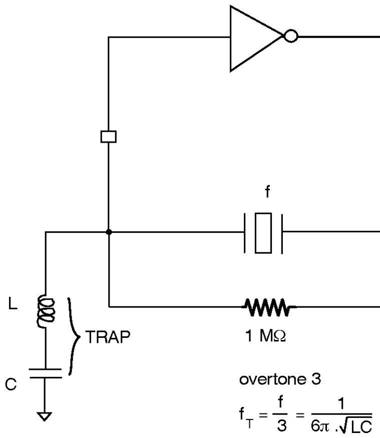


Figure 1.34. Overtone 3 Resonance.



The fig. 1.34 shows the use of a crystal working on an overtone 3 resonance.

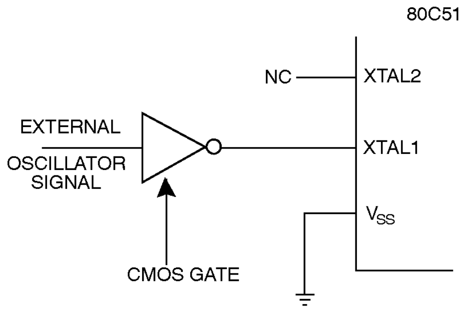
An overtone 3 crystal doesn’t work on its fundamental resonance but on its third overtone. So it’s necessary to catch in its fundamental frequency. The trap consists of the inductor L and the capacitance C. The trap frequency is the running frequency of the crystal divides by 3. An external resistor of 1 MΩ is connected on the both side of the crystal to decrease the gain on the amplifier. Cause the equivalent inductor for a overtone 3 crystal is more larger than a fundamental crystal, the oscillator needs less energy. Without external resistor the level on pin XTAL1 isn’t enough to control the internal clock circuitry.

The MHS-51 parts can by controlled by an external clock. In this case the external clock signal is connected directly on XTAL1 input and XTAL2 in left floating (fig. 1.35).

1.13 Internal Timing

Figures 1.36 through 1.39 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL2 signal and events at other pins.

Figure 1.35. Driving the C51 parts with an external clock source.



Rise and gall times are dependent on the external loading that each pin must drive. They are often taken to be something in the neighborhood of 10 nsec, measured between 0.8 V and 2.0 V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, VCC, and manufacturing lot. If the XTAL2 waveform is taken as the timing reference, prop delays may vary from 25 to 125 nsec.

The AC Timings section of the data sheets do not reference any timing to the XTAL2 waveform. Rather, they relate the critical edges of control and input signals to each other. The timing published in the data sheets include the effects of propagation delays under the specified test conditions.

1.14 C51 Pin Description

VCC : Supply voltage.

VSS : Circuit ground potential.

Port 0 : Port 0 is an 8-bit open drain bidirectional I/O Port. As an open drain output port it can sink 8 LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will functions as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pullups when emitting 1s. Port 0 also emits code bytes during program verification. In that application, external pullups are required.

Port 1 : Port 1 is an 8-bit bidirectional I/O port with internal pullups. The port 1 output buffers can sink/source 4 LS TTL loads. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

In the 80C52, 83C154 and 83C154D pins P1.0 and P1.1 also serve the alternate functions of T2 and T2EX. T2 is the Timer 2 external input. T2EX is the input through which a Timer 2 “capture” is triggered.

Port 2 : Port 2 is an 8-bit bidirectional I/O port with internal pullups. The port 2 output buffers can sink/source 4 LS TTL loads. Port 2 emits the high-order addres byte during accesses to external memory that use 16-bit addresses. In this application it uses the strong internal pullups when emitting 1s.

Port 3 : Port 3 is an 8-bit bidirectional I/O port with internal pullups. It also serves the functions of various special features of the MHS-C51 Family, as listed below :

PORT PIN	ALTERNATE FUNCTION
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

The Port 3 output buffers can source/sink 4 LS TTL loads.

RST : reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE : Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, for external timing or clicking purposes, even when there are no accesses to external memory. (However, one ALE pulse is skipped during each access to external Data Memory).

$\overline{\text{PSEN}}$: Program Store Enable is the read strobe to external Program Memory. When the device is executing out of external Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle (except that two $\overline{\text{PSEN}}$ activations are skipped during accesses to external Data Memory). $\overline{\text{PSEN}}$ is not activated when the deviced is executing out of internal Program Memory.

$\overline{\text{EA}}$: When $\overline{\text{EA}}$ is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH in the 80C51, or 1FFFH in the 80C52, or 3 FFFH in the 83C154 or 7FFFH in the 83C154D). Holding $\overline{\text{EA}}$ low forces the CPU to execute out of external memory regardless of the Program Counter value. In the 80C31, 80C32 and 80C154. $\overline{\text{EA}}$ must be externally wired low.

XTAL1 : Input to the inverting oscillator amplifier.

XTAL2 : Output from the inverting oscillator amplifier.

Figure 1.36. External Program Memory Fetches.

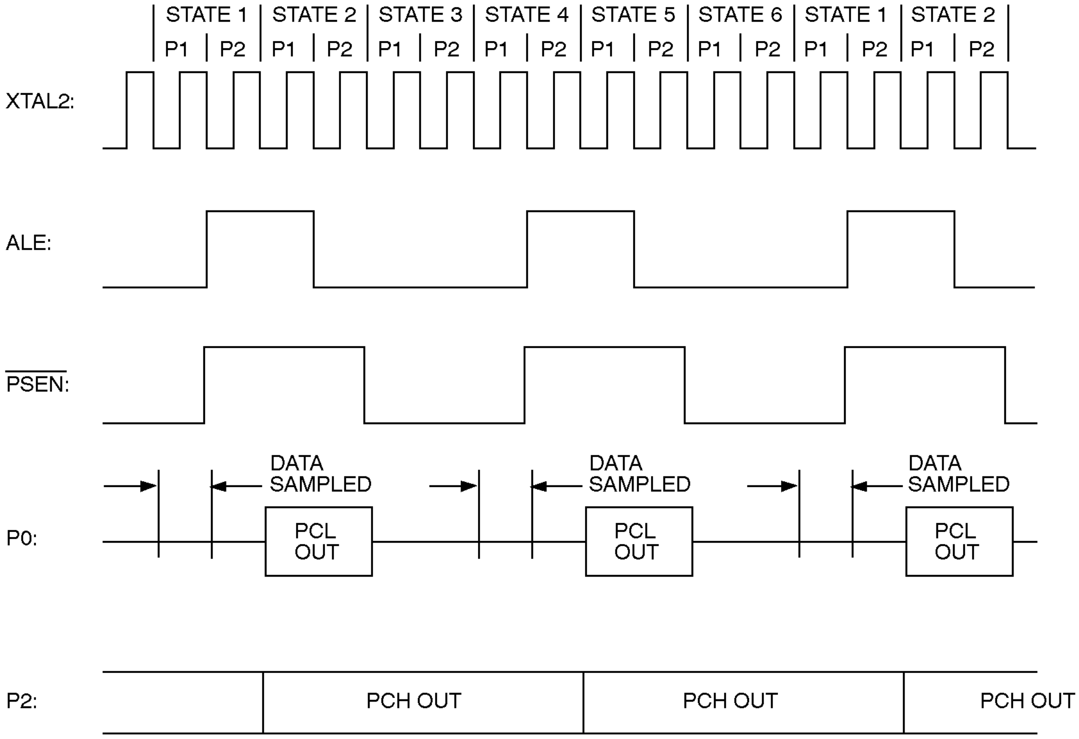


Figure 1.37. External Data Memory Read Cycle.

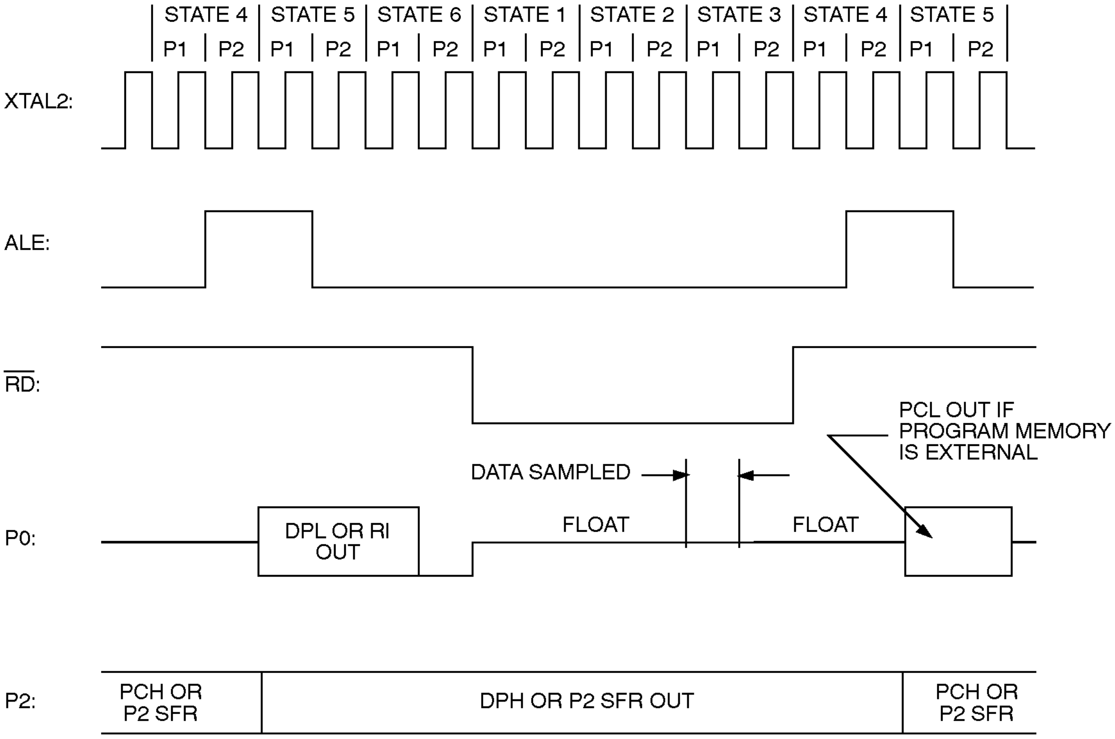


Figure 1.38. External Data Memory Write Cycle.

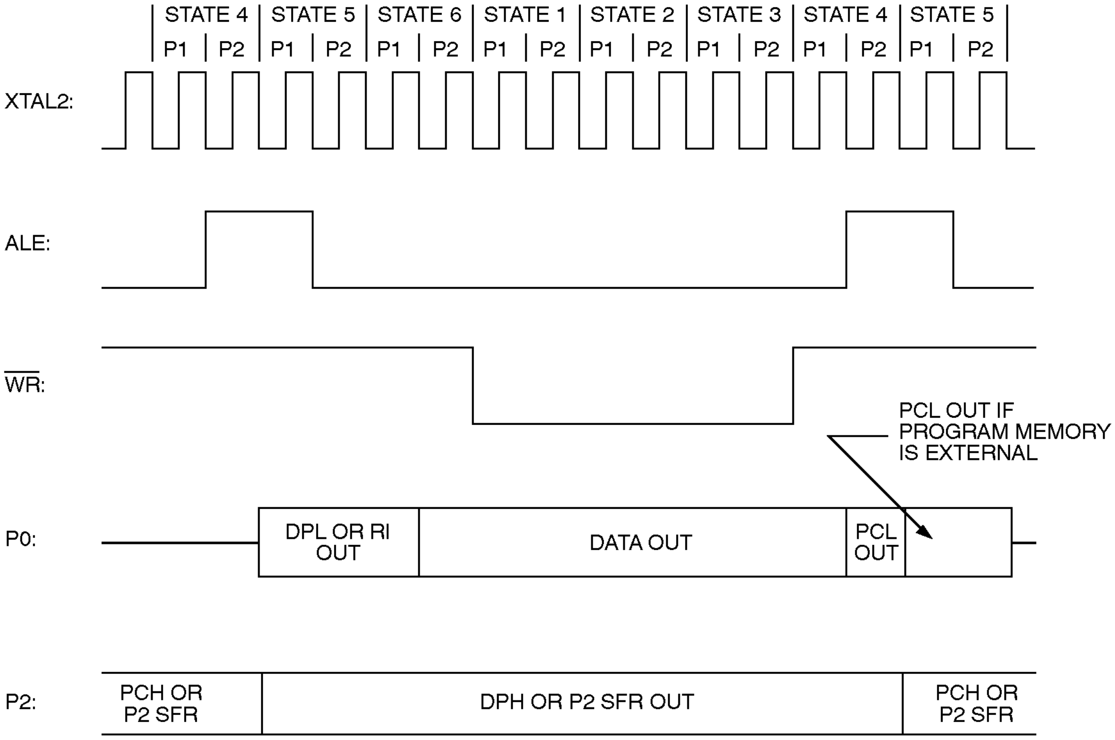
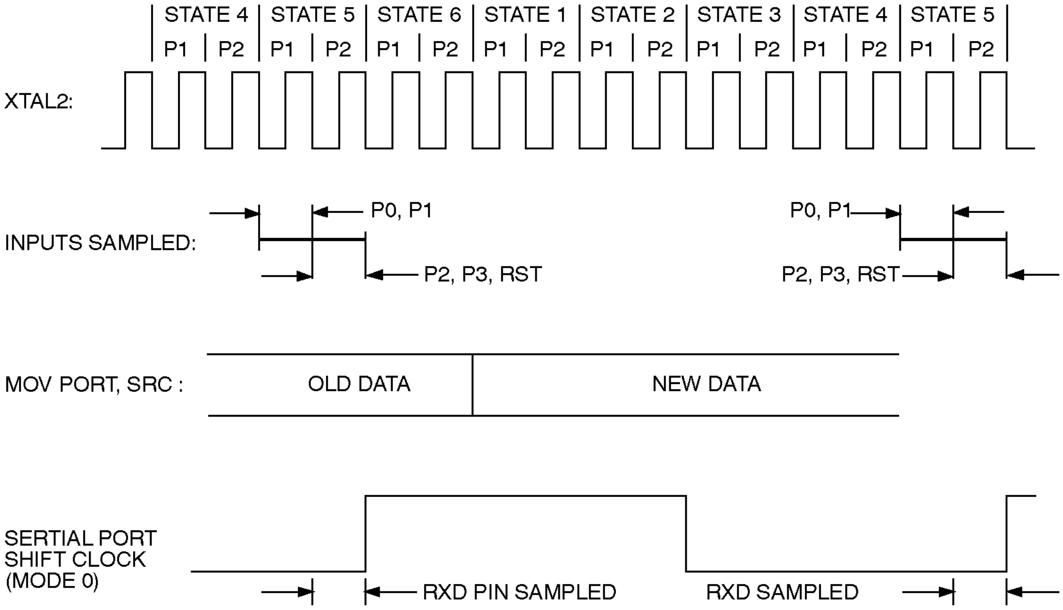


Figure 1.39. Port Operation.



2. More Features for C154 Parts

This chapter explains, in details, the new features of the 83C154 and 83C154D. The only one difference between 83C154 and 83C154D is the internal ROM size (respectively 16 K bytes and 32 K bytes long).

The major new features of MHSC154 is listed below :

- I/O part impedance selection
- Watch-Dog and 32-Bit TIMER/Counter Mode
- Power-down mode
  - Software control
  - Hardware control
- Frame and Overrun error serial link detection.

All these new features are controlled via the IOCON register (new one) and the PCON register.

2.1 I/O Port Impedance

The structure and behaviour of the 83C154s'ports P1, P2 and P3 are indentical to those of the 80C52. Only the control block for the different pullups and pulldowns has been changed. The pullup resistance value can be programmed by means of the IOCON register.

There are three possible values :
– three states (P1, P2, P3 and N are OFF),
– high impedance (100 kΩ, P2 = ON),
– low impedance (10 kΩ, P3 = ON).

Figure 2.1 is a functional diagram of the PORT.

Figure 2.2 shows the configuration of the IOCON register which is used to set the right value of the impedance port.

Figure 2.1. I/O Port Block Diagram.

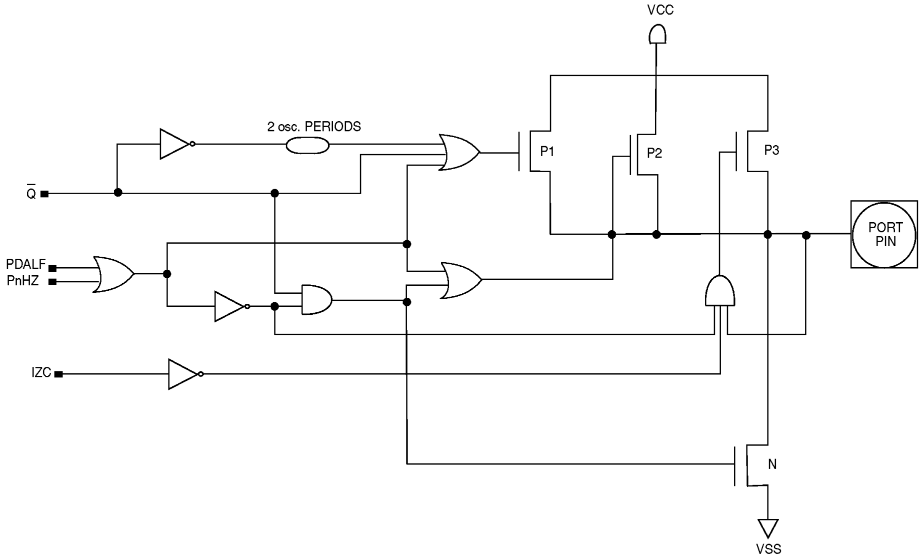


Figure 2.2. IOCON Register Configuration.

(MSB) (LSB)

WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

Symbol	Position	Function
ALF	IOCON.0	– Set to 1 and in Power mode PORTS 1, 2 and 3 are floating.
P1HZ	IOCON.1	– If P1HZ = 0 and IZC = 0, PORT P1 is at low impedance. – If P1HZ = 0 and IZC = 0, PORT P1 is at high impedance. – If P1HZ = 1, PORT P1 is floating.
P2HZ	IOCON.2	– If P2HZ = 0 and IZC = 0, PORT P2 is at low impedance. – If P2HZ = 0 and IZC = 0, PORT P2 is at high impedance. – If p2HZ = 1, PORT P2 is floating.
P3HZ	IOCON.3	– If P3HZ = 0 and IZC = 0, PORT P3 is at low impedance. – If P3HZ = 0 and IZC = 0, PORT P3 is at high impedance. – If P3HZ = 1, PORT P3 is floating.
IZC	IOCON.4	– In conjunction with PnHZ selects the output pullup value.

Low impedance

This mode is the default mode upon a reset and it is compatible with C51 parts. The configuration of IOCON is explained by the figure 2.2. Whenever PnHZ and IZC are equal to zero, P1, P2 and P3 are in this mode. In this case 3 LS TTL loads can be interfaced.

High impedance

This mode is invoked by setting PnHZ = 0 and IZC = 1. Only the transistor PZ of the fig. 2.1 is on and one LS TTL load can be interfaced.

Three states mode

Two different modes can be used. The first one allows to set each of the 3 ports in three states during a normal operation. The second one allows to set all the 3 parts in three states by entering in the Power-Down mode.

Whenever PnHZ is set to 1, the part is in three states mode. If ALF is set to 1, at each time the Power-Down mode is called all the three parts are in the three states. When the C154 part exist from the Power-Down mode, the part impedance is the impedance just before entering in this mode (the part switches as soon as the interrupt request is generated, not after the oscillator start-up). The three states mode switch-off all the transistor.

2.2 Watch-dog and 32-bit Timer/Counter Mode

TIMER/COUNTER of C51 family can be configured in four modes. With C154 parts, two new modes can be used.

Both of them use TIMER0 and/or TIMER 1. The first one is used like a 32-Bit Timer/Counter and the second one is used like a 8/13/16/32-Bit Watch-Dog. Both of this two modes is programmed by software by using the IOCON register. The 32-Bit mode is on by setting the Bit T32. The Watch-Dog mode is on by setting the bit WDT. The fig. 2.3 shows the configuration of the register IOCON.

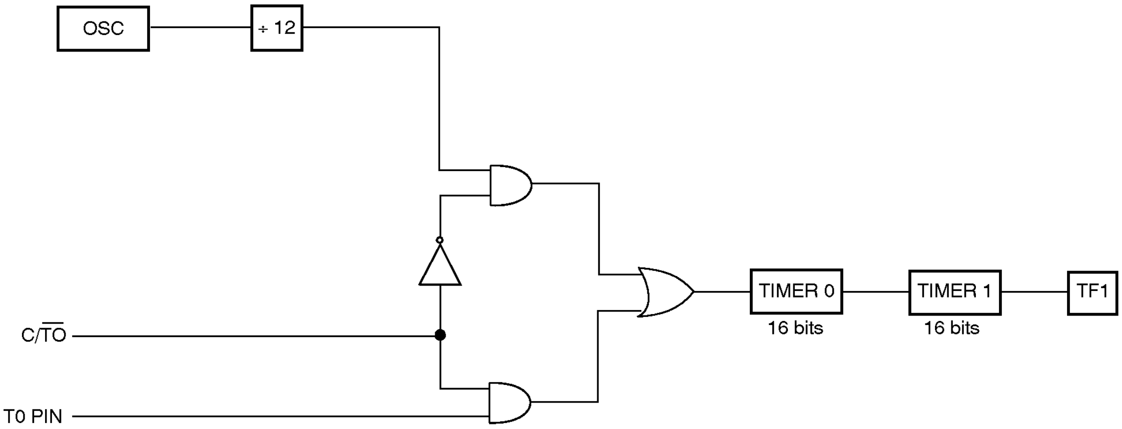
Figure 2.3. 32 Bit/Watchdog Mode.

(MSB) (LSB)

WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF

Symbol	Position	Function
T32	IOCON.6	– If T32 = 1 and if C/T0 = 0, T1 and T0 are programmed as a 32 bit TIMER. – If T32 = 1 and if C/T0 = 1, T1 and T0 are programmed as a 32 bit COUNTER.
WDT	IOCON.7	– If WDT = 1 and according to the mode selected by TMOD, and 8 bit or 32 bit WATCHDOG is configured from TIMERS 0 and 1.

Figure 2.4. 32 Bit Mode.



32 Bit Mode

T32 = 1 enables access to this mode. As shown in figure 2.4 this 32 bit mode consists in cascading TIMER 0 for the LSBs and TIMER 1 for the MSBs.

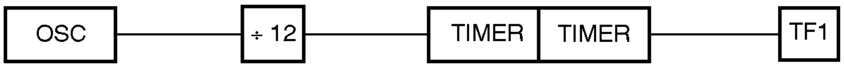
T32 = 1 starts the timer/counter and T32 = 0 stops it.

It should be noted that as soon as T32 = 0, TIMERs 0 and 1 assume the configuration specified by register TMOD. Moreover, if TR0 = 1 or if TR1 = 1, the content of the TIMERs evolves. Consequently, in 32 bit mode, if the TIMER/COUNTER must be stopped (T32 = 0). TR0 and TR1 must be set to 0.

32 Bit Timer

Figure 2.5 illustrates the 32 Bit TIMER mode.

Figure 2.5.



In this mode, T32 = 1 and C/$\overline{T0}$ = 0, the 32 bit timer is incremented on each S3P1 state of each machine cycle. An overflow of TIMER 0 (TF0 has not been set to 1) increments TIMER 1 and the overflow of the 32 bit TIMER is signalled by setting TF1 (S5P1) to 1.

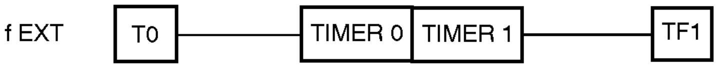
The following formula should be used to calculate the required frequency :

$$f = \frac{OSC}{12 \times (65536-(T0,\ T1))}$$

32 Bit Counter

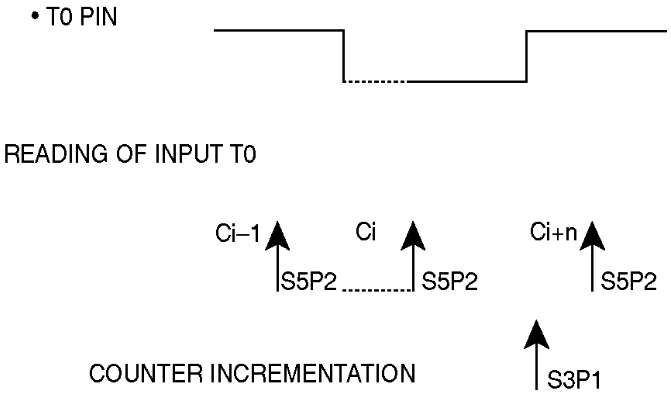
Figure 2.6 illustrates the 32 BIT COUNTER mode.

Figure 2.6. 32 Bit Counter Configuration.



In this mode, T32 = 0 and C/$\overline{T0}$ = 1. Before it can make an increment, the 83C154 must detect two transitions on its T0 input. As shown in figure 2.7 input T0 is sampled on each S5P2 state of every machine cycle or, in other words, every OSC ÷ 12.

Figure 2.7. Counter Incrementation Condition.



The counter will only evolve if a level 1 is detected during state S5P2 of cycle Ci and if a level 0 is detected during state S5P2 of cycle Ci + n.

Consequently, the minimal period of signal fEXT admissible by the counter must be greater than or equal to two machine cycles. The following formula should be used to calculate the operating frequency.

$$f = \frac{fEXT}{(65536–(T0,\ \ T1))}$$

$$fEXT \leq \frac{OSC}{24}$$

Watch Dog Mode

WDT = 1 enables access to this mode. As shown in figure 2.8 all the modes of TIMERs 0 and 1, of which the overflows act on TF1 (TF1 = 1), activate the WATCH DOG Mode.

If C/$\overline{T}$ = 0, the WATCH DOG is a TIMER that is incremented every machine cycle. If C/$\overline{T}$ = 1, the WATCH DOG is a counter that is incremented by an external signal of which the frequency cannot exceed OSC ÷ 24.

The overflow of the TIMER/COUNTER is signalled by raising flag TF1 to 1. The reset of the 83C154/83C154D is executed during the next machine cycle and lasts for the next 5 machine cycles. The results of this reset are identical to those of a hardware reset. The internal RAM is not affected and the special register assume the values shown in Table 2.9.

Table 2.9. Content of the SFRs after a reset triggered by the WATCHDOG.

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	0000H
P0–P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

As there are no precautions for protecting bit WDT from spurious writing in the IOCON register, special care must be taken when writing the program. In particular, the user should use the IOCON register bit handing instructions :

– SETB and CLR x

in preference to the byte handling instructions :

– MOV IOCON, # XXH, ORL IOCON, # XXH,

– ANL IOCON, # XXH,

External Counting in Power-down Mode
(PD = PCON.1 = 1)

In the power-down mode, the oscillator is turned off and the 83C154s'activity is frozen. However, if an external clock is connected to one of the two inputs, T1/T0, TIMER/COUNTERS 0 and 1 can continue to operate. In this case, counting becomes asynchronous and the maximum, admissible frequency of the signal is OSC : 24.

The overflow of either counter TF0 or TF1 causes an interrupt to be serviced or forces a reset if the counter is in the WATCH DOG MODE (T32 = ICON.7 = 1).

2.3 Power-Reducing Mode

Basically the Power-Reducing Mode of the MHS C154 parts are 100 % compatible with the 80C51 and 80C52 parts. However both Idle and Power-Down mode are improved with some new powerful features.

Idle mode is improved by giving the software possibility to execute or not the software interrupt routine when an interrupt request occurs (Recover Power Mode).

Power-Down mode is now more powerful because an interrupt request can avake the MHS C154 parts. In addition an external hardware signal can control entirely the mode (Hardware Power Mode).

Details on these new features are given below.

Idle Mode

This mode is basically compatible with the MHS C51 parts (refer to the chapter 1.11). The new feature concerns the way to exit from this mode. Now with the Recover Power Mode, the software interrupt routine is or not executed when the interrupt routine occurs. This mode is activated by setting the bit RPD in PCON register. In this case the next instruction executed is the next following the IDLE instruction (MOV PLON, # 01).

Power-Down Mode

Software control

This mode is basically compatible with the C51 parts (refer to the chapter 1.11). The new features concern the way to exit from this mode.

With the C51 parts the only way to complete this mode is to apply an hardware reset. The newest thing is the possibility to exist from this mode by an interrupt request coming from the both external interrupts and the both counter 0 and 1 if an external clock is connected on pin T1 or T0.

Likewise IDLE mode, it is possible to execute or not the software interrupt routine. By setting the RPD bit the next instruction executed, after the interrupt request has been processed, will be the instruction following the POWER-DOWN intruction. If the RPD bit is not set, the program will continue by executing the software interrupt routine. The fig. 2.10 shows the behaviour of the Recover Power Mode.

Figure 2.10. Example in Recover Power Mode.

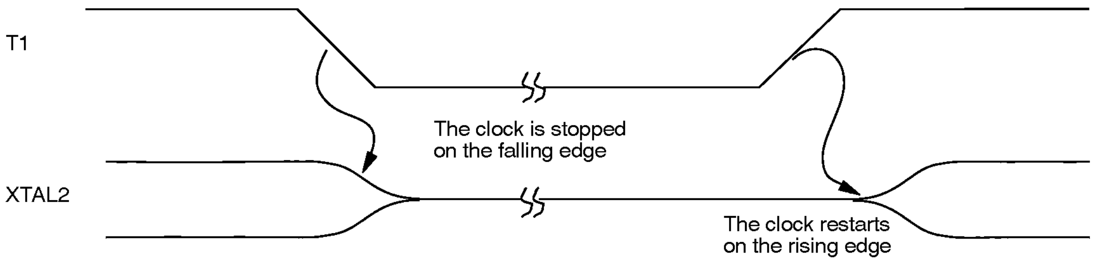
```
ex :
                                ; The Recover Mode is enable
                                ; The Power-Down is enable
    Adr n  MOV  PCON,  #001XXX10...
                                ; The CPU waits for an asynchronous
                                ; event
                                    |       |       |
                                    |       |       |
                                    |       |       |
                                    |       |       |
```

```
                                ; An interrupt request occurs and
                                ; the instruction located at the
                                ; address Adr n + 1 is executed
    Adr n + 1
                                    |       |       |
                                    |       |       |
                                    |       |       |
```

Hardware control

This mode is new and controls the Power-Down by an external hardware signal connected on pin T1 (P3.5). This mode is called by setting HPD bit in the PCON register. The 83C154 will be in Power-Down as soon as T1 input will be drive by a falling edge and will remain in this state until T1 input will be drive by a rising edge. The fig. 2.11 shows the behaviour of the Hardware Power Mode.

Figure 2.11. Behaviour of the Hardware Power Mode (HPD = 1).



T1
XTAL2
The clock is stopped on the falling edge
The clock restarts on the rising edge

The time takes by the oscillator to restart depends of the crystal and the capacitors connected on both side of the crystal (typically 10 ms).

Software and hardware control

This two modes can be mixed to control the Power-Down Mode. Entry to the mode can be made either by setting PD bit to 1 or by setting HPD bit to 1 and presenting a falling edge on T1 input. Exit from this mode can be made if the software and hardware conditions are met : a rising edge on T1 input and an interrupt request. If these two conditions are not satisfied, only an hardware reset can complete the mode.

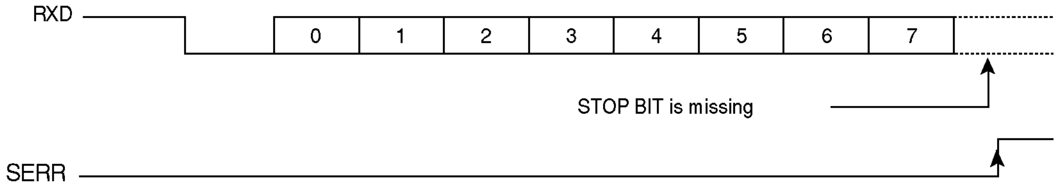
2.4 Frame and Overrun Error Serial Link Detection

This feature is new and allows to the user to detect a serial link error. Two kinds of error can be detected during a reception : OVERRUN ERROR and FRAME ERROR. Both of them set the SERR bit in the IOCON register at the half of the stop bit. This must be cleared by software.

Frame error

This error occurs when the format of the received Data is wrong. The figure 2.12 shows an example of a Frame error.

Figure 2.12. FRAME ERROR example, STOP BIT is missing.

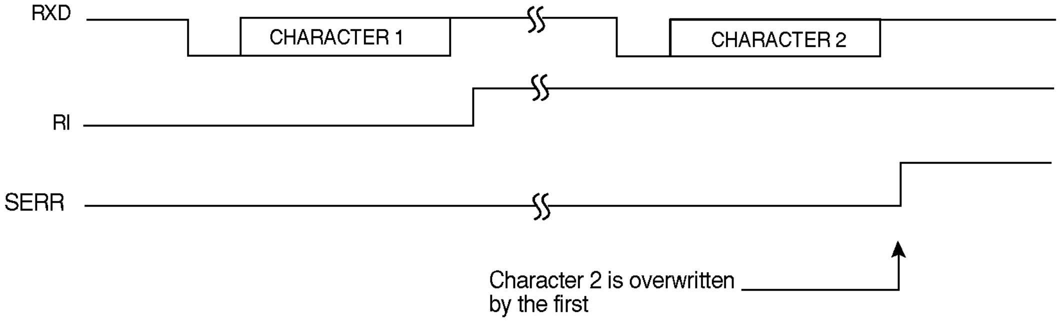


In this example the receiver waits for a STOP BIT to complete the frame reception. Unfortunately the stop bit isn't there and the receiver indicates the frame error by setting to 1 the SERR bit in the IOCON register.

Overrun Error

This error occurs, when a character received and not read by the C.P.U, is overwritten by a new one. The figure 2.13 shows an example of OVERRUN ERROR.

Figure 2.13. OVERRUN ERROR example.



In this example the character 1 is received and the RI bit is set to 1. A second character is sent before the CPU reads the first one. The character 1 is overwritten and SERR bit is set to 1 to indicate the loss of the first character.

Note
With the C154 parts the RI bit isn't set on the same time than the C51 parts. With the C51 parts the RI bit is set on the last data bit. With the C154 the RI bit is set on the stop bit.

The information contained herein is subject to change without notice. No responsibility is assumed by MATRA MHS SA for using this publication and/or circuits described herein : nor for any possible infringements of patents or other rights of third parties which may result from its use.